

# Création d'un site Web avec HTML et CSS

Aimé DIUMI DIKOLO

[www.wiscorp.com](http://www.wiscorp.com)

Ce cours est encore en pleine rédaction, nous comptons sur vos remarques, critiques et suggestions pour améliorer la version finale du cours. Nous nous sommes inspirés du cours de Matthieu Nebra sur HTML et CSS.

## Table des matières

<b>I.</b>	<b>NOTIONS DE BASE EN HTML</b> .....	7
I.1	INTRODUCTION.....	7
I.2	LES BALISES ET LES ATTRIBUTS.....	9
I.2.1	Les balises .....	9
I.2.2	Les attributs .....	10
I.3	STRUCTURE DE BASE D'UNE PAGE HTML .....	11
I.4	ENREGISTREMENT D'UN FICHER HTML .....	15
I.5	Commentaires en HTML .....	18
I.6	ORGANISATION DU TEXTE .....	20
I.6.1	Le retour à la ligne .....	20
I.6.2	Les paragraphes.....	21
I.6.3	Les titres.....	22
I.6.4	Les éléments em, mark et strong.....	23
I.6.5	Les listes .....	26
I.7	LES IMAGES .....	38
I.8	LES LIENS .....	41
I.8.1	Lien vers une autre page du site .....	41
I.8.2	Lien vers un autre site .....	47
I.8.3	Lien vers une ancre (point de repère).....	47
I.8.4	Lien pour télécharger un fichier .....	50
I.8.5	Lien pour envoyer un email.....	50
I.9	L'AUDIO ET LA VIDEO.....	50
I.9.1	L'audio.....	50
I.9.2	La vidéo .....	51
<b>II.</b>	<b>MISE EN FORME DU SITE AVEC CSS</b> .....	54
II.1	GENERALITES .....	54
II.1.1	Structure du code CSS .....	54
II.1.2	Où écrit-on le code CSS .....	55
II.1.3	Les commentaires en CSS.....	66
II.1.4	Appliquer un style particulier avec id et class .....	66
II.1.5	Les balises universelles.....	70
II.2	LES SELECTEURS AVANCES .....	73
a)	Appliquer le même style à plusieurs éléments.....	73
b)	* : Le sélecteur universel .....	74
c)	A B : sélectionne une balise B contenue dans A .....	75

d) A+B : une balise B qui suit la balise A.....	77
e) A[Attribut] : sélectionne une balise contenant un attribut donné.....	78
f) A[Attribut="Valeur" ] .....	79
g) A[attribut*= "Val"] .....	80
h. A > B : sélectionne B qui est enfant (descendant) direct de A .....	81
i. A~B : sélectionne tous les éléments B précédés par l'élément A. ....	83
j. A[b^="c"].....	84
k. A[b\$="c"].....	85
l. A[b*="c"].....	85
<b>II.3 NOTION D'HERITAGE EN CSS.....</b>	<b>86</b>
<b>II.4 MISE EN FORME DU TEXTE.....</b>	<b>89</b>
II.4.1 La taille du texte .....	89
II.4.2 La police .....	92
II.4.3 L'italique .....	96
II.4.4 Mettre le texte en gras.....	98
II.4. 5 Le soulignement du texte .....	99
II.4.6 L'alignement du texte.....	100
<b>II.5 LA COULEUR ET LE FOND .....</b>	<b>106</b>
II.5.1 La couleur du texte .....	106
II.5.2 La couleur de fond .....	110
II.5.3 L'image de fond .....	111
II.5.4 L'opacité d'un élément.....	123
II.5.5 La notation RGBa .....	124
<b>II.6 LES BORDURES ET LES OMBRES .....</b>	<b>126</b>
II.6.1 Bordures standards .....	126
II.6.2 Les bordures arrondies.....	129
II.6.3 Les bordures imagées .....	131
II.6.4 Les ombres .....	134
<b>II.7 PSEUDOS CLASSES ET PSEUDO ELEMENTS.....</b>	<b>137</b>
II.7.1 Pseudo-classes dynamiques (Aspect dynamique).....	137
II.7.2 Pseudo-classe descendante .....	142
II.7.3 Les pseudo-éléments de texte .....	142
II.7.4 Les pseudos éléments : after et : before .....	145
<b>II.8 LES MARQUEURS DES LISTES.....</b>	<b>147</b>
II.8.1 La propriété list-style-type .....	147
II.8.2 La propriété list-style-image .....	153

II.8.3 La propriété list-style-position .....	154
II.9 LES MODELES DE BOITES .....	155
II.9.1 Introduction .....	155
II.9.2 Les dimensions.....	160
II.9.3 Les marges .....	166
II.9.4 Centrer les blocs .....	171
II.9.5 Dépassement du texte d'un bloc .....	173
III.    POSITIONNEMENT ET TRANSFORMATION EN CSS .....	179
III.1 LE POSITIONNEMENT FLOTTANT .....	179
III.2 POSITIONNEMENT AVEC LA PROPRIETE <i>Display</i> .....	189
III.3 POSITIONNEMENT AVEC FLEXBOX.....	198
III.3.1 Généralités.....	198
III.3.2 Propriétés liées a display : flex.....	203
III.4 LE POSITIONNEMENT ABSOLU .....	216
III.5 LE POSITIONNEMENT FIXE.....	222
III.5 POSITIONNEMENT RELATIF .....	225
III.6 POSITIONNEMENT MULTI COLONNE .....	227
III.7 LA TRANSFORMATION EN CSS .....	230
III.7.1 La rotation.....	230
III.7.2 La déformation oblique .....	232
III.7.3 Le déplacement .....	234
III.7.4 Le curseur de la souris .....	237
IV.    LES TABLEAUX .....	240
IV.1 Création d'un tableau .....	240
IV.2 Tableaux comportant des cellules vides.....	245
IV.3 La ligne de l'en-tête.....	247
IV.4 Le titre du tableau .....	248
IV.5 Diviser un gros tableau.....	250
IV.6 Fusion des cellules d'un tableau .....	253
IV.6.1 Fusion de colonnes.....	254
IV.6.2 Fusion de lignes .....	258
V.    LA NUMEROTATION AUTOMATIQUE AVEC CSS .....	262
VI.    COMPLEMENTS .....	266
VI.1 LES CARACTÈRES SPECIAUX EN HTML.....	266
VI.2 INSERTION DU CODE SOURCE DANS UNE PAGE WEB.....	267
VI.2.1 Insertion du code source.....	267

VI.2.2 Coloration syntaxique du code source .....	269
VI.2.3 Quelques plugins de Prism.....	275
<b>VII.    LES FORMULAIRES .....</b>	<b>278</b>
VII.1 Zone de texte monoligne .....	279
VII.2 Zone de texte avec une liste de suggestions .....	282
VII.3 Zone de mot de passe .....	283
VII.4 Zone de texte multiligne .....	284
VII.5 Zones de texte enrichies .....	287
VII.5.1 L'email .....	287
VII.5.2 Une URL .....	288
VII.5.3 Numéro de téléphone.....	288
VII.5.4 Nombre.....	288
VII.5.5 Un curseur .....	289
VII.5.6 La couleur .....	291
VII.5.7 Date .....	292
VII.5.8 Recherche .....	293
VII.6 Les cases à cocher.....	293
VII.7 Les zones d'options .....	294
VII.8 Les listes déroulantes.....	295
VII.8.1 Généralités .....	295
VII.8.2 Options de la liste.....	300
VII.9 Le transfert des fichiers.....	303
VII.10 Regrouper les champs .....	305
VII.11 Compléments sur les champs .....	307
VII.12 Le bouton d'envoi .....	308

# I. NOTIONS DE BASE EN HTML

## I.1 INTRODUCTION

Aujourd'hui il y a pleins des sites web dans le monde. Un site web est un ensemble de pages web interconnectés. Pour visiter ou consulter un site web, on a besoin d'un navigateur web, les plus répandus sont Mozilla Firefox, Opera, Internet explorer, safari, Google chrome...

Pour créer un site web, on utilise le langage html (le seul langage indispensable pour la création des sites web). Css s'ajoute dans la conception de sites web comme complément du html.

- HTML (HyperText Markup Language) s'occupe du contenu et organisation du site : pour insérer les images, textes ; audio, vidéos, titres, liens, paragraphes etc. on le fait grâce au Html. C'est un langage de balisage car il permet d'indiquer la nature des éléments, c'est grâce au HTML qu'on indiquera au navigateur que ceci est un titre, celui-ci une image etc.
- CSS (Cascading Style Sheets) s'occupe de la mise en forme du site : couleur, positionnement, alignement du texte etc...Il gère l'apparence du site.

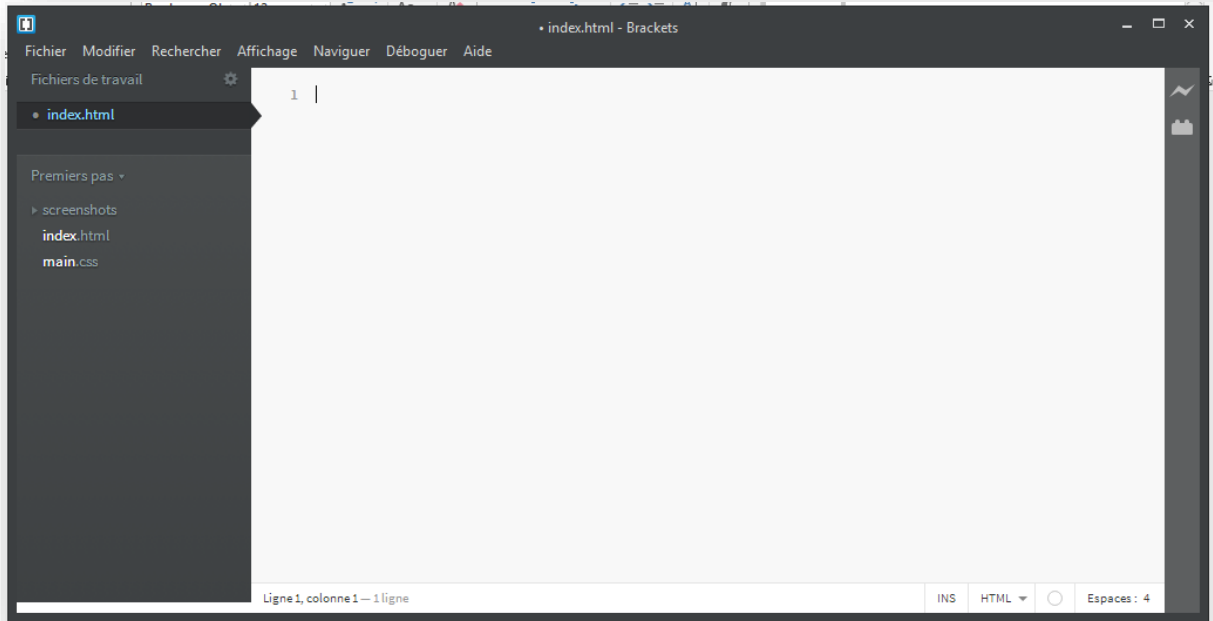
N'utilisez pas le html pour faire le travail de Css, le rôle principal de html est de décrire le contenu : titre, formulaire, paragraphe, sous-titre, lien etc...Et Css s'occupe de la mise en forme de ce contenu : couleur, bordures, positionnement, image et couleur de fond etc.

Les versions actuelles du HTML et du CSS sont HTML5 et CSS3.

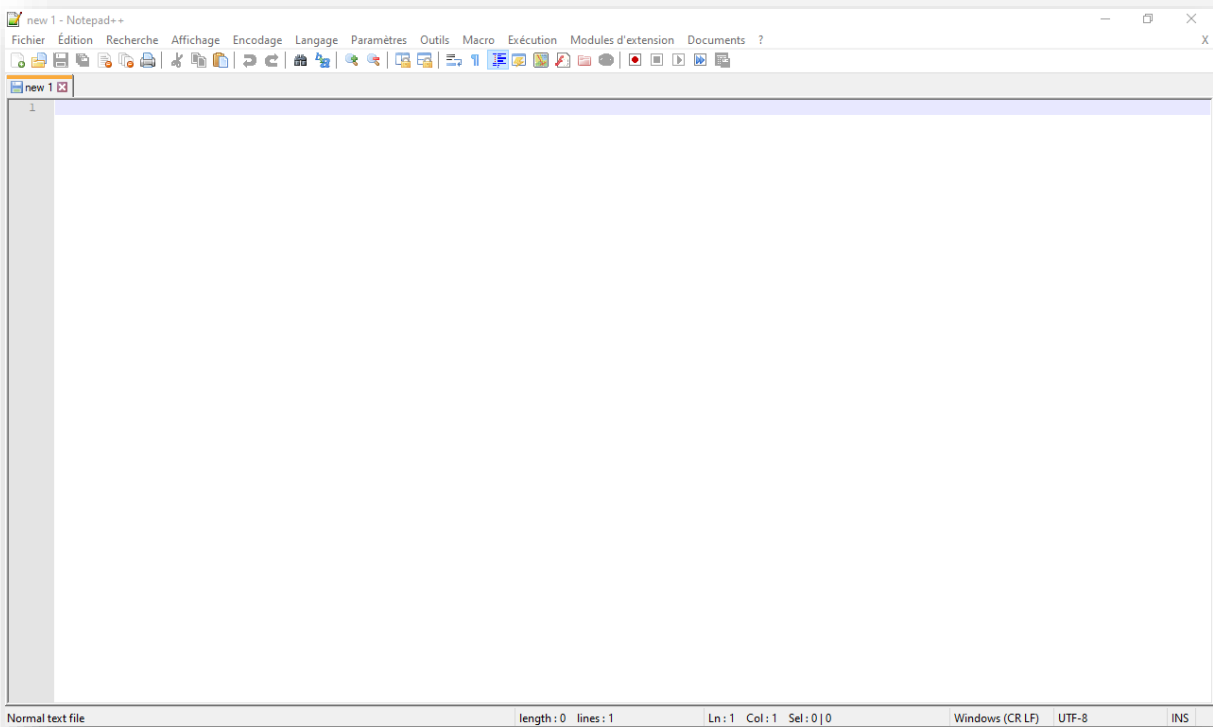
Pour coder un html, on a juste besoin d'un éditeur de texte et des navigateurs pour tester votre site. Les différents navigateurs n'affichent pas le même site exactement de la même façon, c'est pourquoi il est conseillé d'utiliser plusieurs navigateurs web lors de la conception pour se rendre compte de l'affichage de votre site sur les différents navigateurs.

Il existe plusieurs éditeurs de texte pour Windows :

- Notepad++
- Bloc-notes
- Sublime text
- Brackets
- JEdit
- Visual code
- Et tant d'autres

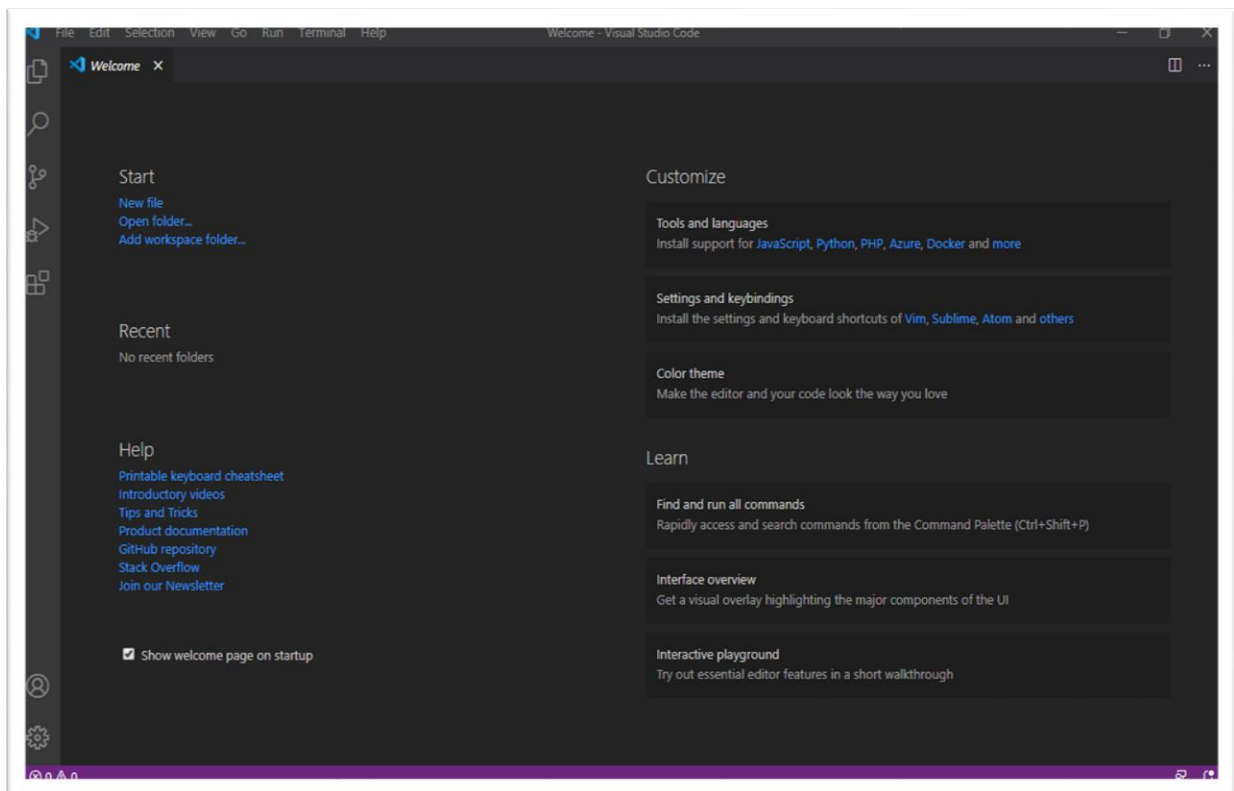


*Editeur Brackets*



*Editeur Notepad++*





*Editeur Visual code*

Choisissez un éditeur selon vos désirs et n'oubliez pas d'installer plusieurs navigateurs pour tester vos pages web.

## I.2 LES BALISES ET LES ATTRIBUTS

### I.2.1 Les balises

Nous avons dit un peu plus haut, que c'est en HTML qu'on insère les différents éléments d'une page web, ainsi pour indiquer la nature d'un élément ou si vous l'accorder, pour donner des ordres au navigateur, on utilise les balises. C'est grâce aux balises qu'on indiquera au navigateur que ça c'est un titre, ceci est une image, ce que qui vient après ceci, doit être sur la ligne suivante, etc.

On distingue deux types des balises : Balises en paires et balises orphelines.

- Balises en paires : comprennent une balise ouvrante et une balise fermante, généralement on met le texte ou le contenu entre les deux.



Elles s'ouvrent, contiennent du texte et se ferment. Elles sont également appelées balises à dimension 1.

- Balises orphelines qui ne comportent une balise fermante, elles sont souvent utilisées pour insérer un élément à un endroit précis de la page, par exemple une image, un saut de ligne etc.

```
<img />|
```

On les appelle aussi balises à dimension 0.

## 1.2.2 Les attributs

Nous venons de voir que les balises permettent de donner la nature des éléments qu'elles encadrent. Prenons le cas d'une image par exemple, on utilise la balise `<img/>` mais à part cette balise, on doit informer au navigateur quelle image on souhaite afficher et que faire ou afficher à la place de l'image si celle-ci n'est pas disponible pour une raison quelconque : C'est dans de cas qu'interviennent les attributs.

Un attribut est une option d'une balise, il permet de donner des informations supplémentaires ou compléter une balise. Il se place après le nom de la balise ouvrante (pour une balise en paire) et après le nom de la balise (pour les balises orphelines). La plupart d'attributs doivent avoir une valeur. Pour le cas de l'image par exemple, c'est l'attribut `src` qui est utilisé pour indiquer le nom de l'image à afficher.

```
<balise attribut="valeur" > </balise>|
```

```
1 <balise attribut="valeur" />
```

Une balise peut avoir plusieurs attributs, il suffit de les séparer par un espace.

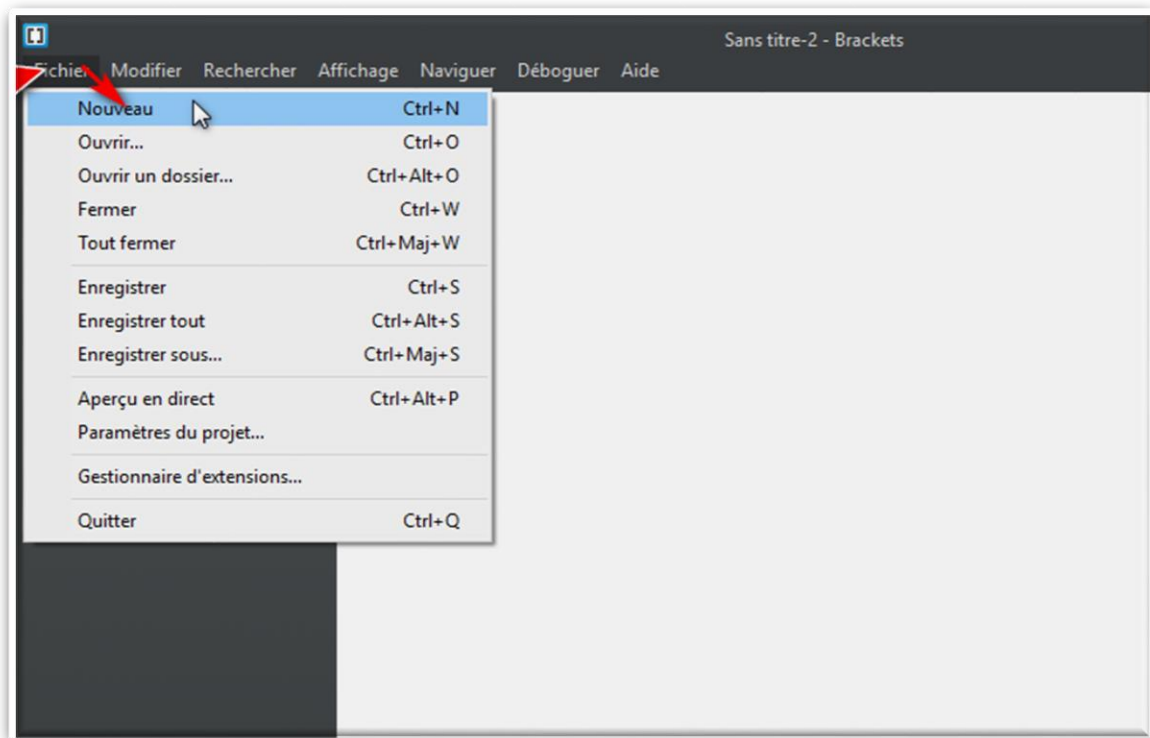
```
<balise attribut1="valeur1" attribut2="Valeur2" > </balise>
```

Pour une balise donnée, il y a des attributs obligatoires (qu'on doit toujours renseigner, le cas de `src` pour la balise `img`) et facultatifs (le cas de `title` pour `img`). Notez aussi que chaque élément de HTML dispose d'une collection

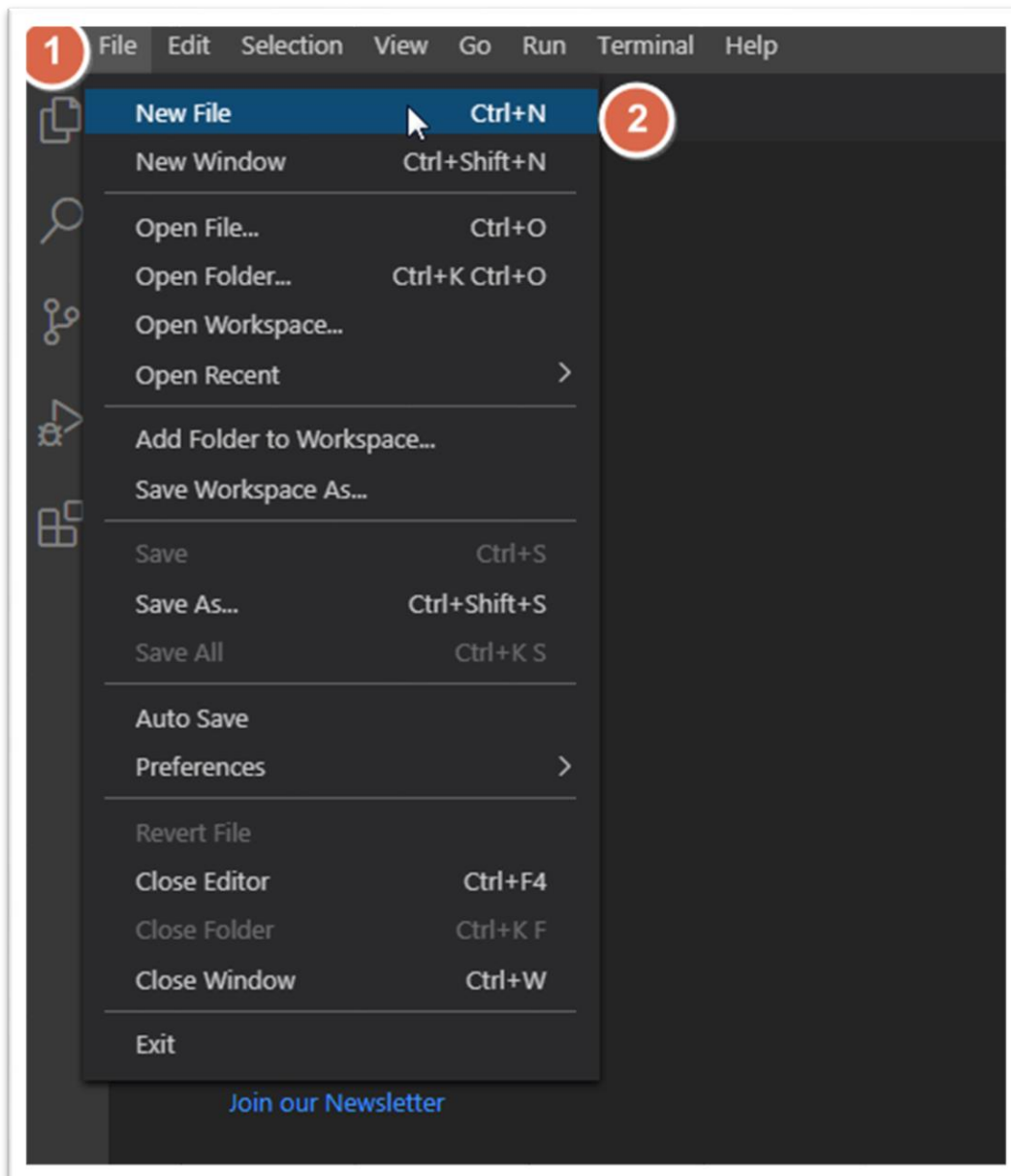
d'attributs qui peuvent être utilisés avec. En d'autres termes, chaque élément HTML a une liste d'attributs qu'il peut accepter. Certains attributs peuvent prendre pour valeur n'importe quel texte, car ils acceptent n'importe quelle valeur. D'autres ne peuvent prendre qu'une valeur parmi une liste de valeurs bien définies.

### I.3 STRUCTURE DE BASE D'UNE PAGE HTML

Pour créer un nouveau projet dans Brackets ou Notepad++, Allez dans le menu « Fichier » puis Nouveau ou la combinaison Ctrl-N.



Si vous utilisez Visual code, rendez-vous dans le menu File, puis cliquez sur New File ou faites simplement la combinaison CTRL+N :



En Informatique, il y a des règles pour chaque langage, HTML ne fait pas exception. Ainsi, toute page écrite en HTML5 doit comporter une certaine structure, un « squelette » qui sera toujours le même.

Toute page html a au minimum le code suivant :

```

<!DOCTYPE html>
<html>

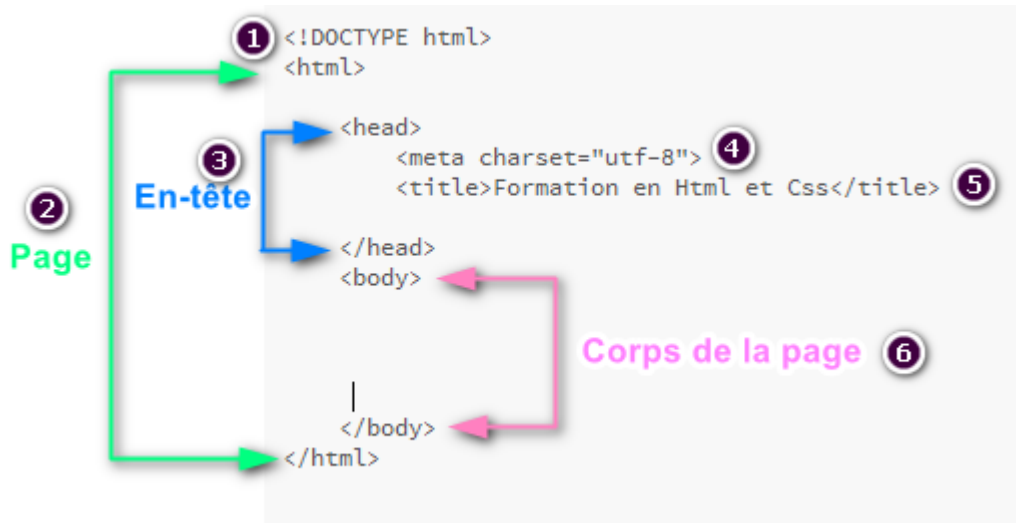
  <head>
    <meta charset="utf-8">
    <title>Formation en Html et Css</title>

  </head>
  <body>

  </body>|
</html>

```

Essayons d'expliquer ce bout de code :



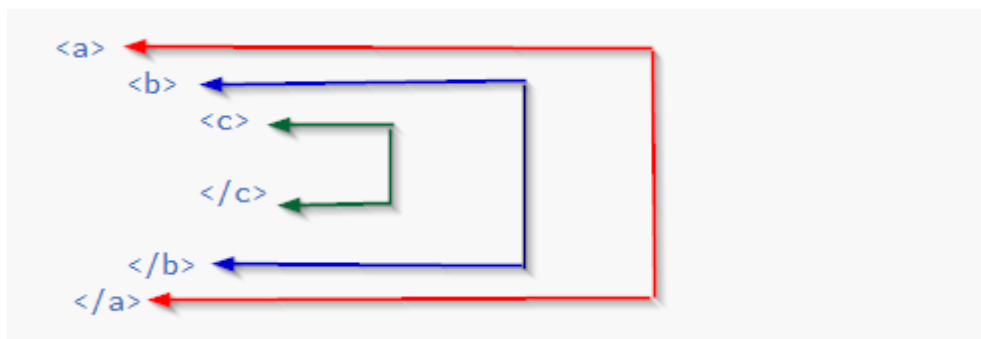
1. DOCTYPE : Le doctype sert à indiquer le type du document, dans notre cas, il indique qu'il s'agit d'une page web html, cette ligne indique aussi que la page a été codée en html5 car les versions antérieures n'étaient pas pareilles. Donc si vous voyez une page web qui commence par cette ligne, comprenez qu'elle a été codée en html5
2. <html> </html> englobe toute la page, elles indiquent le début et la fin de notre page web.
3. <head> </head> désigne l'en-tête de la page web, cette partie donne quelques informations sur la page notamment :
4. <meta...> définit l'encodage utilisé dans votre fichier html.
5. Le titre de la page contenue entre les balises <title></title>. Il ne s'affiche pas sur la page mais au-dessus de celle-ci :



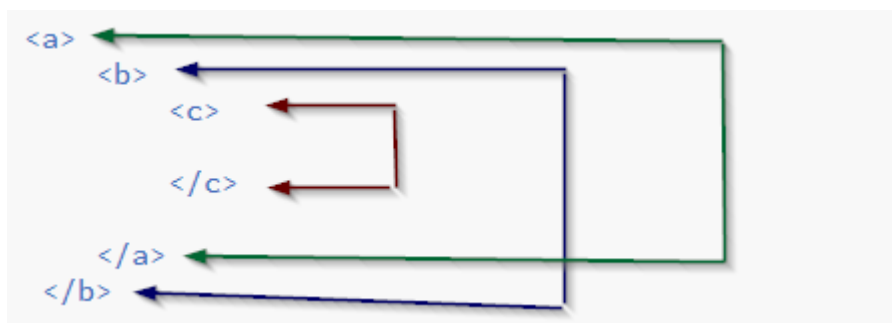
6. `<body>` `</body>` délimite le corps de la page web. C'est dans cette partie que nous écrivons la plupart du code html.

### Remarque

Vous l'avez sans doute remarqué qu'il est possible en HTML d'imbriquer un élément à l'intérieur d'un autre élément. Le cas de meta et title qui se trouvent à l'intérieur de l'élément head. La même chose pour head et body qui se trouvent à l'intérieur de l'élément html. Malgré que cela soit un avantage, il faut faire attention à la fermeture des balises dans le bon ordre. N'oubliez jamais ceci : lorsque les éléments(balises) sont imbriqués les uns dans les autres, le dernier élément ouvert doit être le premier élément à fermer. Les figures suivantes parlent clairement :



Ce code est valide car l'ordre de fermeture est respecté, par contre le suivant n'est pas correct :



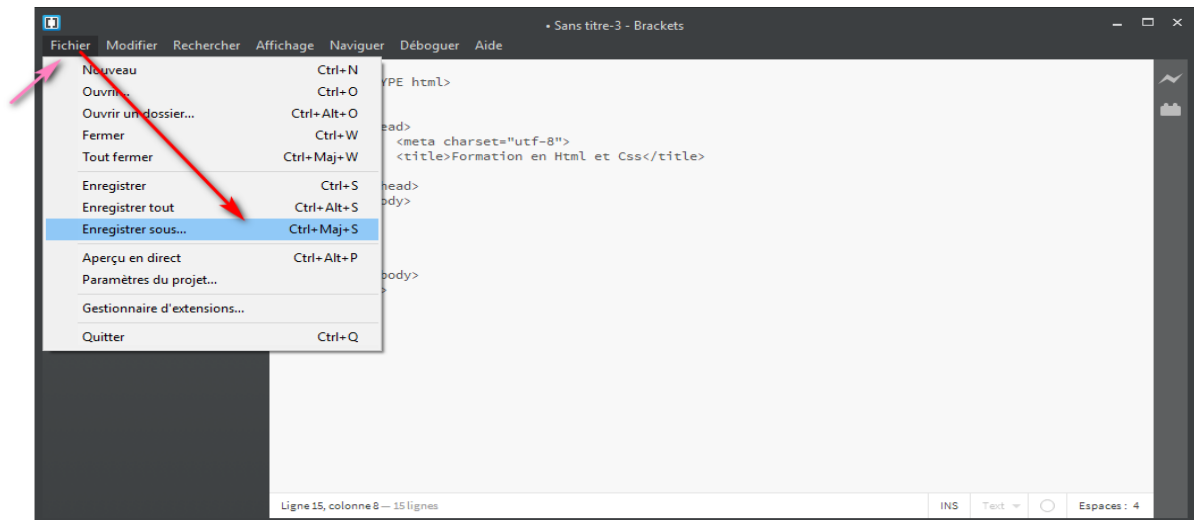
Il y a chevauchement entre les balises a et b : la balise b est contenue dans la balise a et il faut d'abord fermer la balise b avant la balise a.

Pour éviter ces genres d'erreurs, il serait mieux de refermer une balise juste après l'avoir ouverte.

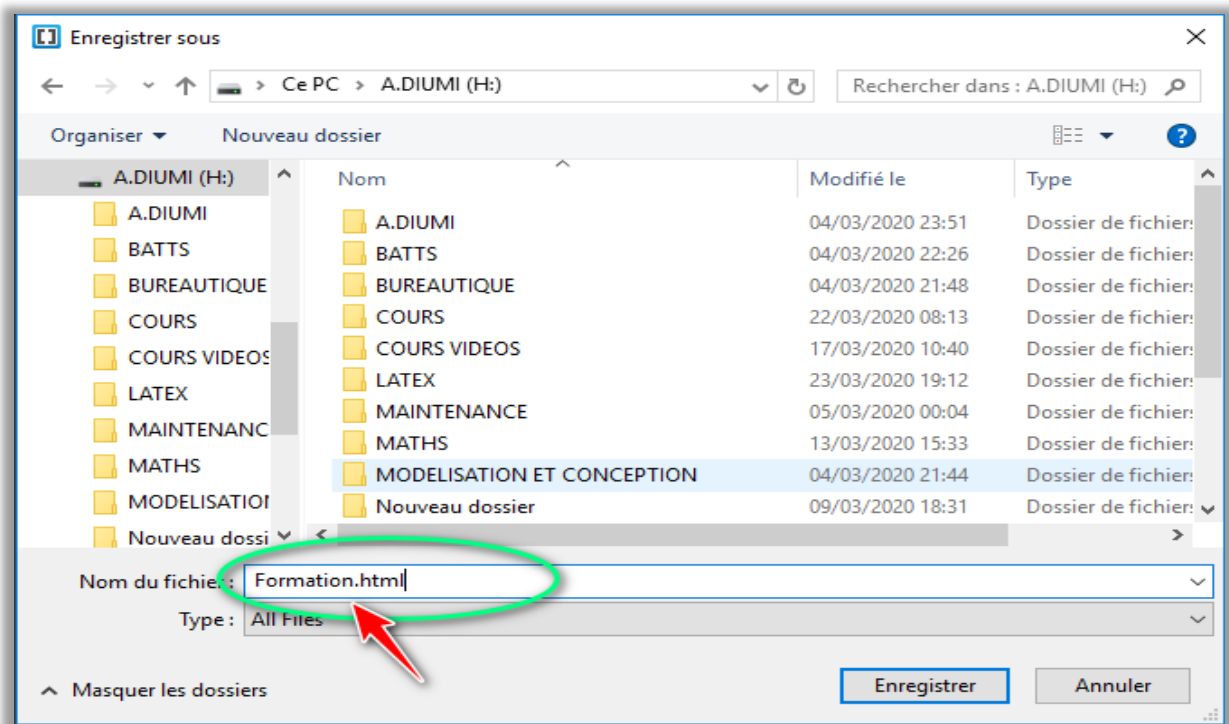
## I.4 ENREGISTREMENT D'UN FICHER HTML

Ecrivez le code ci-haut dans votre éditeur de texte.

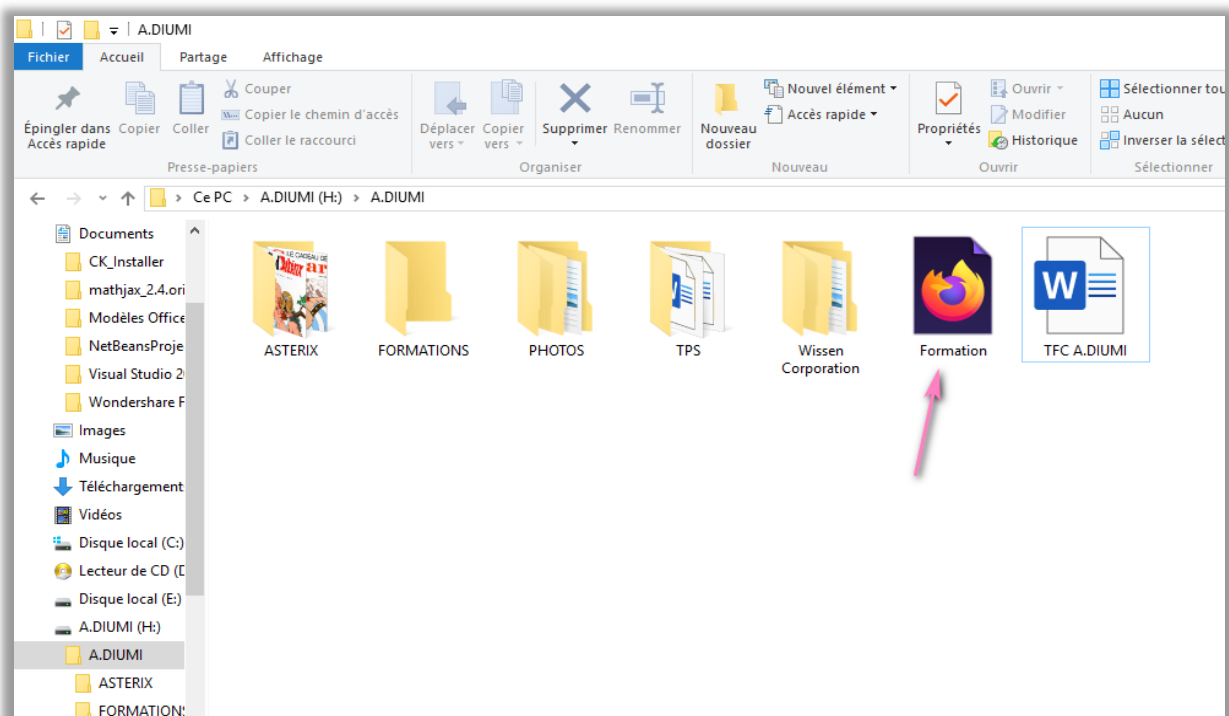
Pour sauvegarder notre fichier html (page web), allez dans le menu Fichier > Enregistrer ou Enregistrement sous (Pour le premier enregistrement les deux font la même chose).



Une fenêtre s'ouvre vous demandant où enregistrer le fichier : choisissez l'emplacement et le nom du fichier sans oublier l'extension html, par exemple Formation.html :



Après l'enregistrement, vous remarquerez que le code sera coloré. Pour exécuter ça, rendez-vous là où vous l'avez enregistré (Explorateur des fichiers) :





L'icône dépend de votre navigateur par défaut, vous avez compris que mon navigateur par défaut est Mozilla Firefox. Et il suffit de double-cliquer sur le fichier et il s'ouvre avec le navigateur.

Essayons d'écrire quelque chose dans notre page.

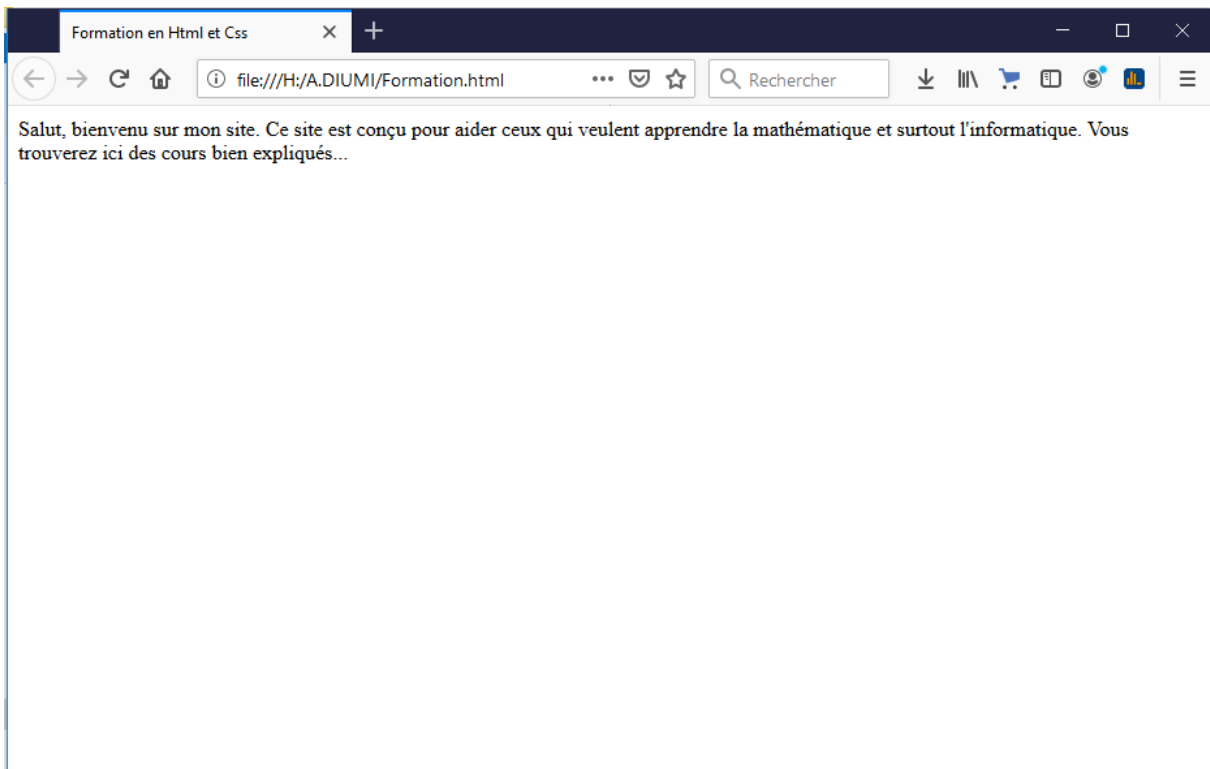
```
<!DOCTYPE html>
<html>

  <head>
    <meta charset="utf-8">
    <title>Formation en Html et Css</title>

  </head>
  <body>
    Salut, bienvenu sur mon site.
    Ce site est conçu pour aider ceux qui veulent apprendre la mathématique et surtout l'informatique.
    Vous trouverez ici des cours bien expliqués...

  </body>
</html>
```

En ouvrant ça, on obtient ceci :



## I.5 Commentaires en HTML

Quand la page devient très longue, on pourra se perdre dans le code HTML, c'est pourquoi il est conseillé d'insérer des commentaires dans votre page HTML. Les commentaires ne seront pas visibles par vos visiteurs (à moins qu'ils affichent le code source de la page).

Pour insérer un commentaire en HTML, on utilise la syntaxe suivante :

```
<!-- ceci est un commentaire -->
```

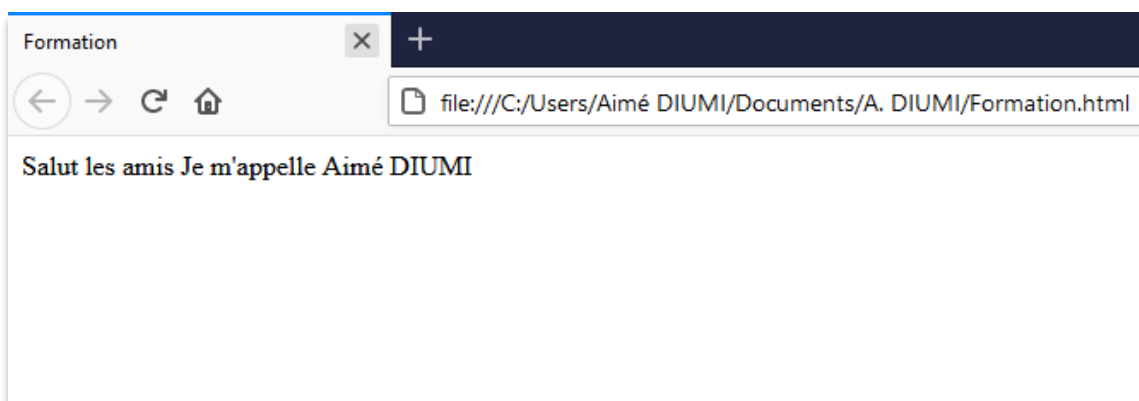
Exemple

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>

  </head>
  <body>
    Salut les amis
    <!-- Ceci est un commentaire, ça me permet de me retrouver dans mon code -->
    Je m'appelle Aimé DIUMI

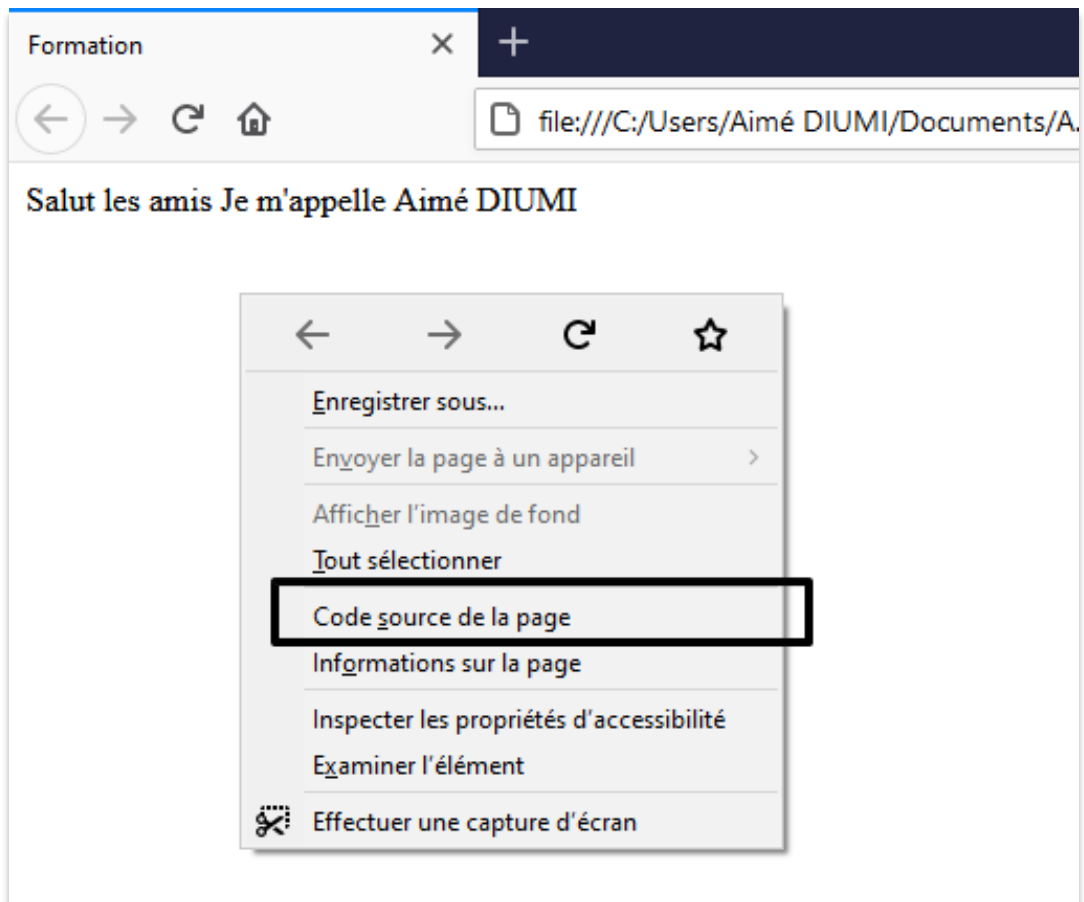
  </body>
</html>
```

Résultat :

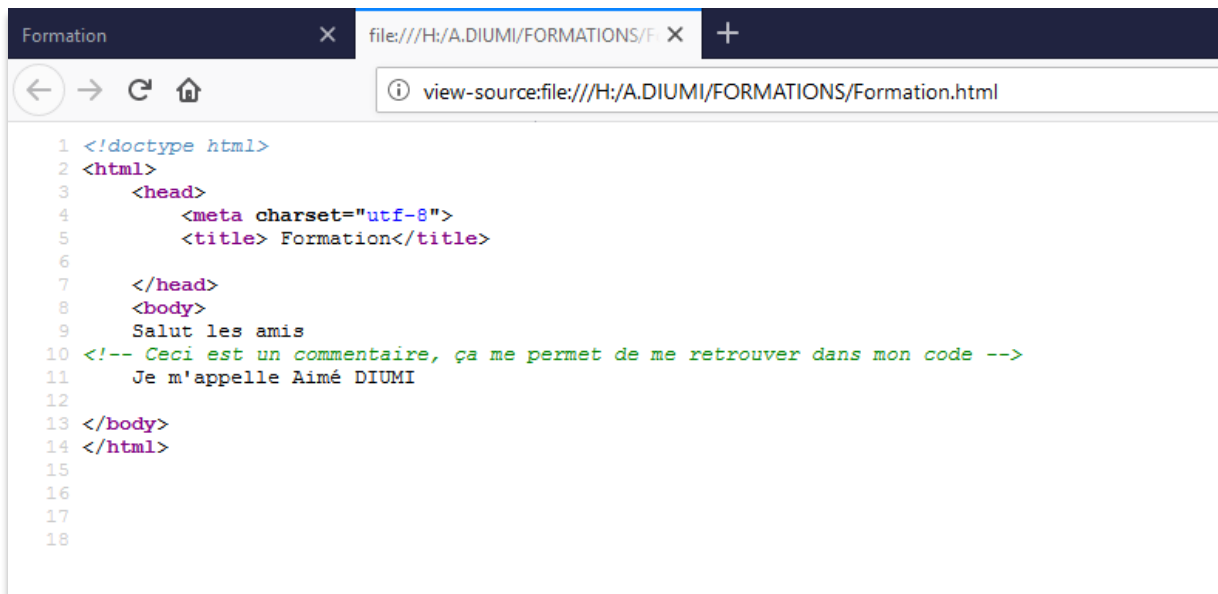


Les commentaires ne sont pas affichés.

Attention à ne jamais mettre d'informations sensibles en commentaire, comme des mots de passe par exemple. En effet, rappelez-vous que tout le monde peut avoir accès au code source de votre page, et donc à votre code HTML. Par exemple, dans Mozilla Firefox, quand vous êtes dans n'importe quelle page web de n'importe quel site, faites un clic droit sur la page :



Cliquez ensuite sur Code source de la page et le code source apparaîtra comme le montre la figure suivante :



## I.6.ORGANISATION DU TEXTE

### I.6.1 Le retour à la ligne

L'exemple suivant :

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>

  </head>
  <body>
    Salut les amis,
    Nous sommes Wissen corporation.

  </body>
</html>
```

produit le résultat suivant :



J'avais écrit le texte sur deux lignes, mais à l'affichage, les deux phrases sont sur la même ligne. Cela est logique car nous avons dit qu'il faut utiliser des balises pour donner des ordres au navigateur. Pour indiquer au navigateur de faire un retour à la ligne, on utilise la balise `<br/>` (break)

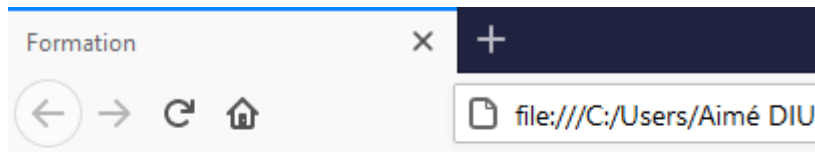
Reprenons le même exemple mais cette fois-ci avec `br`

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>

  </head>
  <body>
```

```
Salut les amis, <br/> Nous sommes Wissen corporation.  
</body>  
</html>
```

Résultat :



Salut les amis,  
Nous sommes Wissen corporation.

Dès que la navigateur rencontre <br/>, il comprend que tout ce qui vient après (que ça soit sur la même ligne ou pas) doit s'afficher à la ligne suivante.

## 1.6.2 Les paragraphes

En html, un paragraphe est délimité par les balises <p> </p> (paragraph)

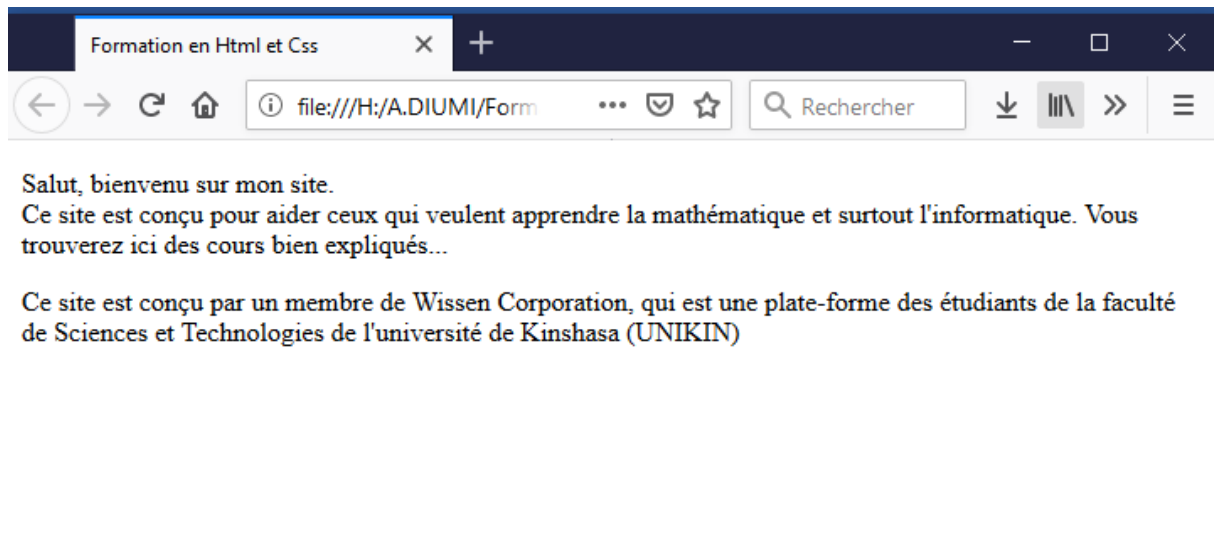


Chaque paragraphe crée un retour à la ligne automatique.

Exemple :

```
1 <!DOCTYPE html>  
2 <html>  
3  
4 <head>  
5 <meta charset="utf-8">  
6 <title>Formation en Html et Css</title>  
7  
8 </head>  
9 <body>  
10 <p> Salut, bienvenu sur mon site.<br/>  
11 Ce site est conçu pour aider ceux qui veulent apprendre la mathématique et surtout l'informatique.  
12 Vous trouverez ici des cours bien expliqués...</p>  
13  
14 <p>  
15 Ce site est conçu par un membre de Wissen Corporation, qui est une plate-forme des  
16 étudiants de la faculté de Sciences et Technologies de l'université de Kinshasa (UNIKIN) en  
17 République démocratique du Congo (R.D.C.)  
18 </p>  
19 </body>  
</html>
```

Ça produit le résultat suivant :



### 1.6.3 Les titres

Il est possible d'utiliser six niveaux de titre différents, cela s'explique par le fait que tous les titres n'ont pas le même niveau d'importance. Il y a des titres principaux, titres secondaires etc.

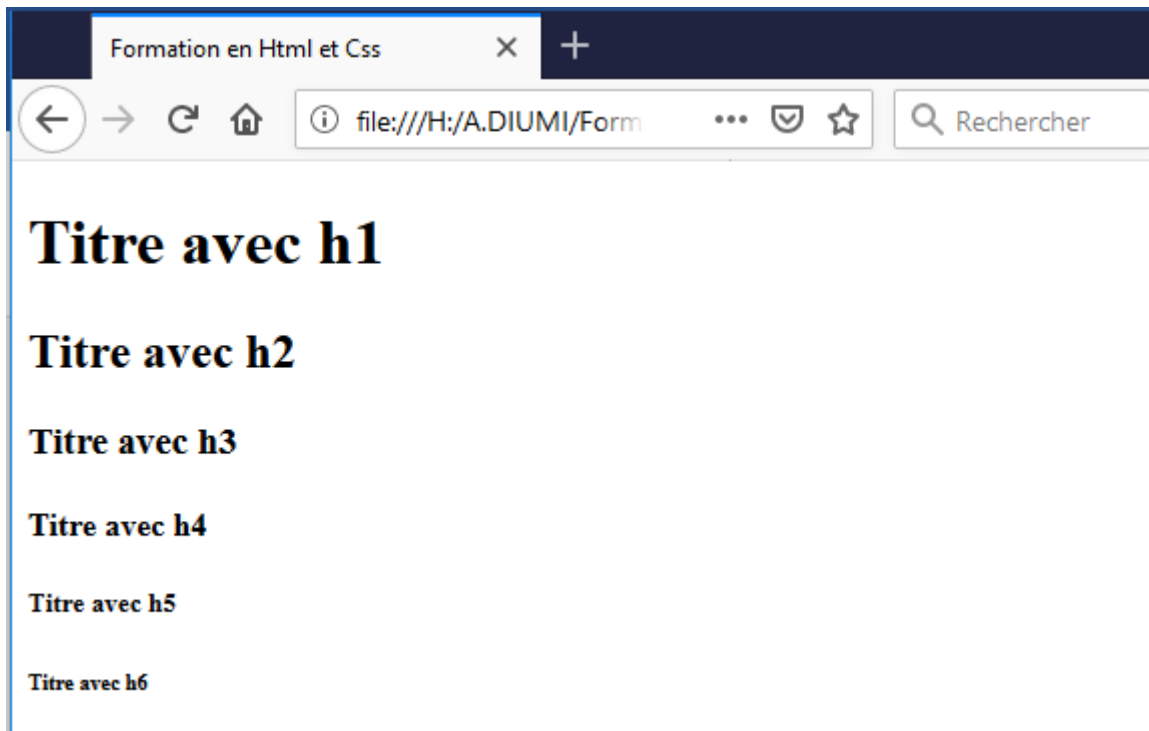
Les 6 balises sont :

- `<h1> </h1>`
- `<h2> </h2>`
- `<h3> </h3>`
- `<h4> </h4>`
- `<h5> </h5>`
- `<h6> </h6>`

Essayons de voir la différence entre ces six titres :

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Formation en Html et Css</title>
6   </head>
7   <body>
8
9     <h1> Titre avec h1</h1>
10    <h2>Titre avec h2</h2>
11    <h3>Titre avec h3</h3>
12    <h4>Titre avec h4</h4>
13    <h5>Titre avec h5</h5>
14    <h6>Titre avec h6</h6>
15    </body>
16 </html>++|
```

On peut facilement voir la différence :



Utilisez ces balises non par rapport au niveau de l'affichage mais par rapport au niveau de l'importance du titre à insérer car la taille de l'affichage peut être modifiée avec CSS.

## 1.6.4. Les éléments em, mark et strong

### a) La balise <strong>

La balise <strong> est utilisée pour indiquer aux moteurs de recherche ou au navigateur que le contenu qu'elle encadre est très important afin que celui-ci soit traité avec plus d'importance.

Exemple

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
  </head>
  <body>
    <h2> A propos du concepteur </h2>
    <p>ce site est conçu par <strong> Aimé DIUMI </Strong>, co-
fondateur et coordonnateur de Wissen Corporation. </p>
  </body>
</html>
```

produit ceci :



Le navigateur a mis le contenu de la balise `<strong>` (Aimé DIUMI) automatiquement en gras. Cela ne veut pas dire qu'on utilise la balise `strong` pour mettre un texte en gras !!!! Non, non et non, elle permet seulement d'indiquer au navigateur que ce contenu est très important et dans cet exemple le navigateur a préféré l'afficher en gras pour faire ressortir visuellement son importance.

### *b. La balise `<em>`*

La balise `<em>`, pour emphase (« emphase » en français) sert elle aussi à signifier qu'un contenu est important, mais moins important tout de même qu'un contenu entre des balises `<strong>`.

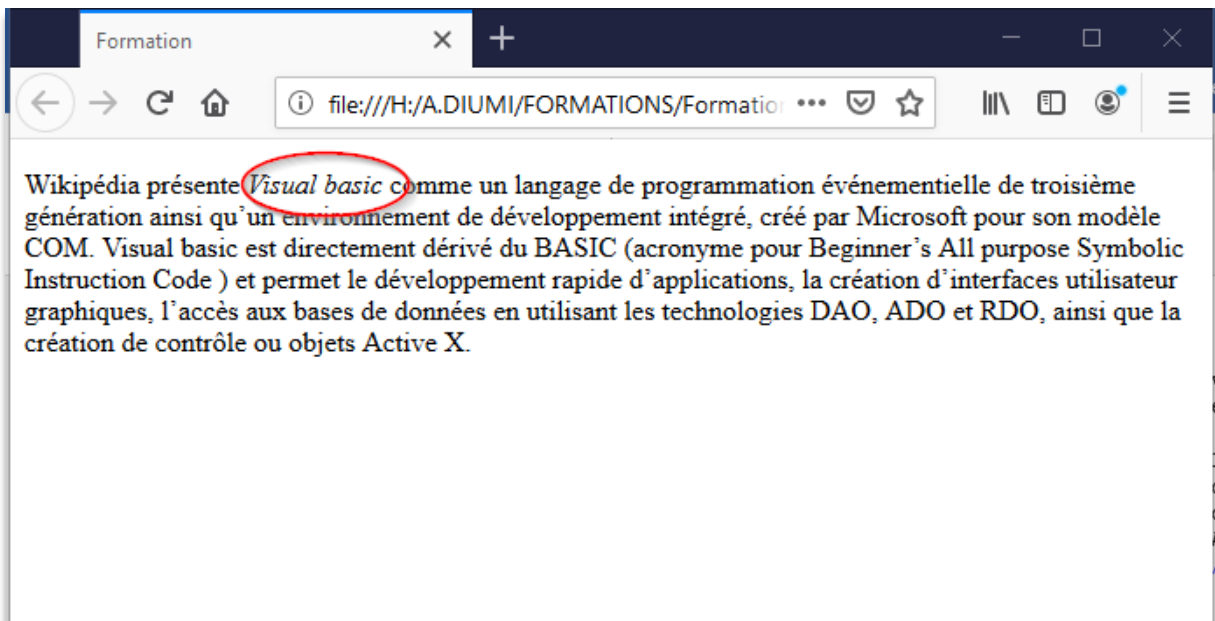
Exemple

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>

  </head>
  <body>
    <p>Wikipédia présente <em>Visual basic </em> comme un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, créé par Microsoft pour son modèle COM. Visual basic est directement dérivé du BASIC (acronyme pour Beginner's All purpose Symbolic Instruction Code ) et permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôle ou objets Active X. </p>
  </body>
</html>
```



Résultat :



Le navigateur a mis le contenu de la balise `<em>` en italique, mais n'utilisez pas la balise `<em>` pour mettre du texte en italique.

### c. La balise `<mark>`

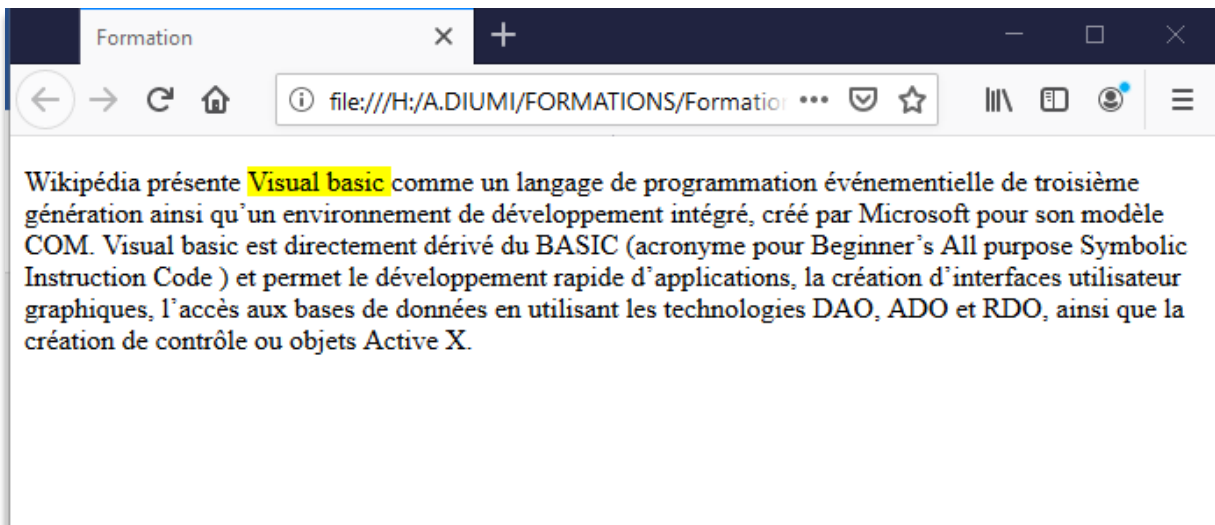
La balise `<mark>` permet de faire ressortir visuellement le contenu qu'elle encadre.

Exemple

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>

  </head>
  <body>
    <p>Wikipédia présente <mark>Visual basic </mark> comme un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, créé par Microsoft pour son modèle COM. Visual basic est directement dérivé du BASIC (acronyme pour Beginner's All purpose Symbolic Instruction Code) et permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôle ou objets Active X.
  </p>
</body>
</html>
```

Résultat :



Parmi les intérêts liés à l'utilisation de ces balises, surtout les deux premières, nous pouvons citer avant tout l'optimisation du référencement de votre site.

## 1.6.5 Les listes

Il existe trois types de listes :

- Listes non ordonnées ou à puces ;
- Listes ordonnées ou numérotées et
- Listes de définitions

### *a) Listes non ordonnées*

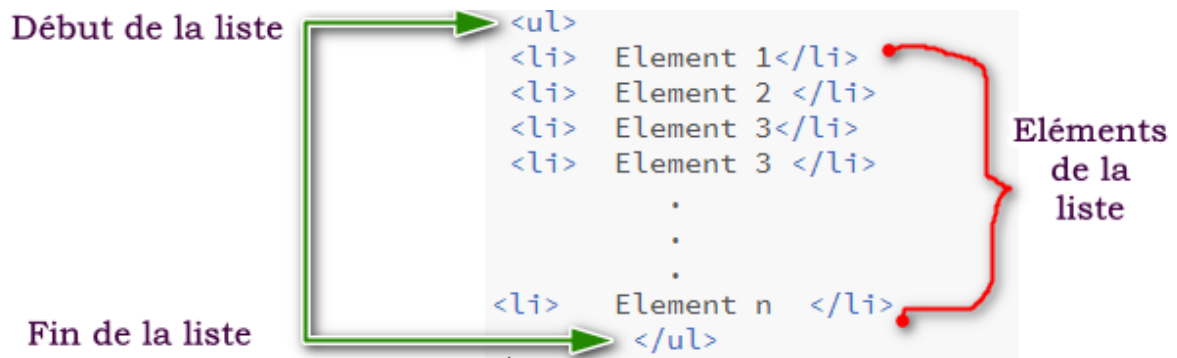
Une liste non ordonnée est une liste dans laquelle les éléments ne sont pas numérotés. Voici l'exemple d'une liste non ordonnée :

- Mathématiques
- Programmation
- Programmation web
- Bureautique

Elle n'est pas numérotée.

Pour créer une telle liste, on utilise la balise `<ul>` `</ul>` (abréviation de unordered list) et chaque élément de la liste doit être entouré des balises `<li>` `</li>` (pour list item).

Schématiquement :



Pour la liste

- Mathématiques
- Programmation
- Programmation web
- Bureautique

On doit écrire le code suivant :

```
<ul>
<li> Mathématiques</li>
<li> Programmation </li>
<li> Programmation web</li>
<li> Bureautique </li>
</ul>
```

J'ai mis seulement le code de la liste, insérez ça dans body.

Il est aussi possible de modifier la puce qui s'affiche à côté des éléments, cela se fait grâce à l'attribut type qui peut prendre une des valeurs suivantes : square, circle ou disc.

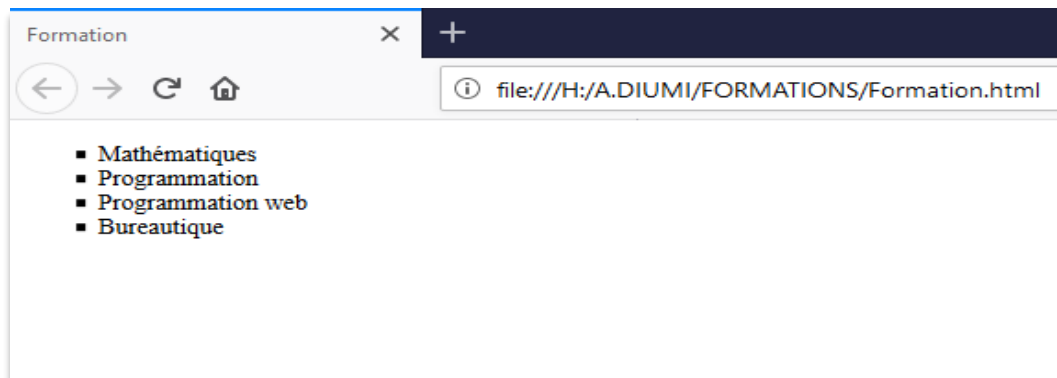
Exemple

```
<ul type="square">

  <li> Mathématiques</li>
  <li> Programmation</li>
  <li> Programmation web </li>
  <li> Bureautique </li>

</ul>
```

Résultat :



### *b) Les listes ordonnées :*

Voici à quoi ressemble une liste ordonnée :

1. Aimé DIUMI
2. Kleine KAMBALA
3. Julie DIKOLO
4. Moise DIKOLO

Pour créer une telle liste, le principe est le même sauf qu'on utilise `<ol>` `</ol>` (pour ordered list) au lieu de `<ul>` :

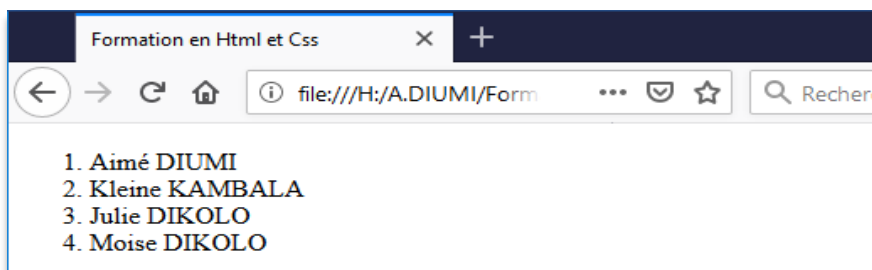
`<ol>` `</ol>` : délimite la liste ordonnée

`<li>` `</li>` : délimite chaque élément de la liste

Exemple

```
<ol>
  <li> Aimé DIUMI</li>
  <li>Kleine KAMBALA</li>
  <li> Julie DIKOLO</li>
  <li>Moise DIKOLO</li>
</ol>
```

Le résultat est :



## Autres paramètres sur les listes ordonnées

### b1) L'attribut start

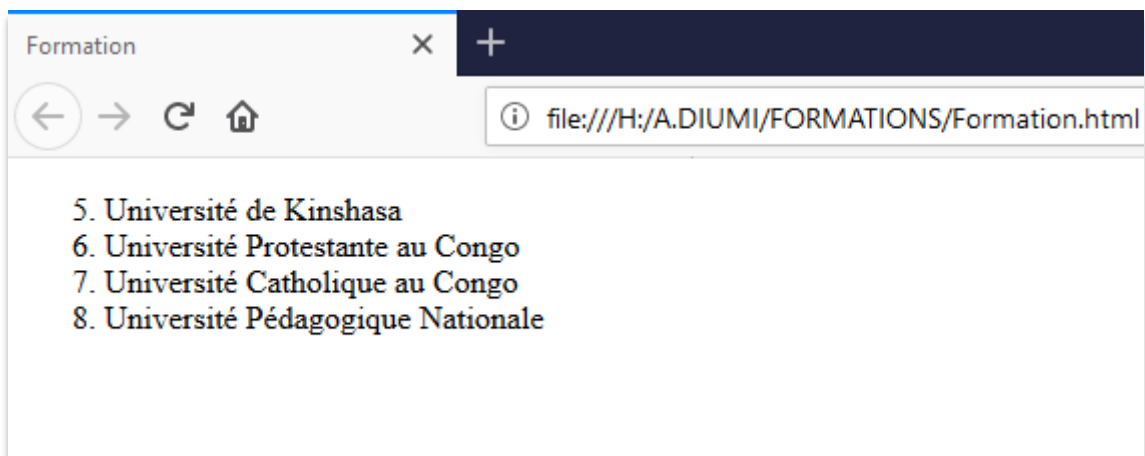
Par défaut, la numérotation comme par 1, sachez qu'il est à fait possible de commencer à compter à un chiffre différent de 1, pour le faire, on utilise l'attribut start.

#### Exemple

Je veux que les numéros des éléments commencent par 5

```
<ol start="5">
  <li> Université de Kinshasa </li>
  <li> Université Protestante au Congo </li>
  <li> Université Catholique au Congo </li>
  <li> Université Pédagogique Nationale </li>
</ol>
```

Résultat :

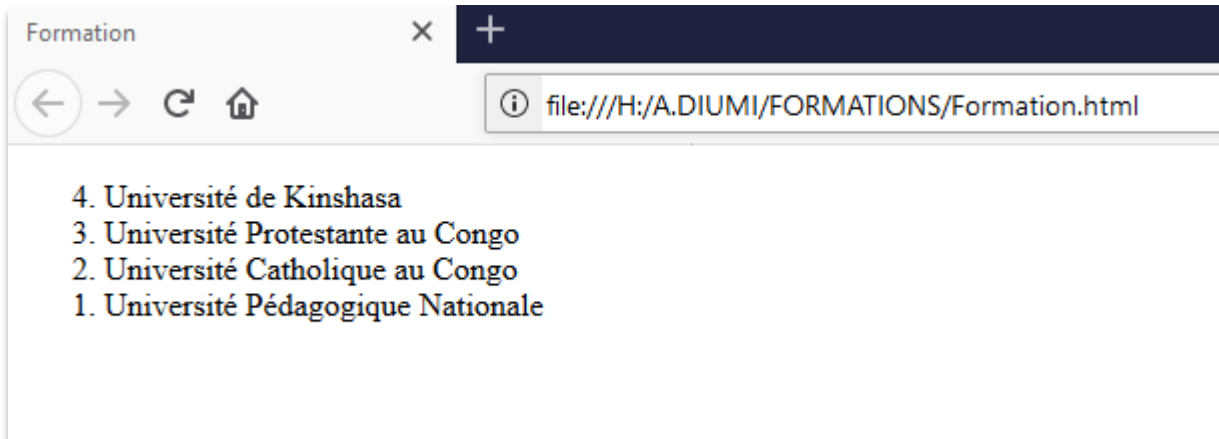


### b2) L'attribut reversed

Il existe aussi l'attribut **reversed** qui permet une numérotation descendante, c'est-à-dire du plus grand numéro au plus petit comme le montre l'exemple suivant :

```
<ol reversed>
  <li> Université de Kinshasa </li>
  <li> Université Protestante au Congo </li>
  <li> Université Catholique au Congo </li>
  <li> Université Pédagogique Nationale </li>
</ol>
```

Résultat :



### b3) L'attribut type

Cet attribut permet de modifier le type de numérotation des éléments de la liste. Elle peut prendre les valeurs suivantes : A, a, I, i ou 1.

A : la numérotation se fera avec les lettres majuscules de l'alphabet

a : avec les lettres minuscules de l'alphabet

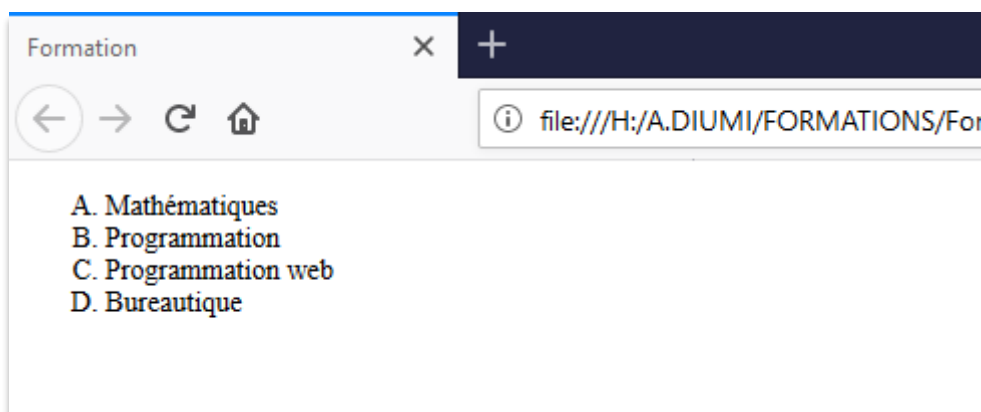
I ou i : Chiffres romains

1 (valeur par défaut) : numérotation décimale

Exemples

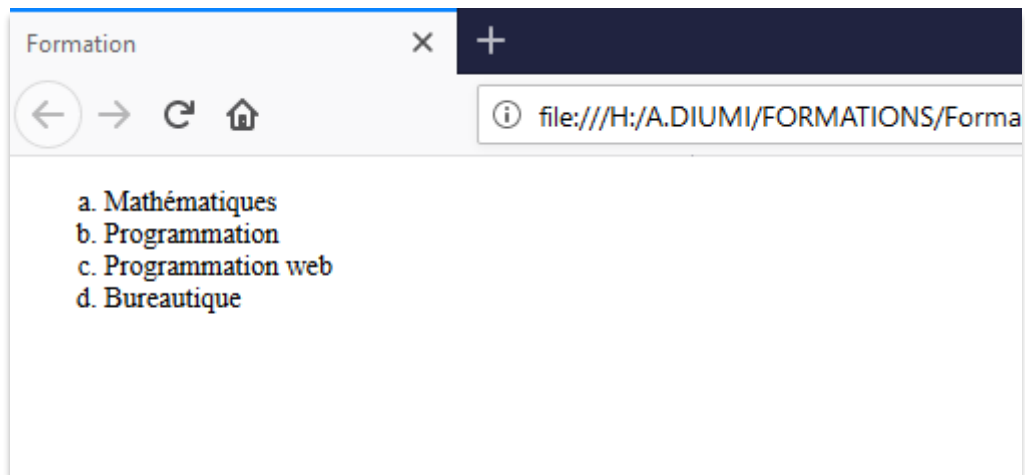
```
<ol type="A">  
  <li> Mathématiques</li>  
  <li> Programmation</li>  
  <li> Programmation web </li>  
  <li> Bureautique </li>  
</ol>
```

Résultat :



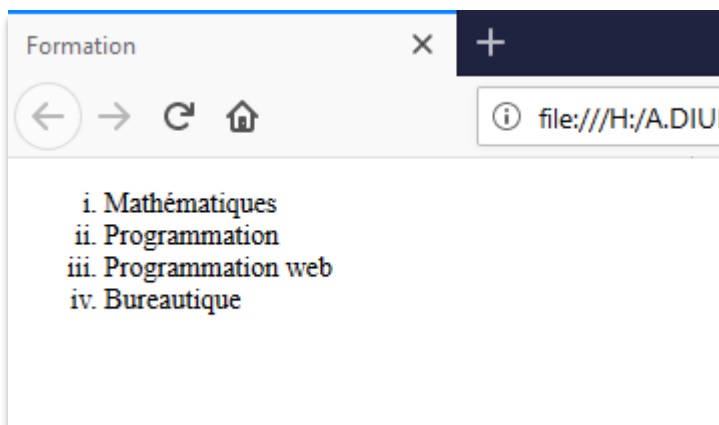
```
<ol type="a">
  <li> Mathématiques</li>
  <li> Programmation</li>
  <li> Programmation web </li>
  <li> Bureautique </li>
</ol>
```

Résultat :



```
<ol type="i">
  <li> Mathématiques</li>
  <li> Programmation</li>
  <li> Programmation web </li>
  <li> Bureautique </li>
</ol>
```

Résultat :



On peut aussi définir le type de puces ou de numérotation grâce à CSS, cela est expliqué un peu plus loin dans ce livre.

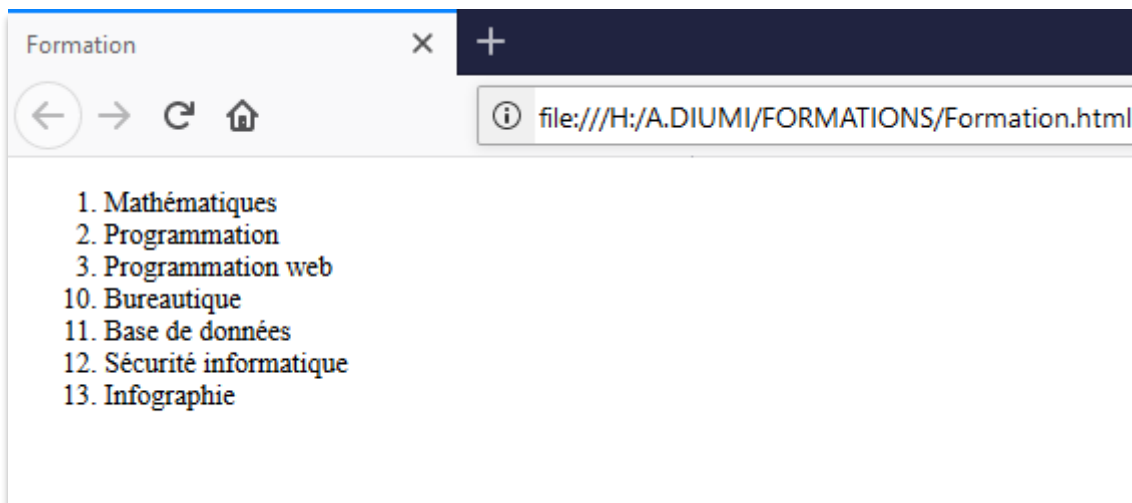
#### b4) L'attribut value

Cet attribut permet de réinitialiser la numérotation des éléments.

Exemple

```
<ol>
  <li> Mathématiques</li>
  <li> Programmation</li>
  <li> Programmation web </li>
  <li value="10"> Bureautique </li>
  <li> Base de données </li>
  <li> Sécurité informatique </li>
  <li> Infographie</li>
</ol>
```

Résultat :



A partir de « Bureautique », les éléments sont numérotés à partir de 10 car nous avons réinitialisé la numérotation avec l'attribut **value="10"**.

#### c) Listes de définitions

Elles sont utilisées pour définir un terme. Pour créer une liste de définition, il va nous falloir utiliser la balise <dl> (pour definition list), la balise <dt> (pour definition term) et l'élément <dd> pour la définition en soi.

Vous devez toujours placer l'élément dt avant l'élément dd, c'est-à-dire le terme à définir avant sa définition.



## Exemple

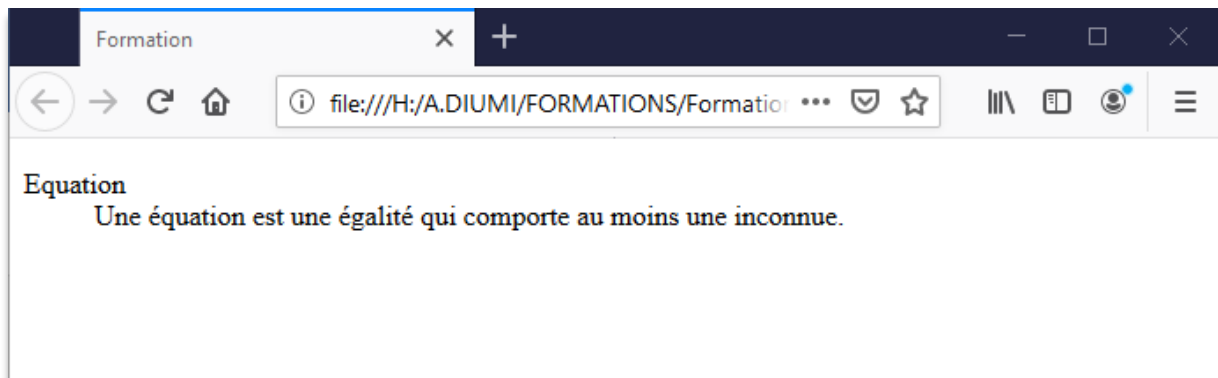
```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>

  </head>
  <body>

  <dl>
    <dt>Equation </dt>
<dd>Une équation est une égalité qui comporte au moins une inconnue.</dd>
  </dl>

</body>
</html>
```

Résultat :



### *Remarque : listes imbriquées*

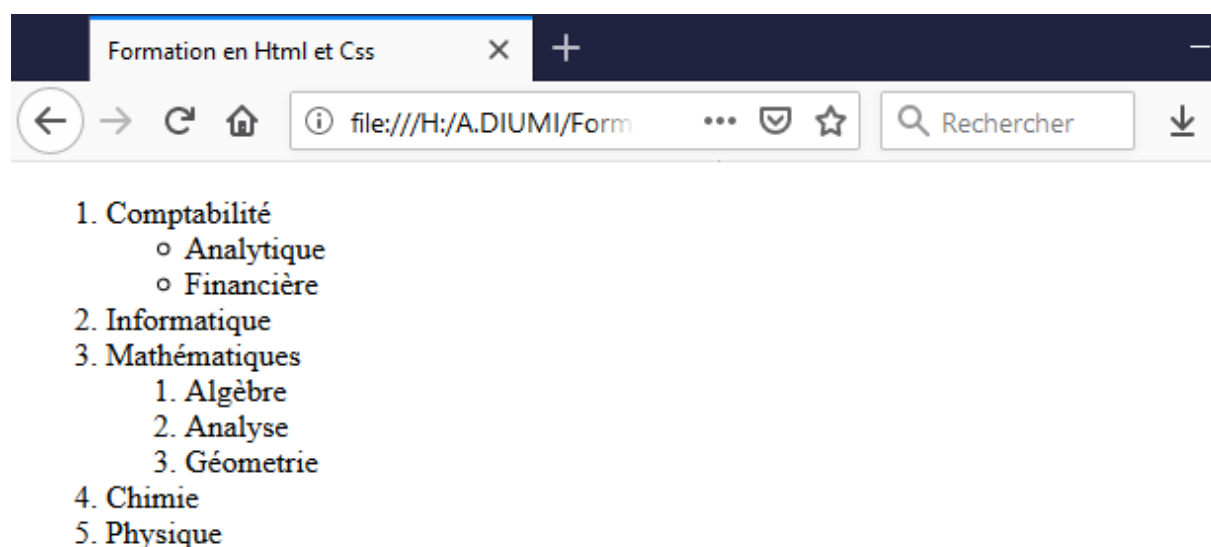
Sachez qu'il est possible d'imbriquer une liste dans une autre liste, il suffit d'ajouter une balise `<ul></ul>` ou `<ol> </ol>` à l'intérieur d'une balise `<li> </li>`

## Exemple

Tester le code suivant : (j'ai écrit seulement le code relatif à la liste, mettez-le dans `<body> </body>`. Il m'arrivera souvent de ne pas mettre le code entier, mais de votre côté n'oubliez pas qu'on écrit tout ça dans body.

```
<ol>
<li>Comptabilité
  <ul>
    <li> Analytique</li>
    <li> Financière</li>
  </ul>
</li>
<li> Informatique</li>
<li>Mathématiques
  <ol>
    <li>Algèbre</li>
    <li> Analyse</li>
    <li> Géométrie</li>
  </ol>
</li>
<li>Chimie</li>|
<li> Physique</li>
</ol>
```

Voici le résultat :



Essayons de réviser tout ce qu'on a étudié jusque-là :

<p> </p> : délimite un paragraphe

<br/> : fait un retour à la ligne

<h1> </h1> jusqu'à <h6> </h6> : délimite un titre

<ul> </ul> : délimite une liste non ordonnée

<ol> </ol> délimite une liste non ordonnée

<li> </li> : délimite un élément de la liste.

Voici un exemple qui contient les éléments vus jusqu'ici :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Formation en Html et Css</title>
  </head>
  <body>
<h1> WISSEN CORPORATION</h1>
    <h2> Présentation</h2>
    <p> Wissen Corporation est un groupe d'étudiants de la faculté de Sciences et Technologies de l'université de Kinshasa (UNIKIN) en République démocratique du Congo (R.D.C.)<br/>
    Le premier objectif poursuivi par ce groupe est le partage de la connaissance , car la connaissance c'est comme un chandelier et sa lumière ne diminue pas s'il allume un autre chandelier</p>
    <h2> Formations organisées</h2>
<p>Nous organisons des formations suivantes: </p>
    <ol>
<li> Programmation
      <ul>
        <li> Visual basic</li>
        <li> C#</li>
        <li> Java</li>
        <li> Python</li>
        <li> langage C</li>
      </ul>
</li>
<li>Base de données </li>
<li> Programmation web
      <ul>
        <li>HTML ET CSS</li>
        <li> PHP</li>
        <li> JavaScript</li>
        <li> WordPress</li>
      </ul>
</li>
<li> Et autres...</li>
</ol>
<p> Nos cours ne demandent aucun prérequis, vous n'avez pas besoin de connaître quelque chose pour lire nos cours<br/>
    Nous écrivons nos cours avec l'hypothèse que vous ne connaissez presque rien sur la matière, c'est pourquoi nous expliquons avec les moindres détails</p>
<h2> Les formateurs</h2>
    Les formateurs de Wissen Corporation sont :
```

```
<ul>
  <li> Aimé DIUMI DIKOLO</li>
  <li> Kleine KAMBALA</li>
  <li> Nestor ELOMBO WASSI</li>
  <li> Andy KALOMBOV MBIOLA</li>
  <li> Grace PHAKA NZUNZI</li>
  <li> Billy PAUL BOLIKO</li>
  <li> Ruben MOSEPINZA</li>
  <li>Armel KINGOLO</li>
</ul>
</body>
</html>
```

Le code ci-dessus produit le résultat suivant :

Formation en Html et Css

file:///C:/Users/Aimé DIUMI/Docume...

# WISSEN CORPORATION

## Présentation

Wissen Corporation est un groupe d'étudiants de la faculté de Sciences et Technologies de l'université de Kinshasa (UNIKIN) en République démocratique du Congo (R.D.C.)  
Le premier objectif poursuivi par ce groupe est le partage de la connaissance, car la connaissance c'est comme un chandelier et sa lumière ne diminue pas s'il allume un autre chandelier

## Formations organisées

Nous organisons des formations suivantes:

1. Programmation
  - o Visual basic
  - o C#
  - o Java
  - o Python
  - o langage C
2. Base de données
3. Programmation web
  - o HTML ET CSS
  - o PHP
  - o JavaScript
  - o WordPress
4. Et autres...

Nos cours ne demandent aucun prérequis, vous n'avez pas besoin de connaître quelque chose pour lire nos cours  
Nous écrivons nos cours avec l'hypothèse que vous ne connaissez presque rien sur la matière, c'est pourquoi nous expliquons avec les moindres détails

## Les formateurs

- Aimé DIUMI DIKOLO
- Kleine KAMBALA
- Nestor ELOMBO WASSI
- Andy KALOMBOV MBIOLA
- Grace PHAKA NZUNZI
- Billy PAUL BOLIKO
- Ruben MOSEPINZA
- Armel KINGOLO

## I.7 LES IMAGES

Il existe différents formats d'images que l'on peut utiliser sur les sites web, si l'image est volumineuse à télécharger, cela ralentit le chargement de la page web : faites un choix judicieux des images à utiliser !!!!

Pour insérer une image en html, on utilise la balise orpheline <img/>, mais on doit renseigner deux attributs obligatoires :

- **Src** : permet d'indiquer le nom de l'image à insérer, il permet aussi d'indiquer où se trouve l'image en question (dans le cas où l'image ne se trouve pas dans le même dossier que le fichier html).
- **Alt** : Il peut arriver que le navigateur soit incapable d'afficher l'image pour une raison quelconque, alt permet d'afficher quelque chose à la place de l'image dans un cas pareil. En d'autres termes alt est le texte alternatif à l'image.

Si tout fonctionne normalement, l'image sera affichée, sinon c'est le texte que vous avez défini avec alt qui va s'afficher à la place de l'image. Sachez aussi que alt aide les robots des moteurs de recherche pour les recherches d'images.

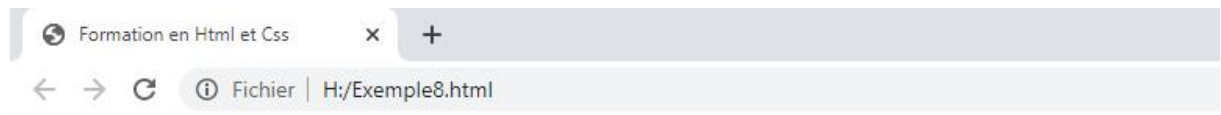
### Exemple

Insérons deux images, l'une se trouvant dans le même dossier que mon fichier html et l'autre se trouvant ailleurs :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Formation en Html et Css</title>
  </head>
  <body>
<p>Les images de la formation passée: <br/>


  </p>
  </body>
</html>
```

Et voici le résultat :



Les images de la formation passée:



Mettons un nom d'image qui n'existe pas à la place de la deuxième image :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Formation en Html et Css</title>
  </head>
  <body>
    <p>Les images de la formation passée: <br/>
    
    
  </p>
  </body>
</html>
```

On peut voir qu'à la place de la deuxième photo, le navigateur a affiché le texte que nous avons défini pour l'attribut alt (1<sup>er</sup> jour de la formation) parce qu'il n'a pu télécharger l'image (qui n'existe pas).



## Une infobulle pour l'image

Pour afficher une infobulle lorsque le curseur touche l'image, on utilise l'attribut title :

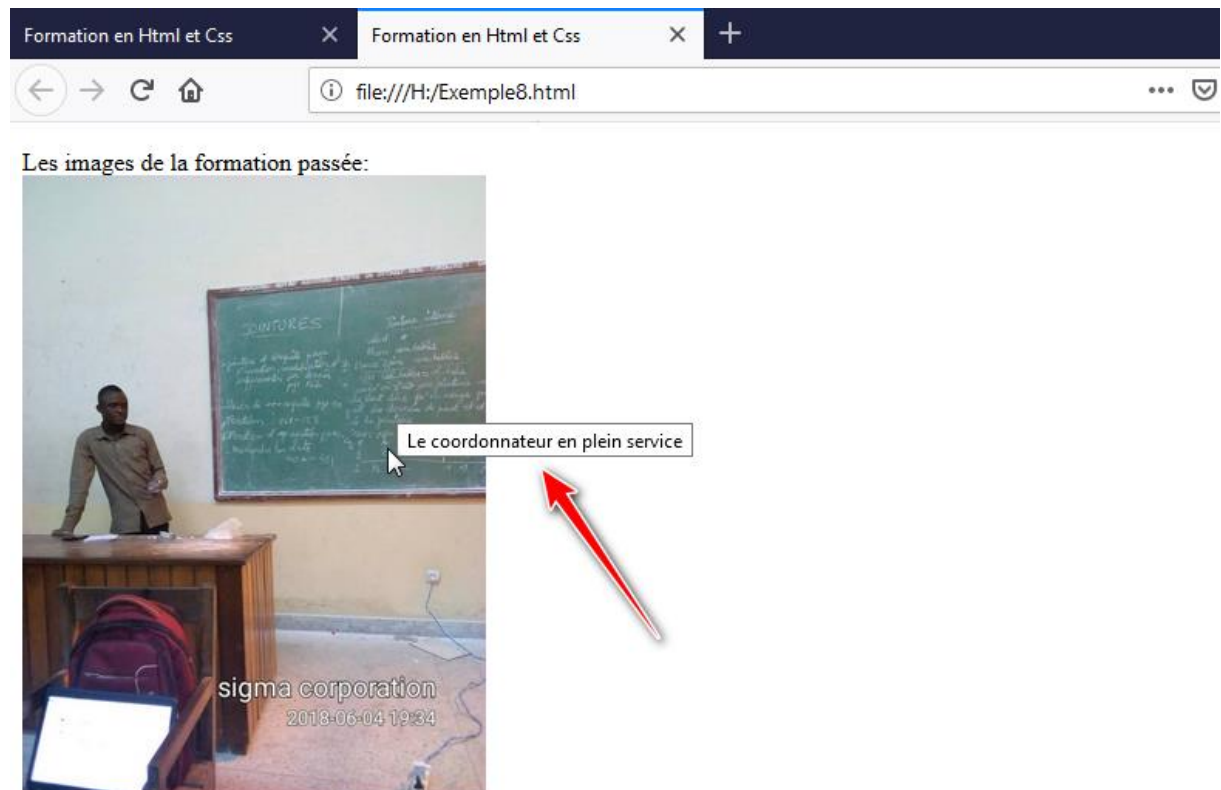
Ajoutons ça dans le code précédent :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Formation en Html et Css</title>
  </head>
  <body>
<p>Les images de la formation passée: <br/>

  </body>
</html>
```

Le résultat :





## I.8.LES LIENS

Nous avons vu qu'un site web est un ensemble de pages web interconnectés, nous allons voir comment créer des liens entre les pages d'un site web.

Un lien, en HTML, va généralement servir à transporter un utilisateur d'un endroit vers un autre après que celui-ci ait cliqué dessus.

### I.8.1. Lien vers une autre page du site

On va examiner deux cas :

- ❖ Les deux pages se trouvent dans le même dossier
- ❖ Les deux pages sont dans des dossiers différents

#### *a) Les deux pages sont situées dans le même dossier*

Nous allons créer deux pages que nous enregistrerons dans le même dossier.

Dans chaque page j'ai juste écrit un peu de texte pour indiquer le numéro de la page.

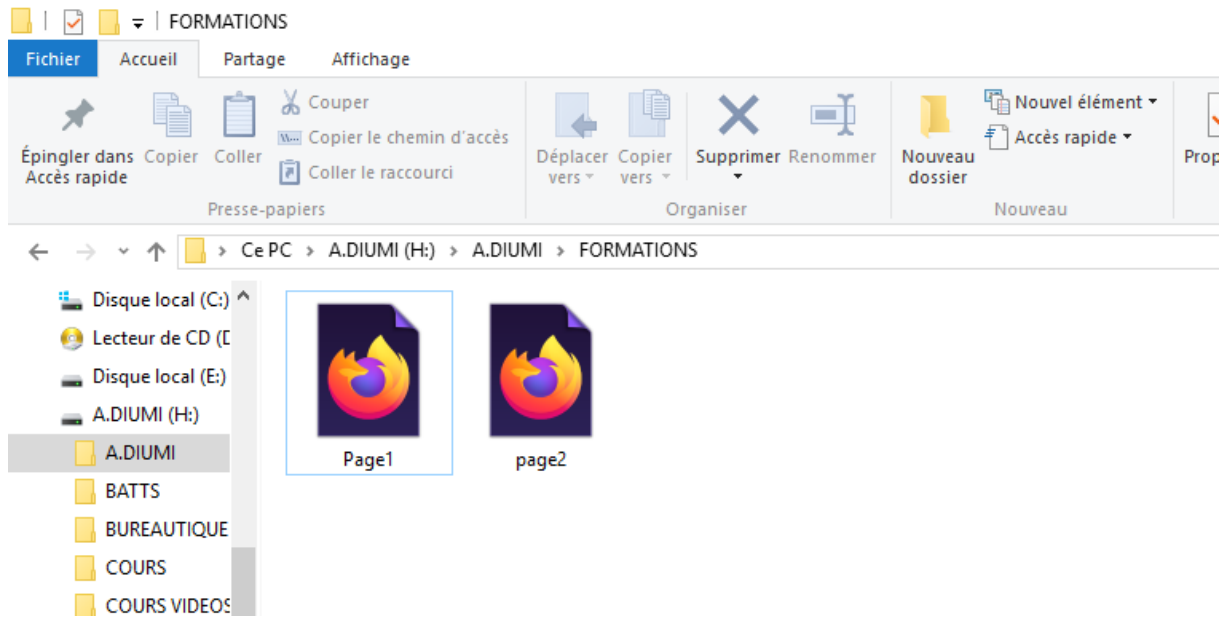
Code page1.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Page 1</title>
  </head>
  <body>
    <p> Bonjour, ici c'est la première page</p>
  </body>
</html>
```

Code page2.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Page 1</title>
  </head>
  <body>
    <p> Bonjour, ici c'est la deuxième page</p>
  </body>
</html>
```

Rassurez-vous avoir enregistré les deux fichiers dans le même dossier (Pour moi, c'est dans le dossier FORMATIONS) :



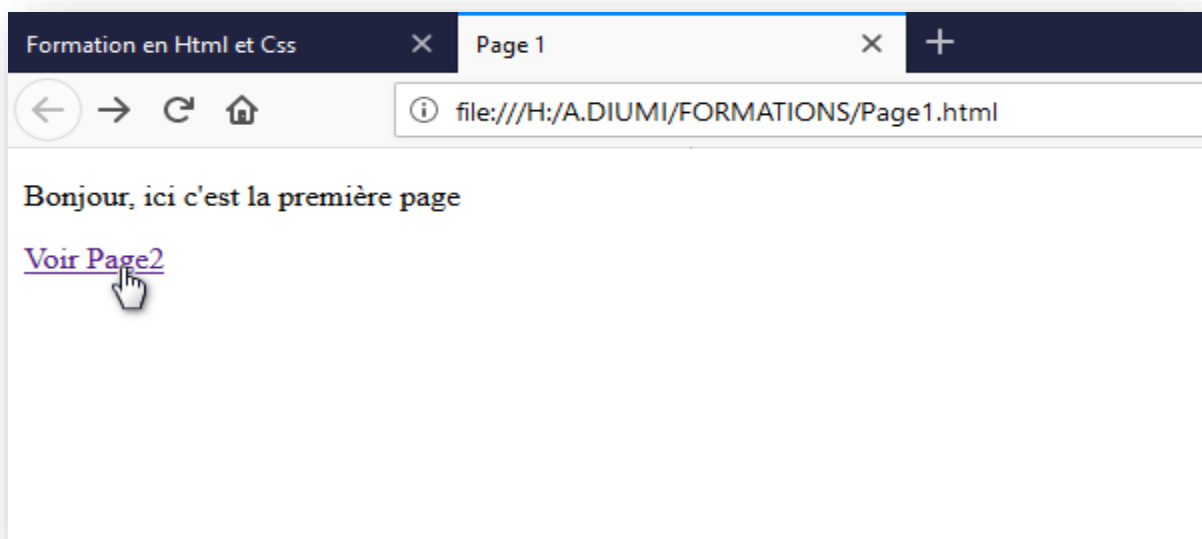
Notre souci est d'atteindre page2 à partir page1.

C'est la balise `<a>` `</a>` qui permet de faire des liens, elle demande l'attribut href (initiales d'Hypertexte Reference) pour indiquer le nom ou l'adresse de la page ciblée.

Il suffit d'ajouter cette ligne dans le code de page1.html

```
<a href="page2.html" > Voir Page2</a>
```

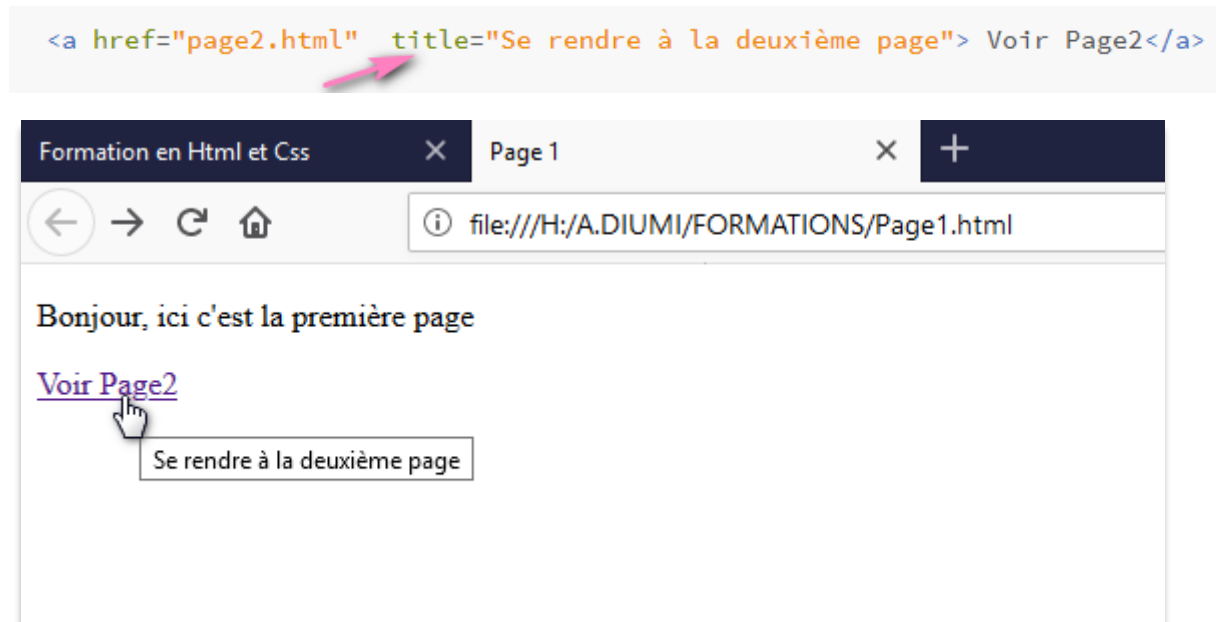
Le texte entre la balise ouvrante et fermante est ce qui va s'afficher (ce que l'utilisateur verra) comme le montre la figure suivante :



Il suffit de cliquer sur ce lien (Voir page2) pour se rendre à la page2.

Par défaut, le lien s'affiche en bleu souligné. Si vous avez déjà ouvert la page, le lien s'affiche en violet. Mais, on peut changer ça avec CSS.

On peut ajouter une infobulle comme on l'a vu avec les images :

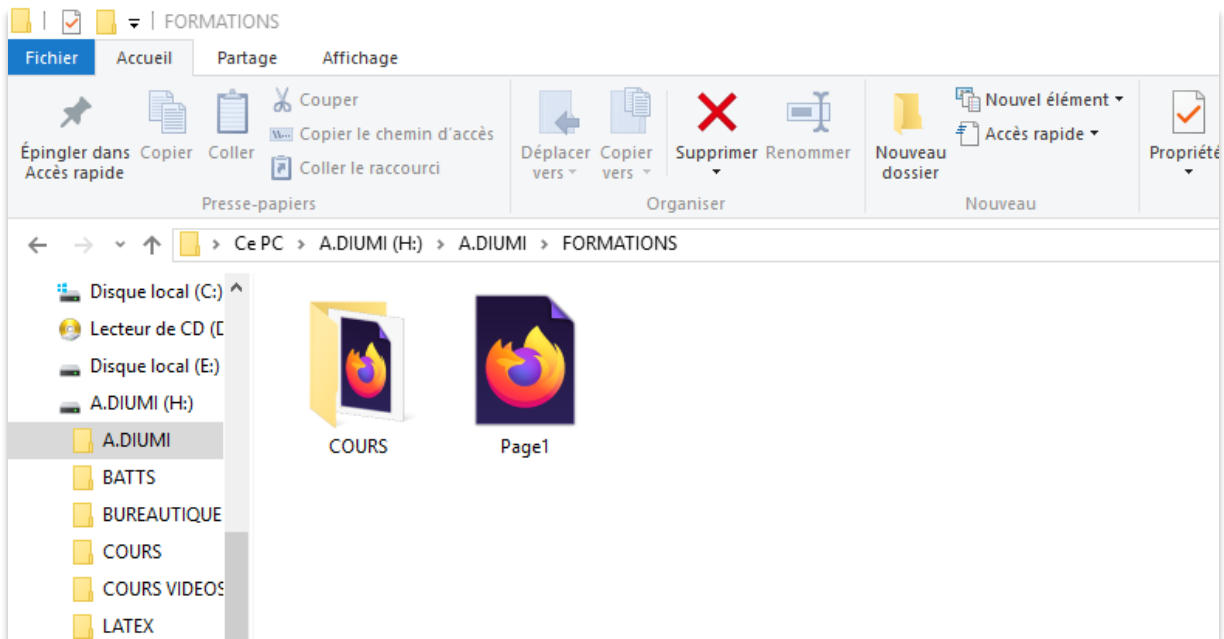


### *b) Les deux pages sont situées dans deux dossiers différents*

Examinons tous les cas possibles :

1<sup>er</sup> cas :

Utilisons toujours nos deux pages, mais cette fois ci enregistrons page2.html dans un sous-dossier du dossier contenant page1.html comme le montre la figure suivante :



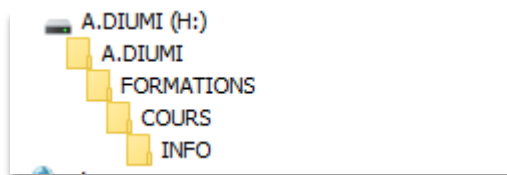
Page2.html est située dans le dossier COURS qui est un sous-dossier du dossier contenant page1.html (dans mon cas, c'est le dossier FORMATIONS).

Pour créer un lien, il suffit d'écrire  
NomDuSousDossier/nom\_de\_la\_page\_ciblée

```
<a href="COURS/page2.html" > Voir Page2</a>
```

2<sup>e</sup> cas :

Supposons que page2.html se situe dans un sous-sous dossier comme le montre la figure suivante :



Page1.html est située dans FORMATIONS et page2.html se situe dans INFO.

Pour créer le lien à partir de page1.html vers page2.html, on écrira :

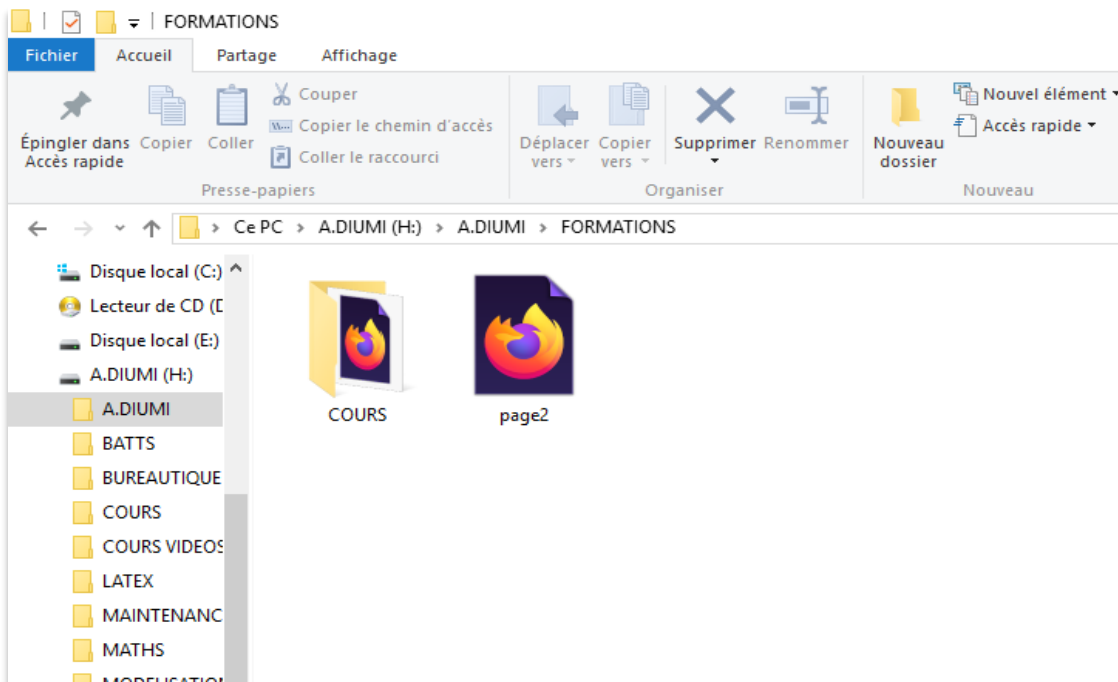
```
<a href="COURS/INFO/page2.html" > Voir Page2</a>
```

Si page1 était située dans le dossier A.DIUMI, le lien serait :

```
<a href="FORMATIONS/COURS/INFO/page2.html" > Voir Page2</a>
```

3<sup>e</sup> cas : si la page cible se trouve dans un dossier parent :

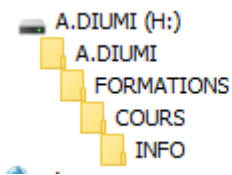
Prenons un exemple simple : nous disposons de deux pages : page1.html et page2.html, la première est située dans le sous-dossier et la deuxième se trouve dans le dossier parent :



Nous voulons créer un lien de page1 vers page2 qui se situe dans le dossier parent par rapport à page1 (origine du lien). Dans ce cas, on écrira :

```
<a href=" ../page2.html" > Voir Page2</a>
```

Pour terminer, reprenons le schéma ci-dessous :



Nous voulons créer un lien de page1.html (qui se situe dans INFO) vers page2.html (située dans FORMATIONS) :

```
<a href=" ../../page2.html" > Voir Page2</a>
```

J'ose croire que certains parmi vous ont deviné...si page2 était située dans A.DIUMI, le lien serait :

```
<a href=" ../../../../page2.html" > Voir Page2</a>
```

En bref, pour ce qui concerne les liens vers une page située dans un autre dossier : on utilise le nom du dossier ou sous dossier pour descendre dans l'arborescence et on utilise ../ pour remonter.

## I.8.2 Lien vers un autre site

Pour créer le lien vers un autre site, il suffit d'indiquer l'adresse du site dans l'attribut href comme le montre l'exemple suivant :

```
<a href="http://wissen-corporation.com" > Visiter le site de Wissen </a>
```

## I.8.3 Lien vers une ancre (point de repère)

Une ancre est un point de repère, il est parfois intéressant d'atteindre un point précis de la page ou du site.

Par exemple, si la page est longue, on peut créer un lien permettant au visiteur d'atteindre directement le haut de la page en un clic.

Prenons l'exemple de cette page :

Copier-coller ce code dans votre éditeur de texte :

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Page 1</title>
  </head>
  <body>
<h1>AVANT PROPOS </h1>
<h2>WISSEN CORPORATION </h2>
<p>C'est un groupe formé essentiellement de meilleurs étudiants de nos groupes
d'étude et d'encadrement qui se sont fixés plusieurs objectifs dont le premier
est celui du partage de la connaissance en Informatique et en mathématique.
</p>
<p>
Albert Einstein a dit : « Il ne faut pas faire de l'intelligence un dieu, car
si celle-
ci est douée de muscles puissants, elle n'a pas de personnalité » ...<br/>
Parfois, il n'existe pas de gens plus intelligents que les autres, mais c'est
le travail qui nous différencie. <br/>
Pour bâtir haut, il faut creuser profond et tu seras fort dans ce que tu te do
nnes. </p>
<p>Il y a une naissance en toute connaissance, Thucydide a dit : « Avoir des c
onnaissances sans les partager, c'est se mettre au niveau de celui qui n'a pas
d'idées ». Wissen Corporation est là pour assurer votre formation, votre enca
drement suivant vos désirs. </p>
<h1> LES BASES DU LANGAGE ET MODE CONSOLE </h1>
<p>Wikipédia présente Visual basic comme un langage de programmation événement
ielle de troisième génération ainsi qu'un environnement de développement intég
```

ré, créé par Microsoft pour son modèle COM. Visual basic est directement dérivé du BASIC (acronyme pour Beginner's All purpose Symbolic Instruction Code) et permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO,

ADO et RDO, ainsi que la création de contrôle ou objets Active X.

Pour coder dans un langage de programmation, il faut des outils adaptés : un éditeur de texte, un compilateur ainsi qu'un débogueur. Il y a des programmes qui combinent les trois outils et ces programmes sont appelés Environnement de développement intégré (EDI) ou IDE en anglais, pour Integrated development Environment.

L'éditeur de texte nous permet d'écrire le code source du programme. Le compilateur (Interpréteur) pour transformer le code source en binaire qui est le seul langage compréhensible par la machine.

Le débogueur permet de traquer les erreurs dans le programme.

Pour Visual basic, nous pouvons utiliser : Visual studio, Visual studio Express, SharpDevelop, MonoDevelop etc.

## I. LES VARIABLES

Visual Basic, comme la plupart des langages de programmation, utilise des variables pour stocker des valeurs.

Elles servent à conserver momentanément des données en mémoire.

Pratiquement, nous pouvons supposer que déclarer des variables : c'est comme réserver des cases mémoires dans l'ordinateur et dans ces cases mémoires seront stockés momentanément les valeurs saisies par les utilisateurs ou les résultats de calculs de nos programmes.

Une variable a un nom (le mot que vous utilisez pour faire référence à la valeur que la variable contient). Elle a également un type de données (lequel détermine le genre des données que la variable peut stocker) et peut contenir à un instant donné une valeur.

Vous pouvez nommer vos variables comme vous le voulez, mais il y a des règles et conseils à suivre, notamment :

- 

- Vous pouvez mettre des chiffres et des lettres dans les noms de variables mais le premier caractère doit être un caractère ou le trait de soulignement ( \_ )

- N'utilisez pas d'espaces ou des symboles

- N'utilisez pas les mots clés de Visual basic comme noms de variables, par exemple Dim, integer, date, select etc.

- Choisissez des noms explicites, significatifs et évocateurs qui rappellent ce que contient la variable. Par exemple nombre\_de\_vie est explicite et ce n'est pas le cas pour ndv

- Evitez les abréviations

- 

Lorsqu'Option Explicit est On c'est-à-dire activé (réglage par défaut), vous devez déclarer explicitement les variables avant leurs utilisations, mais lorsqu'il est Off vous pouvez déclarer implicitement une variable en l'utilisant dans votre programme. Mais il est viveme



```
nt conseillé de déclarer explicitement vos variables avant de les utiliser. </  
p>  
    </body>  
</html>
```

La page est assez longue, on peut donner au visiteur la possibilité de rentrer directement en haut de la page s'il est en bas.

Le lien vers une ancre se fait en deux étapes :

- D'abord utiliser l'attribut id pour donner un nom à la balise (ou le point) qui va servir de repère.

Pour notre exemple, si on veut que lien puisse l'amener sur AVANT PROPOS, on doit ajouter l'attribut id pour identifier ce point de repère :

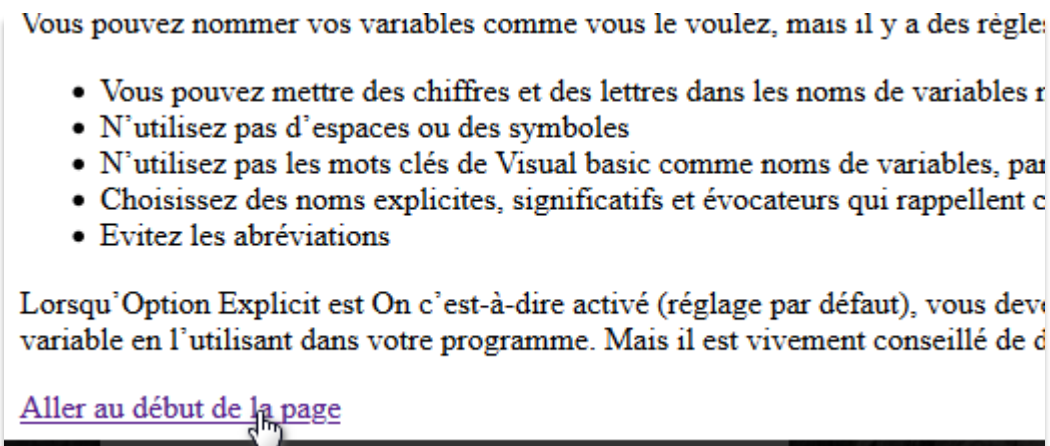
```
<h1 id="Repere">AVANT PROPOS </h1>
```

- Ensuite, il suffit de créer un lien comme d'habitude, mais cette fois l'attribut href contiendra un dièse (#) suivi du nom de l'ancre.

Pour notre exemple, ajoute cette ligne avant la balise fermante de body

```
<a href="#Repere" > Aller au début de la page</a>
```

Et voici le résultat :



Vous pouvez nommer vos variables comme vous le voulez, mais il y a des règles :

- Vous pouvez mettre des chiffres et des lettres dans les noms de variables
- N'utilisez pas d'espaces ou des symboles
- N'utilisez pas les mots clés de Visual basic comme noms de variables, par exemple
- Choisissez des noms explicites, significatifs et évocateurs qui rappellent ce que la variable va contenir
- Evitez les abréviations

Lorsqu'Option Explicit est On c'est-à-dire activé (réglage par défaut), vous devez déclarer chaque variable en l'utilisant dans votre programme. Mais il est vivement conseillé de déclarer les variables même si l'option est désactivée.

[Aller au début de la page](#)

Il suffit de cliquer sur le lien et on remonte en haut de la page.

Pour créer un lien vers une ancre située dans une autre page, on écrit :

```
<a href="NomPage.html#Repere" > Message</a>
```

NomPage étant le nom de la page contenant l'ancre ciblée et Repère le nom de l'ancre.

## 1.8.4 Lien pour télécharger un fichier

Le lien se crée de la même façon que les autres liens, sauf qu'il faut juste mettre le nom du fichier à télécharger comme valeur à l'attribut href.

Supposons que le fichier à télécharger est : 1001Equations.pdf, il suffit d'écrire :

```
<a href="1001Equations.pdf"> Télécharger</a>
```

Le navigateur lancera le processus de téléchargement dès que le visiteur cliquera sur le lien.

## 1.8.5 Lien pour envoyer un email

On utilise le lien de type mailto comme le montre l'exemple ci-après :

```
<a href="mailto:VotreAdresseMail">Envoyez-moi un e-mail </a>
```

Il suffit donc de faire commencer le lien par mailto : et d'écrire l'adresse e-mail où on peut vous contacter. Si vous cliquez sur le lien, un nouveau message vide s'ouvre, prêt à être envoyé à votre adresse e-mail.

# 1.9 L'AUDIO ET LA VIDEO

## 1.9.1 L'audio

Pour insérer une audio, on utilise la balise `<audio>` `</audio>` à laquelle on ajoute les attributs suivants :

- **src** : pour indiquer l'URL (l'adresse) de l'audio.
- **controls** : pour ajouter les boutons « Lecture », « Pause » et la barre de défilement. Car ils ne figurent pas par défaut.
- **width** : permet de modifier la largeur de l'outil de lecture audio.
- **Loop** : indique si le fichier sera joué en boucle, c'est-à-dire le morceau sonore est joué à nouveau s'il se termine.
- **autoplay** : la musique sera jouée dès le chargement de la page sans attendre le lancement par le visiteur.
- **preload** : indique si la musique peut être préchargée dès le chargement de la page ou non. Cet attribut peut prendre les valeurs suivantes :
  - auto : le navigateur décide s'il doit précharger toute la musique, uniquement les métadonnées ou rien du tout. C'est la valeur par défaut.
  - metadata : charge uniquement les métadonnées (durée, etc.).
  - none : pas de pré chargement. Cette valeur est utile si vous ne voulez pas gaspiller de bande passante sur votre site.

## Exemple

```
<audio src="chanson.mp3" controls> </audio>
```

Résultat :



Sachez que l'apparence du lecteur audio change en fonction du navigateur.

Etant donné que tous les navigateurs ne reconnaissent pas tous les formats des audio, il est conseillé d'insérer plusieurs formats de la même audio pour résoudre ce problème.

Pour cela, on utilise la balise `<source>` qui permet de spécifier plusieurs sources audio comme le montre l'exemple suivant :

```
<audio controls>
  <source src="chanson_ogg.ogg">
  <source src="chanson_mp3.mp3">
  <source src="chanson_acc.acc">
</audio>
```

La charge est maintenant au navigateur de choisir le format qu'il prend en compte.

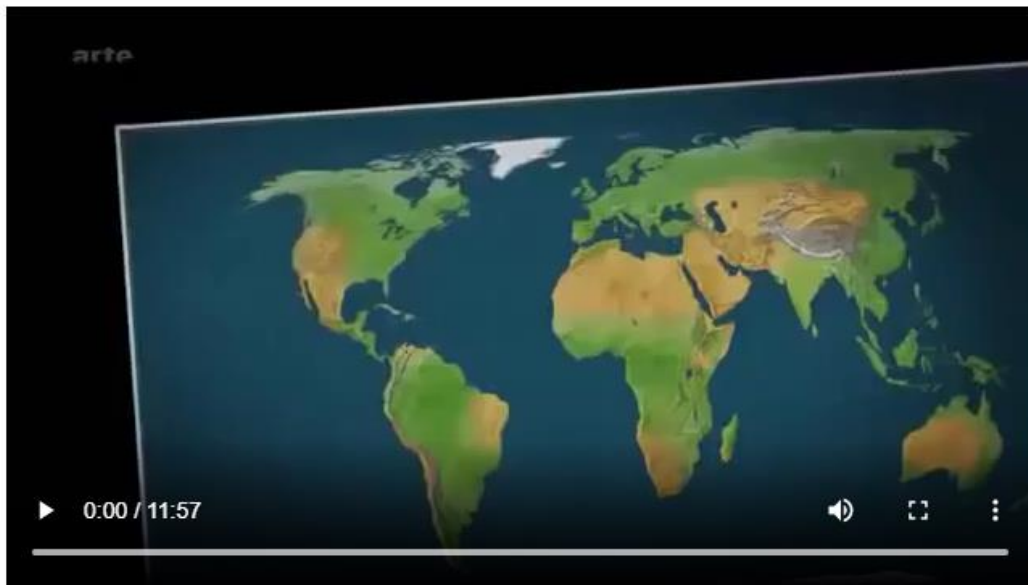
## 1.9.2 La vidéo

Pour insérer une vidéo, on utilise la balise `<video>` `</video>`, à laquelle on peut ajouter les attributs suivants :

- **src** : pour indiquer l'URL (l'adresse) de la vidéo.
- **controls** : pour ajouter les boutons « Lecture », « Pause » et la barre de défilement.

```
<video src="documentaire.mp4" controls > </video>
```

Résultat :



- **poster** : permet d'indiquer l'image à afficher à la place de la vidéo tant que celle-ci n'est pas lancée. Par défaut, le navigateur prend la première image de la vidéo. Dans plusieurs cas, il s'agit souvent d'une image noire ou d'une image peu représentative de la vidéo, c'est pourquoi il est conseillé de créer soi-même une image.

Par exemple, j'aimerais afficher l'image suivante (une capture de la vidéo en un moment) avant le lancement de la vidéo :



```
<video src="documentaire.mp4" controls poster="capture.png" > </video>
```

Résultat :



Et voilà l'image avant le lancement de la vidéo a changé.

- **width** : permet de modifier la largeur de la vidéo.
- **height** : permet de modifier la hauteur de la vidéo.
- **autoplay** : la vidéo sera jouée dès le chargement de la page sans attendre le lancement par le visiteur.
- **preload** : indique si la vidéo peut être préchargée dès le chargement de la page ou non. Voici les valeurs possibles pour cet attribut :
  - **auto** : le navigateur décide s'il doit précharger toute la vidéo, uniquement les métadonnées ou rien du tout. C'est la valeur par défaut.
  - **metadata** : le navigateur charge uniquement les métadonnées (durée, dimensions, etc.).
  - **none** : pas de préchargement. Utile si vous souhaitez éviter le gaspillage de bande passante sur votre site.

## II. MISE EN FORME DU SITE AVEC CSS

CSS signifie Cascading Stylesheets, soit « feuilles de style en cascade ». Il a été créé en 1996 et a pour rôle principal de mettre en forme du contenu en lui appliquant ce qu'on appelle des styles. Css s'occupe de la mise en forme, l'embellissement du site (couleur, alignement, bordures etc.).

Cependant, beaucoup confondent les fonctions respectives du HTML et du CSS et utilisent un langage pour faire le travail de l'autre. En l'occurrence, c'est le HTML qui est souvent utilisé à mauvais escient, pour mettre en forme du contenu or que c'est le travail du Css. N'oubliez jamais ceci : Il ne faut JAMAIS utiliser le HTML pour faire le travail du CSS : c'est une mauvaise pratique.

### II.1 GENERALITES

Dans html, on utilise des balises et dans Css, ce sont des propriétés qui sont utilisées.

#### II.1.1 Structure du code CSS

Les feuilles de style en CSS sont constituées de règles. Chacune des règles est en trois parties :

1. Le sélecteur, qui indique au navigateur quelle partie du document est affectée par la règle ;
2. La propriété, qui spécifie quel aspect de l'affichage est paramétré et
3. la valeur, qui indique la valeur de la propriété de style.

Schématiquement, le code CSS a la structure suivante :

```

Selecteur1
{
  propriété1:valeur;
  propriété2: valeur;
  ....
}

```

```

Selecteur2
{
  propriété: valeur;
  propriété: valeur;
  propriété : valeur;
  .....
}
.....

```

Html a plusieurs éléments, le sélecteur permet de préciser à quel élément html on veut appliquer les propriétés contenues entre accolades. Donc un sélecteur permet de cibler un élément HTML.

Si par exemple, on veut les appliquer aux paragraphes, le sélecteur sera p :

```

p
{
  propriété: valeur;
  propriété: valeur;
  propriété: valeur;
  .....
}

```

Cela veut simplement dire que les propriétés ci-dessous vont

s'appliquer uniquement aux paragraphes.

On peut mettre autant de propriétés que l'on veut à l'intérieur des accolades. Chaque propriété est suivie du symbole « deux-points » ( : ) puis de la valeur correspondante. Enfin, chaque ligne se termine par un point-virgule ( ; ). A part les noms de balises html, il existe d'autres sélecteurs qu'on verra plus loin. Maintenant que nous savons comment écrire les codes CSS, voyons où les écrire...

## II.1.2 Où écrit-on le code CSS

Avant d'aborder le sujet, copier-coller ce code html dans votre éditeur de texte :

```

<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
  </head>
  <body>
    <!doctype html>
    <html>
      <head>
        <meta charset="utf-8">
        <title> Page 1</title>
      </head>

```

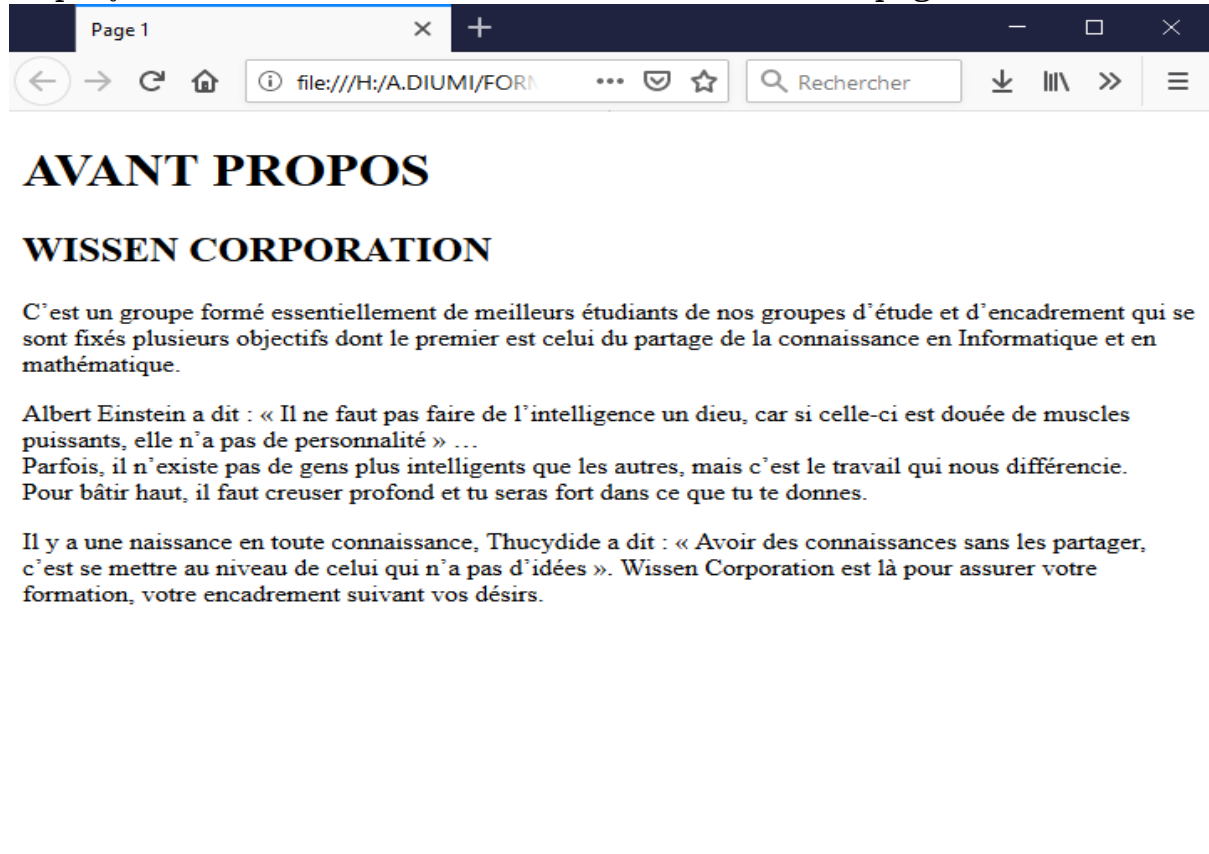
```

<body>
<h1 >AVANT PROPOS </h1>

<h2>WISSEN CORPORATION </h2>
<p>C'est un groupe formé essentiellement de meilleurs étudiants de nos
groupes d'étude et d'encadrement qui se sont fixés plusieurs objectifs dont l
e premier est celui du partage de la connaissance en Informatique et en mathém
atique. </p>
<p> Albert Einstein a dit : « Il ne faut pas faire de l'intelligence
un dieu, car si celle-
ci est douée de muscles puissants, elle n'a pas de personnalité » ...<br/>
Parfois, il n'existe pas de gens plus intelligents que les autres, mai
s c'est le travail qui nous différencie. <br/>
Pour bâtir haut, il faut creuser profond et tu seras fort dans ce que
tu te donnes. </p>
<p>Il y a une naissance en toute connaissance, Thucydide a dit : « Avo
ir des connaissances sans les partager, c'est se mettre au niveau de celui qui
n'a pas d'idées ». Wissen Corporation est là pour assurer votre formation, vo
tre encadrement suivant vos désirs. </p>
</body>
</html>

```

L'aperçu de cette page est :





On peut écrire le code CSS dans trois endroits différents :

### *a) Directement dans la balise concernée*

Si je souhaite que le dernier paragraphe soit en bleu, je vais me rendre directement à la balise concernée (ici p), dans la balise ouvrante, j'ajoute l'attribut style et c'est là que j'écrirai les propriétés à appliquer à cette balise.

Sachez que la propriété qui permet de changer la couleur est Color, la couleur bleue c'est blue. Le code devient :

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Page 1</title>
  </head>
  <body>

<h1 >AVANT PROPOS </h1>

<h2>WISSEN CORPORATION </h2>

<p>C'est un groupe formé essentiellement de meilleurs étudiants de nos
groupes d'étude et d'encadrement qui se sont fixés plusieurs objectifs dont
le premier est celui du partage de la connaissance en Informatique et en
mathématique. </p>
<p>
Albert Einstein a dit : « Il ne faut pas faire de l'intelligence un dieu,
car si celle-ci est douée de muscles puissants, elle n'a pas de personnalité » ...<br/>
Parfois, il n'existe pas de gens plus intelligents que les autres, mais c'est
le travail qui nous différencie.<br/>
Pour bâtir haut, il faut creuser profond et tu seras fort dans ce que tu te donnes. </p>
<p style="color: blue">Il y a une naissance en toute connaissance, Thucydide a dit : « Avoir
des connaissances sans les partager, c'est se mettre au niveau de celui qui
n'a pas d'idées ». Wissen Corporation est là pour assurer votre formation,
votre encadrement suivant vos désirs. </p>

</body>
</html>
```

Et voilà, le dernier paragraphe est en bleu :



Et si on mettait le titre en rouge ?

La procédure est la même, ajouter l'attribut style à la balise et on met Color : red (red c'est rouge en anglais)

```

<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Page 1</title>
  </head>
  <body>
    <h1 style="color: red" >AVANT PROPOS </h1> |
    <h2>WISSEN CORPORATION </h2>

    <p>C'est un groupe formé essentiellement de meilleurs étudiants de nos
    groupes d'étude et d'encadrement qui se sont fixés plusieurs objectifs dont
    le premier est celui du partage de la connaissance en Informatique et en
    mathématique. </p>
    <p>
    Albert Einstein a dit : « Il ne faut pas faire de l'intelligence un dieu,
    car si celle-ci est douée de muscles puissants, elle n'a pas de personnalité » ...<br/>
    Parfois, il n'existe pas de gens plus intelligents que les autres, mais c'est
    le travail qui nous différencie.<br/>
    Pour bâtir haut, il faut creuser profond et tu seras fort dans ce que tu te donnes. </p>

    <p style="color: blue">Il y a une naissance en toute connaissance, Thucydide a dit : « Avoir
    des connaissances sans les partager, c'est se mettre au niveau de celui qui
    n'a pas d'idées ». Wissen Corporation est là pour assurer votre formation,
    votre encadrement suivant vos désirs. </p>

  </body>
</html>

```

Et voilà :



Imaginez-vous que votre page a cinquante paragraphes et vous voulez qu'ils soient tous en bleu, avec cette méthode, vous serez obligé d'écrire le même code dans les cinquante paragraphes : Avouez que c'est fatigant !!! c'est pour cette raison que cette méthode n'est pas recommandée.

### b) Dans l'en-tête <head> </head> du fichier html

On ajoute la balise <style> </style> dans <head> </head>

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Page 1</title>
    <style>
      ↑
      ↓
    </style>
    </head>
    <body>
      <h1 >AVANT PROPOS </h1>
```

*on écrit le code CSS ici.*

Reprenons l'exemple précédent, enlever le code CSS que nous avons ajouté au dernier paragraphe et au titre h1. Nous voulons maintenant que le titre soit en jaune et le paragraphe en bleu.

```

<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Page 1</title>
    <style>
      p
      {
        color: blue;
      }

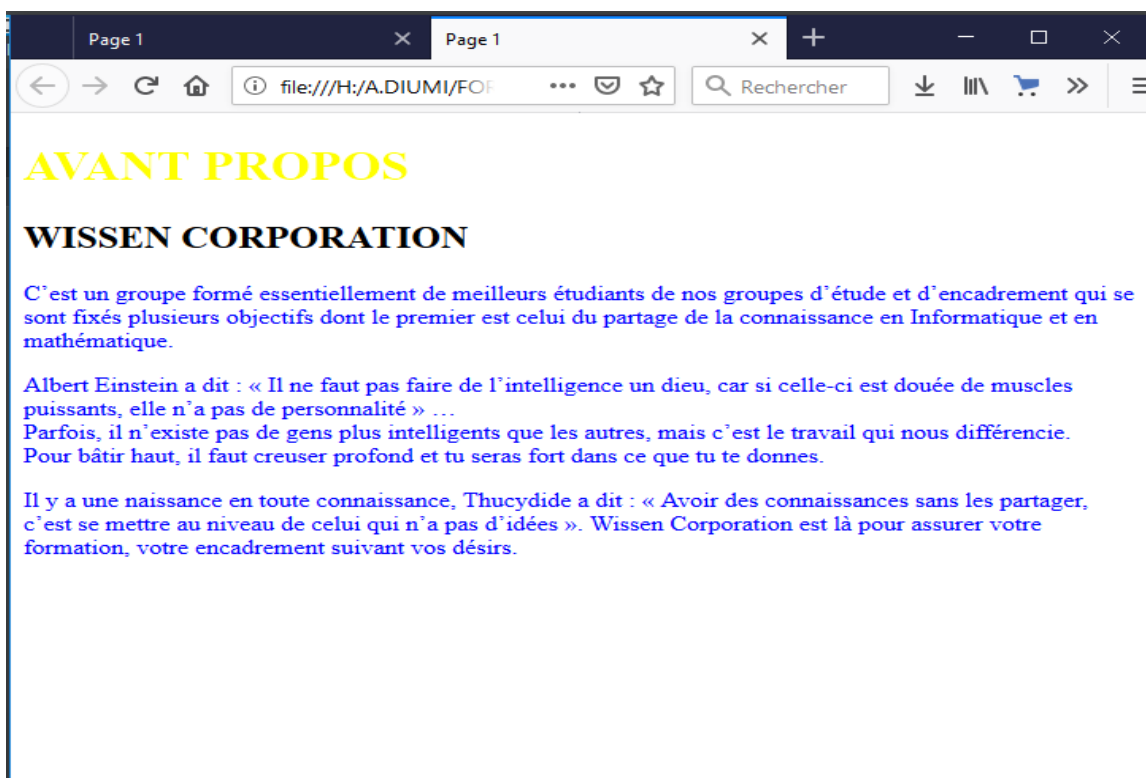
      h1|
      {
        color: yellow;
      }
    </style>
  </head>
  <body>

<h1 >AVANT PROPOS </h1>

```

*Le reste reste inchangé*

On peut voir le résultat :



On voit que tous les paragraphes sont en bleu. L'explication à cela est très simple : Souvenez-vous que la syntaxe du code CSS est :

```
Sélecteur
{
  propriété: valeur;
  propriété: valeur;
  .....
}
```

Nous avons vu que le sélecteur permet d'indiquer les éléments html auxquels appliquer les propriétés entre accolades. Dans le cas des paragraphes, on a écrit :

```
p
{
  color: blue;
}
```

Le sélecteur ici est <p>, c'est-à-dire la propriété s'applique à tous les paragraphes : c'est pourquoi ils sont tous en bleu.

Cette méthode est moins recommandée.

### c) Dans un fichier CSS

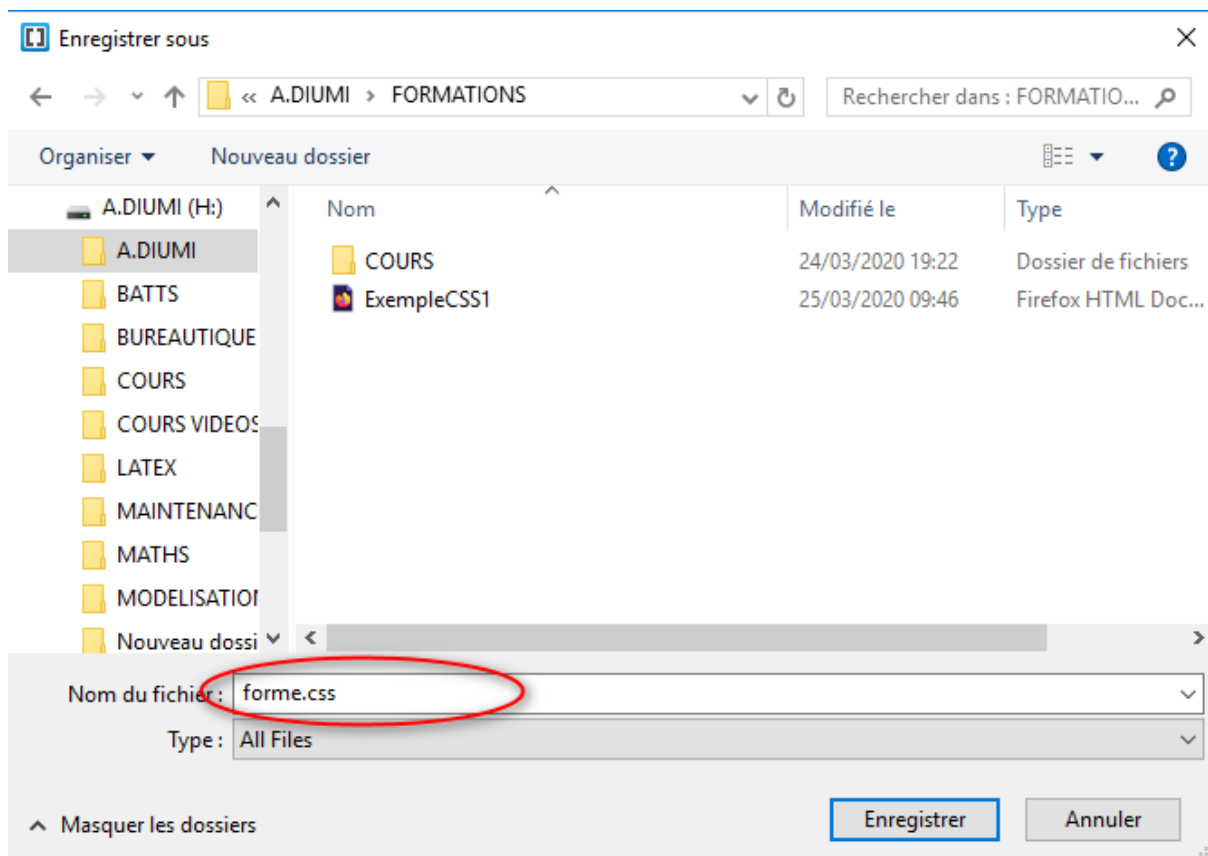
On va écrire le code CSS dans un fichier css (ayant l'extension .css)

Commençons par créer un nouveau projet dans notre éditeur de texte et mettez-y le code suivant :

```
p
{
  color: green;
}

h1
{
  color: yellow;
}
```

Enregistrez-le sans oublier avec l'extension .css



Nommez-le comme bon vous semble, pour ma part, je l'ai appelé forme.css

Maintenant, pour que ce qu'on écrira dans forme.css aie effet dans le fichier html, on doit créer le lien entre les deux fichiers.

Pour associer les deux fichiers, on écrit ceci dans l'en-tête du fichier html (dans <head> </head>) :

```
<link rel=stylesheet href="NomFichier.css" />
```

NomFichier représente le nom de votre fichier css.

Pour notre exemple

Fichier html

```

<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Page 1</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>

<h1 >AVANT PROPOS </h1>

<h2>WISSEN CORPORATION </h2>

<p>C'est un groupe formé essentiellement de meilleurs étudiants de nos
groupes d'étude et d'encadrement qui se sont fixés plusieurs objectifs dont
le premier est celui du partage de la connaissance en Informatique et en
mathématique. </p>
<p>
Albert Einstein a dit : « Il ne faut pas faire de l'intelligence un dieu,
car si celle-ci est douée de muscles puissants, elle n'a pas de personnalité » ...<br/>
Parfois, il n'existe pas de gens plus intelligents que les autres, mais c'est
le travail qui nous différencie.<br/>
Pour bâtir haut, il faut creuser profond et tu seras fort dans ce que tu te donnes. </p>

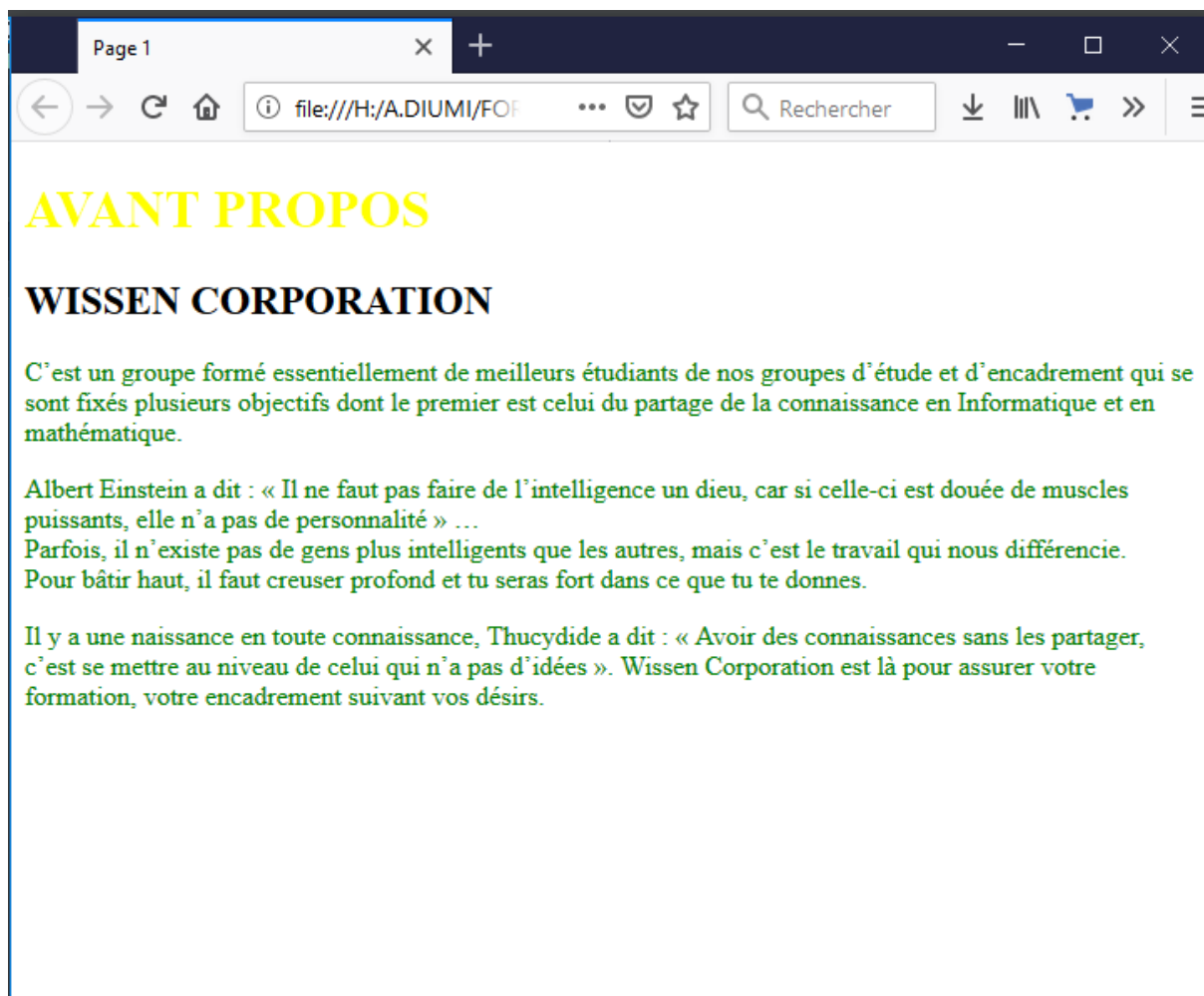
<p>Il y a une naissance en toute connaissance, Thucydide a dit : « Avoir
des connaissances sans les partager, c'est se mettre au niveau de celui qui
n'a pas d'idées ». Wissen Corporation est là pour assurer votre formation,
votre encadrement suivant vos désirs. </p>

</body>
</html>

```

Dès que le lien est créé, tout ce qu'on fera dans forme.css prendra effet dans formation.html comme le montre la figure suivante :





Il est inutile d'ouvrir directement le fichier forme.css dans le navigateur. Il faut ouvrir le fichier Formation.html (il fera automatiquement appel au fichier forme.css).

C'est la méthode la plus pratique et la plus souple. Cela nous évite de tout mélanger dans un même fichier : elle est la méthode recommandée, donc nous allons l'utiliser pour la suite du cours.

Sachez qu'une feuille de style (fichier css) est applicable à une infinité de documents HTML, ce qui en facilite la maintenance et réduit les temps de chargement.

## II.1.3 Les commentaires en CSS

Nous avons parlé des commentaires en HTML, il est également possible d'insérer des commentaires en CSS. La syntaxe est la suivante :

```
/*ceci est un commentaire */
```

Exemple

Forme.css

```
/*Ce style s'appliquera à tous les paragraphes */  
p  
{  
    color: red;  
}
```

On peut les placer partout entre les règles, voire au sein des règles (mais pas entre une propriété et sa valeur) comme le montre suivant :

```
1  
2     /* Ce style s'applique aux paragraphes */  
3     p  
4     {  
5         /*pour définir la couleur du texte */  
6         color: red;  
7     }  
8  
9     /*celui ci s'applique aux titres */  
10    h1  
11    {  
12    color: blue;  
13    }  
14  
15
```

## II.1.4 Appliquer un style particulier avec id et class

Reprenons notre exemple :

Formation.html

```
<!doctype html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <title> Page 1</title>  
    <link rel="stylesheet" href="forme.css" />  
  </head>  
  <body>  
<h1 >AVANT PROPOS </h1>  
  
<h2>WISSEN CORPORATION </h2>
```

```

<p>C'est un groupe formé essentiellement de meilleurs étudiants de nos groupes
d'étude et d'encadrement qui se sont fixés plusieurs objectifs dont le premier
est celui du partage de la connaissance en Informatique et en mathématique.
</p>
<p> Albert Einstein a dit : « Il ne faut pas faire de l'intelligence un dieu,
car si celle-
ci est douée de muscles puissants, elle n'a pas de personnalité » ...<br/>
Parfois, il n'existe pas de gens plus intelligents que les autres, mais c'est
le travail qui nous différencie. <br/>
Pour bâtir haut, il faut creuser profond et tu seras fort dans ce que tu te do
nnes. </p>
<p>Il y a une naissance en toute connaissance, Thucydide a dit : « Avoir des c
onnaissances sans les partager, c'est se mettre au niveau de celui qui n'a pas
d'idées ». Wissen Corporation est là pour assurer votre formation, votre enca
drement suivant vos désirs. </p>
</body>
</html>

```

Forme.css

```

p
{
    color: blue;
}

```

En ouvrant Formation.html, on obtient :



On voit que tous les paragraphes sont en bleu, que faire si l'on souhaite que le dernier paragraphe soit en rouge et les autres en bleu ?

C'est là qu'interviennent les attributs class et id. Ils permettent de donner un nom particulier à une balise pour pouvoir lui appliquer un style particulier dans CSS.

Dans notre cas, on ajoute l'attribut class à la balise <p> du dernier paragraphe :

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Page 1</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>

  <h1 >AVANT PROPOS </h1>

  <h2>WISSEN CORPORATION </h2>

  <p>C'est un groupe formé essentiellement de meilleurs étudiants de nos
groupes d'étude et d'encadrement qui se sont fixés plusieurs objectifs dont
le premier est celui du partage de la connaissance en Informatique et en
mathématique. </p>
  <p>
  Albert Einstein a dit : « Il ne faut pas faire de l'intelligence un dieu,
  car si celle-ci est douée de muscles puissants, elle n'a pas de personnalité » ...<br/>
  Parfois, il n'existe pas de gens plus intelligents que les autres, mais c'est
  le travail qui nous différencie.<br/>
  Pour bâtir haut, il faut creuser profond et tu seras fort dans ce que tu te donnes. </p>
  <p class="Troisieme" >Il y a une naissance en toute connaissance, Thucydide a dit : « Avoir
  des connaissances sans les partager, c'est se mettre au niveau de celui qui
  n'a pas d'idées ». Wissen Corporation est là pour assurer votre formation,
  votre encadrement suivant vos désirs. </p>

  </body>
</html>
```

Je l'ai appelé Troisieme, libre à vous de le nommer comme bon vous semble. On a donné un nom à la balise <p> du troisième paragraphe, maintenant on a la possibilité de lui appliquer un style différent des autres balises <p>

Sur ce, on va dans css, on écrit :

```
p
{
  color: blue;
}

.Troisieme
{
  color: red;
}
```

J'espère que vous avez remarqué le point (.) devant le nom du sélecteur Troisieme !!!!

Ce code peut être expliqué comme ceci : je veux que tous les paragraphes soient en bleu, sauf celui qui s'appelle Troisieme qui doit être en rouge, la figure suivante montre le résultat :



L'attribut id fonctionne de la même façon que class, la seule différence est que deux attributs class dans deux endroits différents peuvent avoir la même valeur, ce qui n'est pas le cas pour l'attribut id, on ne peut pas trouver deux attributs id avec la même valeur dans une page.

Si vous avez utilisé l'attribut class, pour l'identifier dans Css, il faut faire précéder le nom de class de point (.), dans le cas de l'attribut id, on le fait précéder de dièse (#) :

```
.NomAttributClass
{
    .....
}

#NomAttributId
{
    .....
}
```

Utilisez l'attribut id pour les éléments uniques de votre page.

Sachez que l'attribut id a plus de poids que l'attribut class en cas de conflit pour déterminer un style particulier.

## II.1. 5 Les balises universelles

Créer un nouveau projet et écrivez le code suivant :

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Page 1</title>
    <link rel="stylesheet" href="forme.css" />
  </head>
  <body>

<h1> A PROPOS </h1>
<p> Je m'appelle Aimé DIUMI DIKOLO, <br/>
je suis co-fondateur de Wissen Corporation</p>

</body>
</html>
```

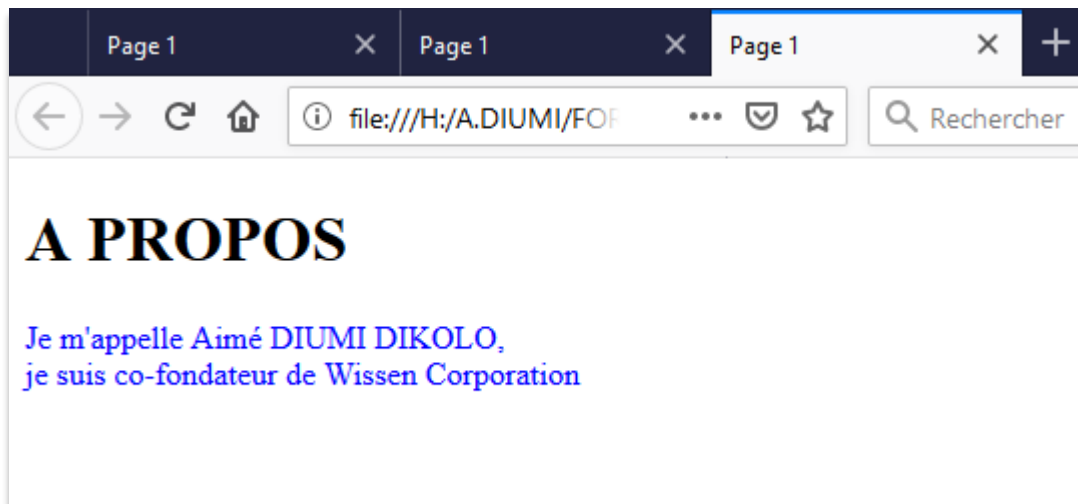
Enregistrez-le avec l'extension html

Créer un autre et mettez-y ceci :

```
p
{
  color: blue;
}
```

Enregistrez-le en le nommant forme.css

Le résultat :



Et si on veut que Aimé DIUMI DIKOLO soit en rouge ? Pas moyen de le faire car Aimé DIUMI DIKOLO n'est pas entouré des balises.

Heureusement, qu'on a inventé deux balises qui n'ont pas une signification particulière : Span et div, elles servent à entourer un mot ou un bloc qui au départ n'était pas entouré par des balises pour lui donner un style voulu.

- Span : est de type inline, c'est-à-dire qu'elle se place à l'intérieur des autres balises pour sélectionner certains mots uniquement (dans un paragraphe par exemple)
- Div : est de type block, elle permet de sélectionner un bloc de texte et elle provoque un retour à la ligne automatique.

Dans notre cas, nous allons utiliser span car on doit sélectionner quelques mots dans un paragraphe : On entoure Aimé DIUMI DIKOLO des balises span sans oublier l'attribut class pour lui donner un nom particulier.

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Page 1</title>
    <link rel="stylesheet" href="forme.css" />
  </head>
  <body>

  <h1> A PROPOS </h1>
  <p> Je m'appelle <span class="nom">Aimé DIUMI DIKOLO</span>, <br/>
  je suis co-fondateur de Wissen Corporation</p>

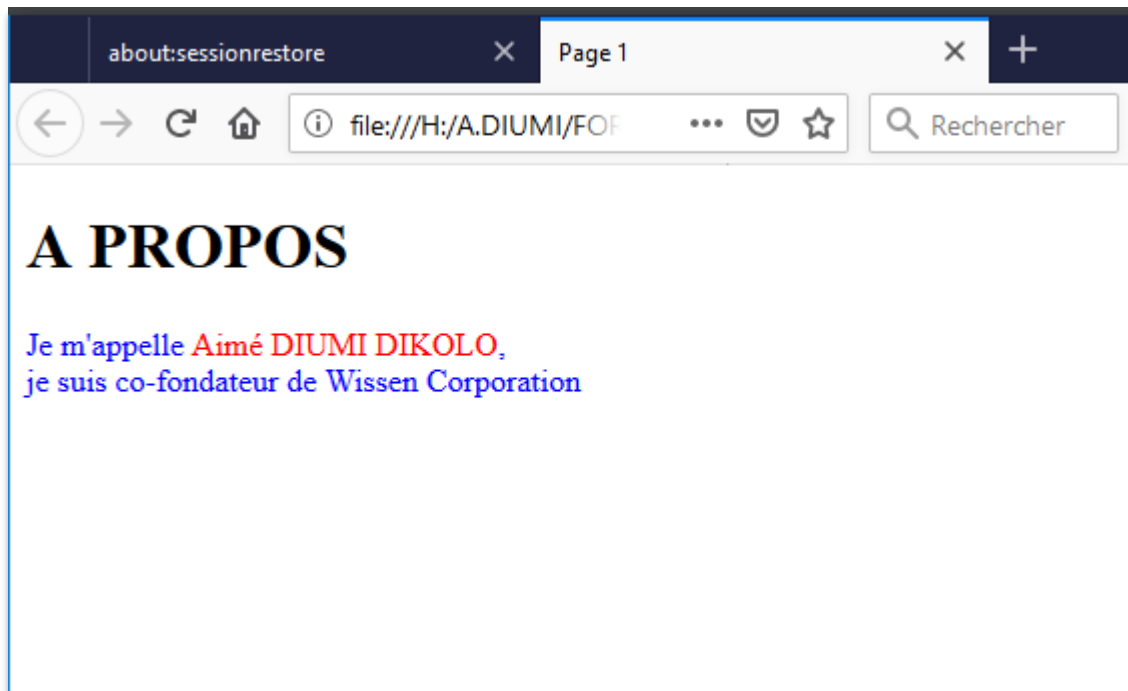
</body>
</html>
```

Si vous ouvrez le fichier, vous verrez qu'il n'y aura aucun changement car la balise span n'a aucun effet particulier. Maintenant, que le nom est entouré d'une balise, on peut lui appliquer un style grâce au CSS :

```
p
{
    color: blue;
}

.nom
{
    color: red;
}
```

On peut voir le résultat :



Le reste du texte est en bleu, sauf Aimé DIUMI DIKOLO est en rouge.



## II.2 LES SELECTEURS AVANCES

Nous avons vu plus haut que le sélecteur permettait d'indiquer les éléments auxquels appliquer le style CSS. Jusque-là nous avons vu comment sélectionner un élément grâce au nom de la balise qui l'entoure ainsi qu'aux attributs class et id, il est temps de voir d'autres sélecteurs :

### a) Appliquer le même style à plusieurs éléments

Il est possible d'appliquer le même style à plusieurs éléments à la fois, il suffit de séparer les sélecteurs de ces éléments par des virgules.

Prenons l'exemple suivant :

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Page 1</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
<h1 >AVANT PROPOS </h1>

<h2>WISSEN CORPORATION </h2>

<p>C'est un groupe formé essentiellement de meilleurs étudiants de nos groupes
d'étude et d'encadrement qui se sont fixés plusieurs objectifs dont le premier
est celui du partage de la connaissance en Informatique et en mathématique.
</p>

<h2> A PROPOS </h2>
<p> Je m'appelle Aimé DIUMI DIKOLO, <br/>
je suis co-fondateur de Wissen Corporation</p>
</body>
</html>
```

Si on veut que le titre AVANT PROPOS et tous les paragraphes soient en blue, il suffit d'écrire dans le fichier css

Forme.css

```
p, h1
{
  color: blue;
}
```

Le résultat :

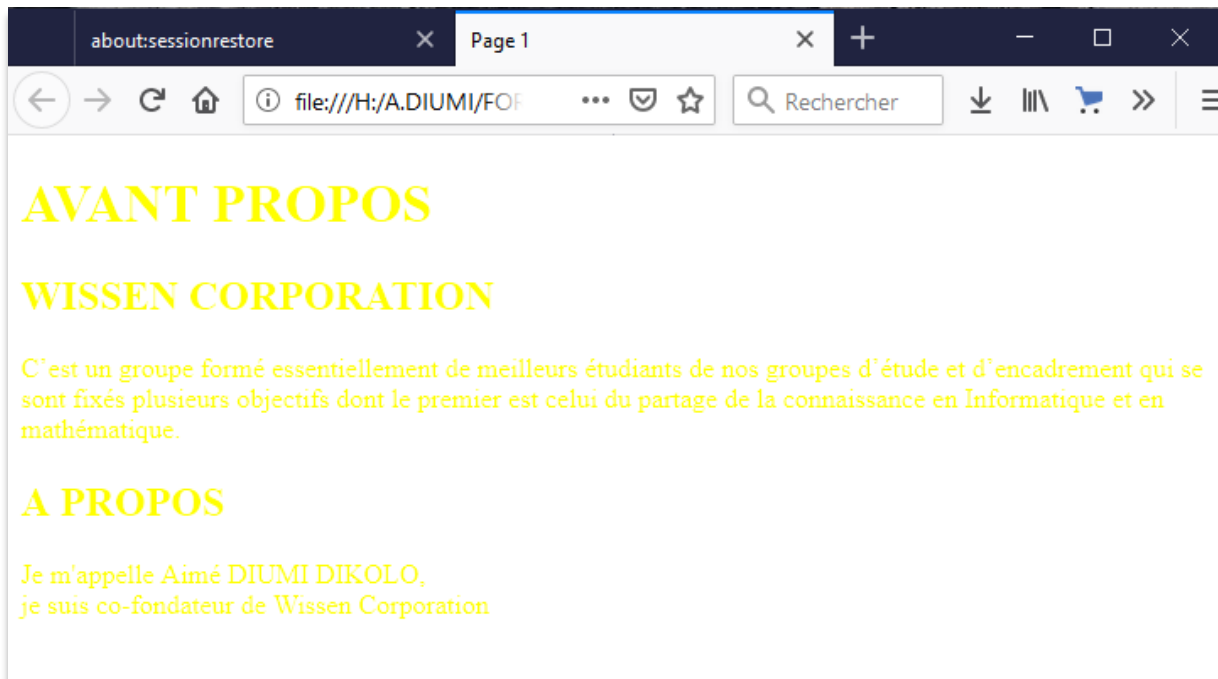


### b) \* : Le sélecteur universel

\* est un sélecteur universel, il sélectionne tous les éléments. Reprenons le code html ci-dessous, modifions seulement le fichier css :

```
*  
{  
    color: yellow;  
}
```

Le résultat : tout est en jaune :



Il est souvent utilisé pour mettre le compteur à zéro, par exemple le cas des marges (étant donné que tous les navigateurs n'ont pas les mêmes marges par défaut pour tous les éléments)

### c) A B : sélectionne une balise B contenue dans A

Prenons cet exemple :

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Page 1</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
<h1 >AVANT PROPOS </h1>

<h2><span>WISSEN </span> CORPORATION </h2>

<p>C'est un groupe formé essentiellement de <span>meilleurs étudiants</span> d
e nos groupes d'étude et d'encadrement qui se sont fixés plusieurs <span>objec
tifs</span> dont le premier est celui du partage de la connaissance en Informa
tique et en mathématique. </p>

<h2> A PROPOS </h2>
<p> Je m'appelle Aimé DIUMI DIKOLO, <br/>
```

```
je suis co-fondateur de Wissen Corporation</p>
</body>
</html>
```

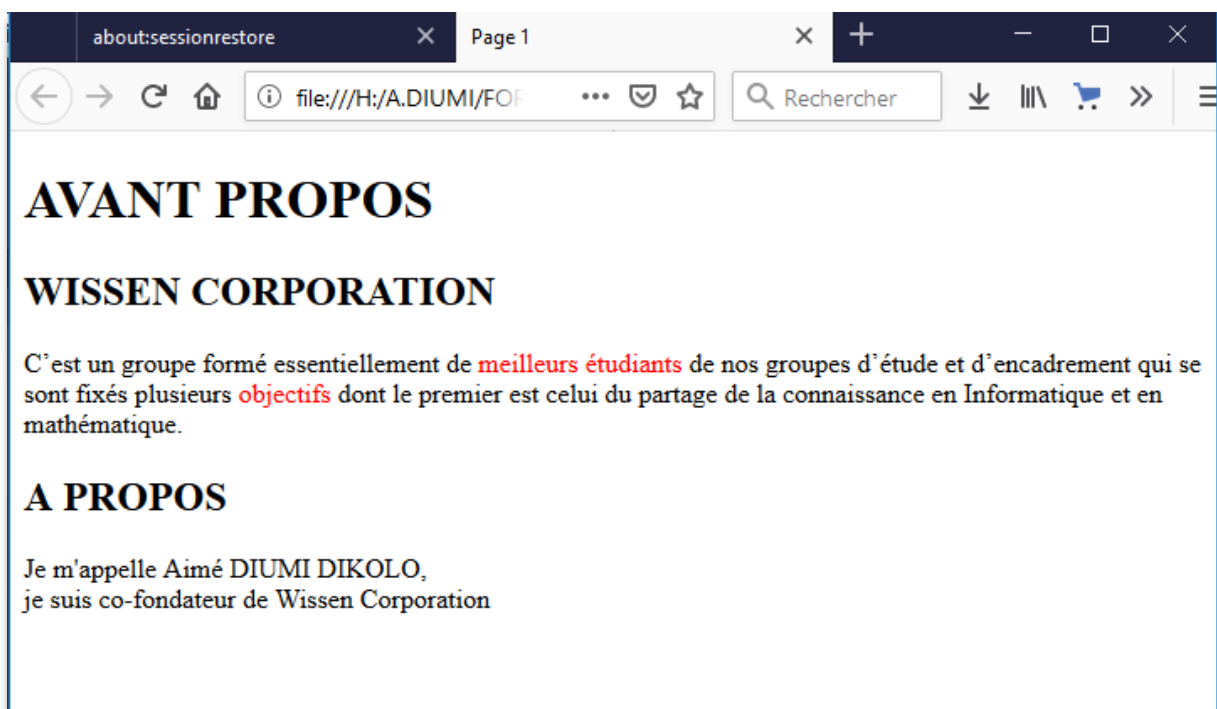
Les balises span encadrent les mots suivants : WISSEN, meilleurs étudiants et objectifs.

Forme.css

```
p span
{
    color: red;
}
```

En écrivant ceci : seules les balises span contenues dans p seront en rouge. Les autres span ne sont pas concernées.

Voici le résultat :



Il y a que deux span qui sont en rouge, l'autre span n'est pas contenue dans p.

## d) A+B : une balise B qui suit la balise A

Sélectionne la première balise B qui vient juste après la balise A

Essayons cet exemple :

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Page 1</title>
    <link rel="stylesheet" href="forme.css" />
  </head>
  <body>
<h1 >AVANT PROPOS </h1>
<p> Avant d'aborder le sujet du livre, parlons un peu de Wissen</p>
<h2>WISSEN CORPORATION </h2>
<p>C'est un groupe formé essentiellement de meilleurs étudiants de nos groupes
  d'étude et d'encadrement qui se sont fixés plusieurs objectifs dont le premier
  est celui du partage de la connaissance en Informatique et en mathématique.
</p>

<p> Je m'appelle Aimé DIUMI DIKOLO, <br/>
je suis co-fondateur de Wissen Corporation</p>

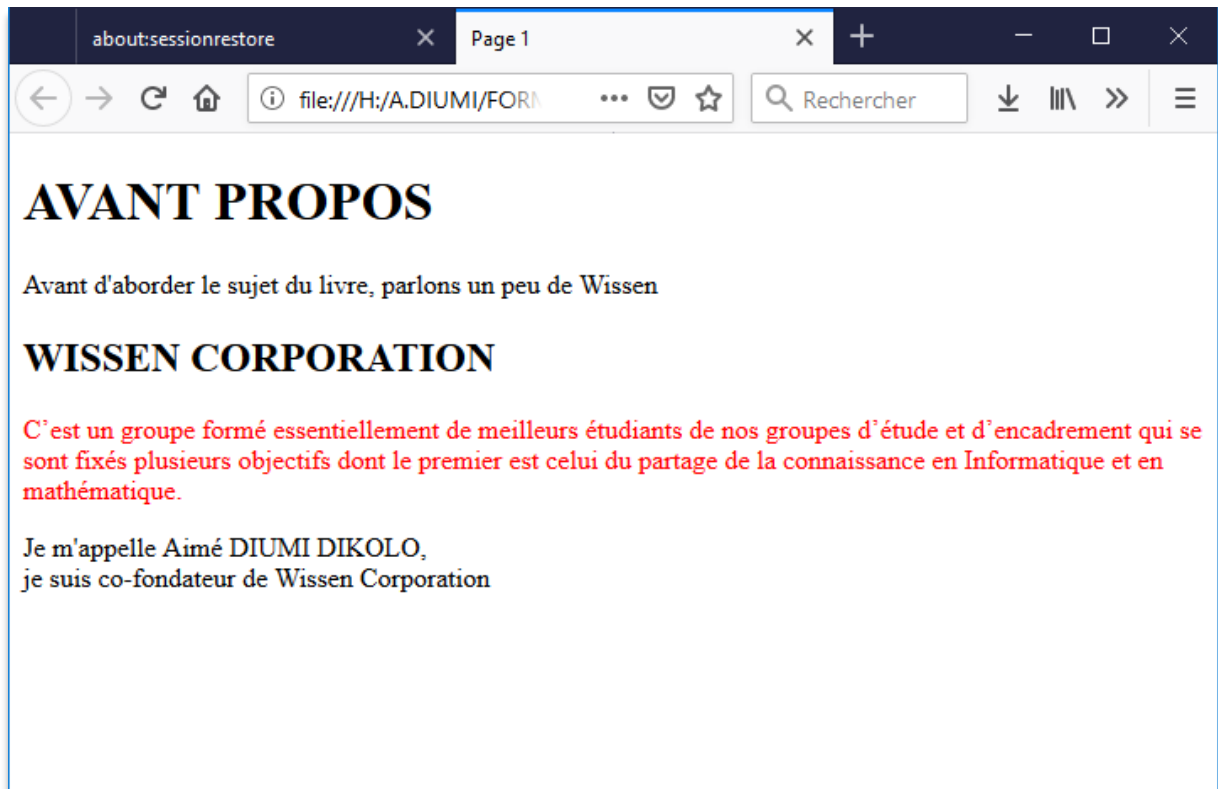
</body>
</html>
```

Forme.css

```
h2+p|
{
  color: red;
}
```

Seule la première balise p qui suit h2 sera en rouge

Résultat :



La première balise p n'a pas été modifiée car elle vient avant h2, le dernier paragraphe est intact car la balise vient à la deuxième position après h2, seule p qui vient juste après h2 sera en rouge.

### e) A[Attribut] : sélectionne une balise contenant un attribut donné

Exemple :

Formation.html

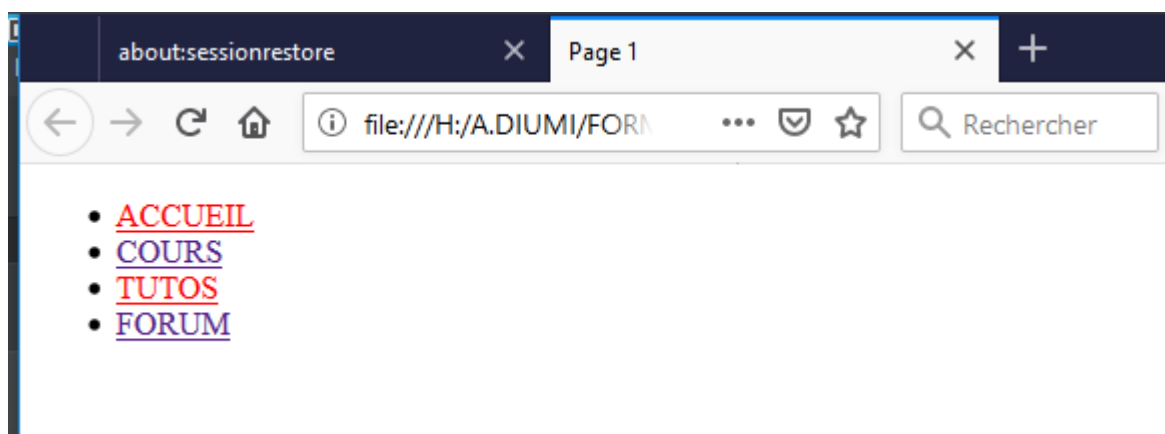
```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Page 1</title>
    <link rel="stylesheet" href="forme.css" />
  </head>
  <body>
<ul>
  <li> <a href="" title="Voir Accueil" >ACCUEIL< /a></li>
  <li>   <a href="" >      COURS </a></li>
  <li> <a href="" title="Meilleurs tutoriels" > TUTOS</a></li>
    <li> <a href="" > FORUM </a></li>
  </ul>
```

```
</body>
</html>
```

Forme.css

```
a[title]
{
    color: red;
}
```

Résultat :



Seuls ACCUEIL et TUTOS sont en rouge car ce sont les deux balises <a> qui ont l'attribut title.

## f) A[Attribut="Valeur" ]

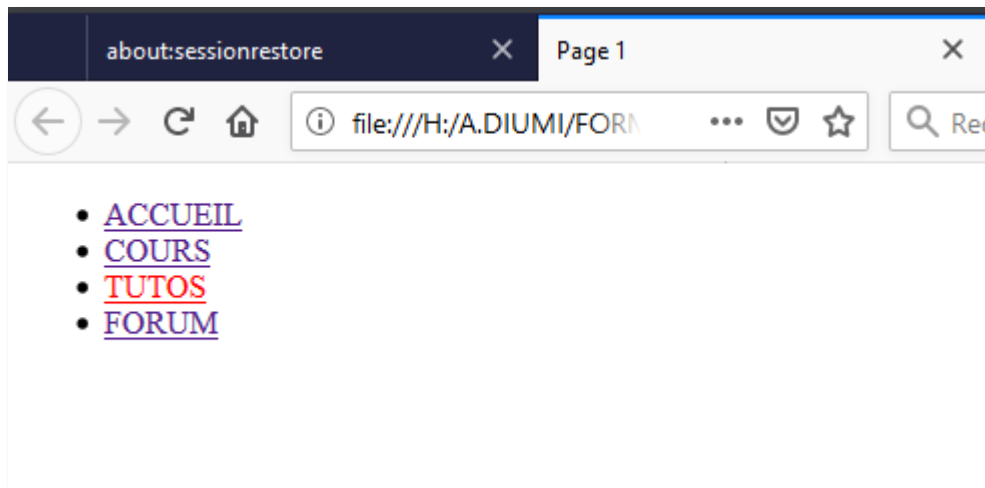
Sélectionne seulement les balises qui ont un attribut et une valeur précise.

Dans l'exemple précédant, si on modifiait le code css :

Forme.css

```
a[title="Meilleurs tutoriels"]
{
    color: red;
}
```

Résultat :



Seul TUTOS est en rouge car c'est la seule balise qui a l'attribut title ayant pour valeur "Meilleurs tutoriels".

### g) A[attribut\*= "Val"]

Ici Val doit figurer dans la valeur de l'attribut.

Exemple :

Formation.html

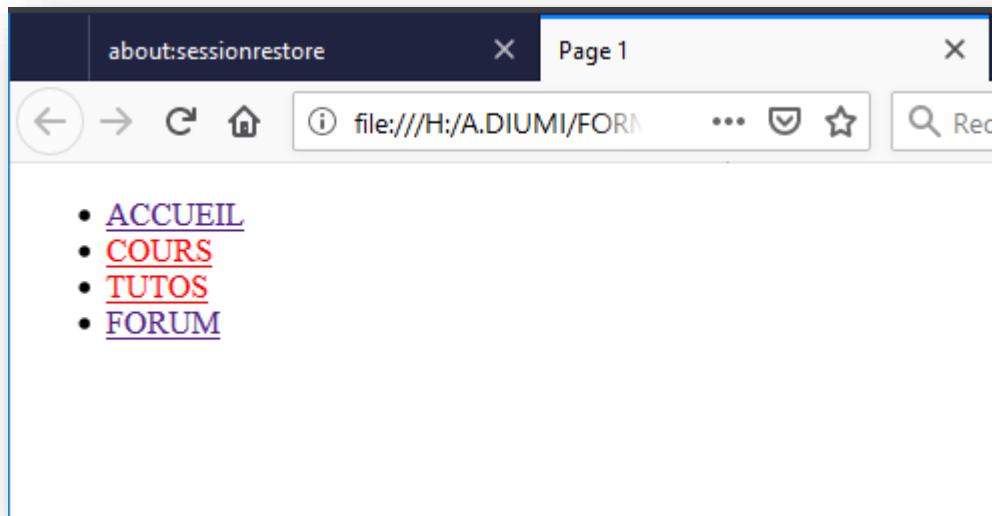
```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Page 1</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
  <ul>
    <li> <a href="" title="Voir Accueil" >ACCUEIL< /a></li>
    <li> <a href="" title="Meilleurs cours" > COURS </a></li>
    <li> <a href="" title="Meilleurs tutoriels" > TUTOS</a></li>
    <li> <a href="" > FORUM </a></li>
  </ul>
</body>
</html>
```

Forme.css

```
a[title*="Meilleurs"]
{
  color: red;
}
```



Résultat :



Le style s'applique aux liens COURS et TUTOS car dans la valeur de leur attribut title, on retrouve le mot **Meilleurs**

## h. A > B : sélectionne B qui est enfant (descendant) direct de A

Prenons l'exemple suivant :

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
    <div class="bloc_principal">
      <p> Premier paragraphe </p>
      <p> Deuxième paragraphe </p>
      <div class="bloc_secontaire" >
        <p> Troisième paragraphe </p>
        <p> Quatrième paragraphe </p>
      </div>
    </div>
  </body>
</html>
```

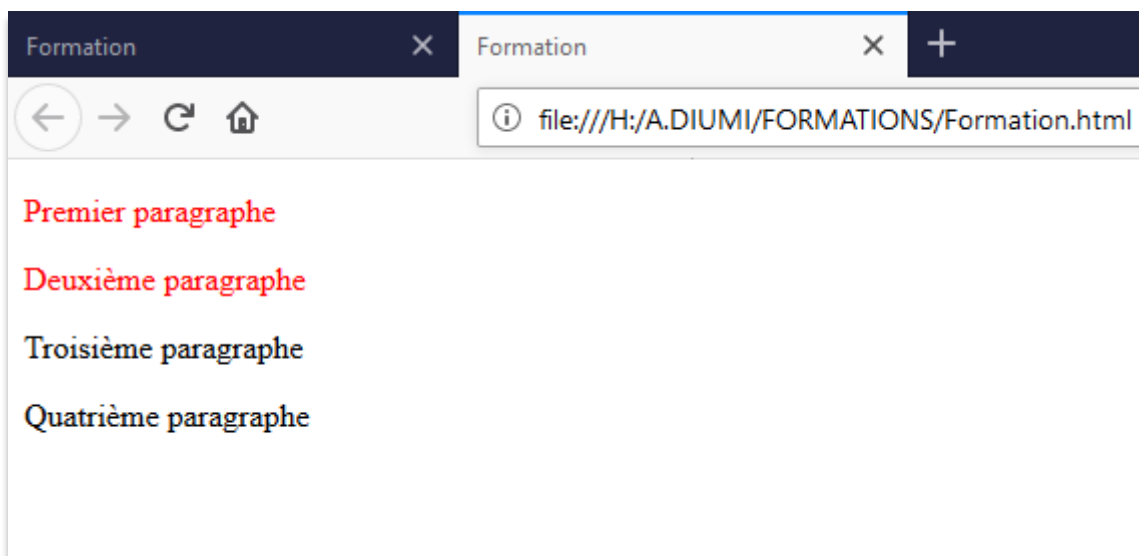
Le premier et le deuxième paragraphe ainsi que le bloc secondaire sont les enfants directs de bloc\_principal tandis que le troisième et quatrième paragraphe ne sont pas les enfants directs de bloc\_principal car ils sont contenus dans le bloc\_secondaire qui est lui aussi contenu dans bloc\_principal...On peut dire que le bloc\_principal est le grand père du troisième et quatrième paragraphe. En appliquant le code css suivant :

Forme.css

```
.bloc_principal>p
{
color: red;
}
```

Le style s'appliquera seulement aux paragraphes (p) qui sont enfants directs de bloc\_principal.

Résultat :



- i. **A~B** : sélectionne tous les éléments B précédés par l'élément A.

En d'autres termes, ce sélecteur permet de sélectionner un ou plusieurs éléments qui viennent après un élément donné.

Exemple

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
<h1> VISUAL BASIC </h1>
<p> Premier paragraphe </p>
<p> Deuxième paragraphe </p>

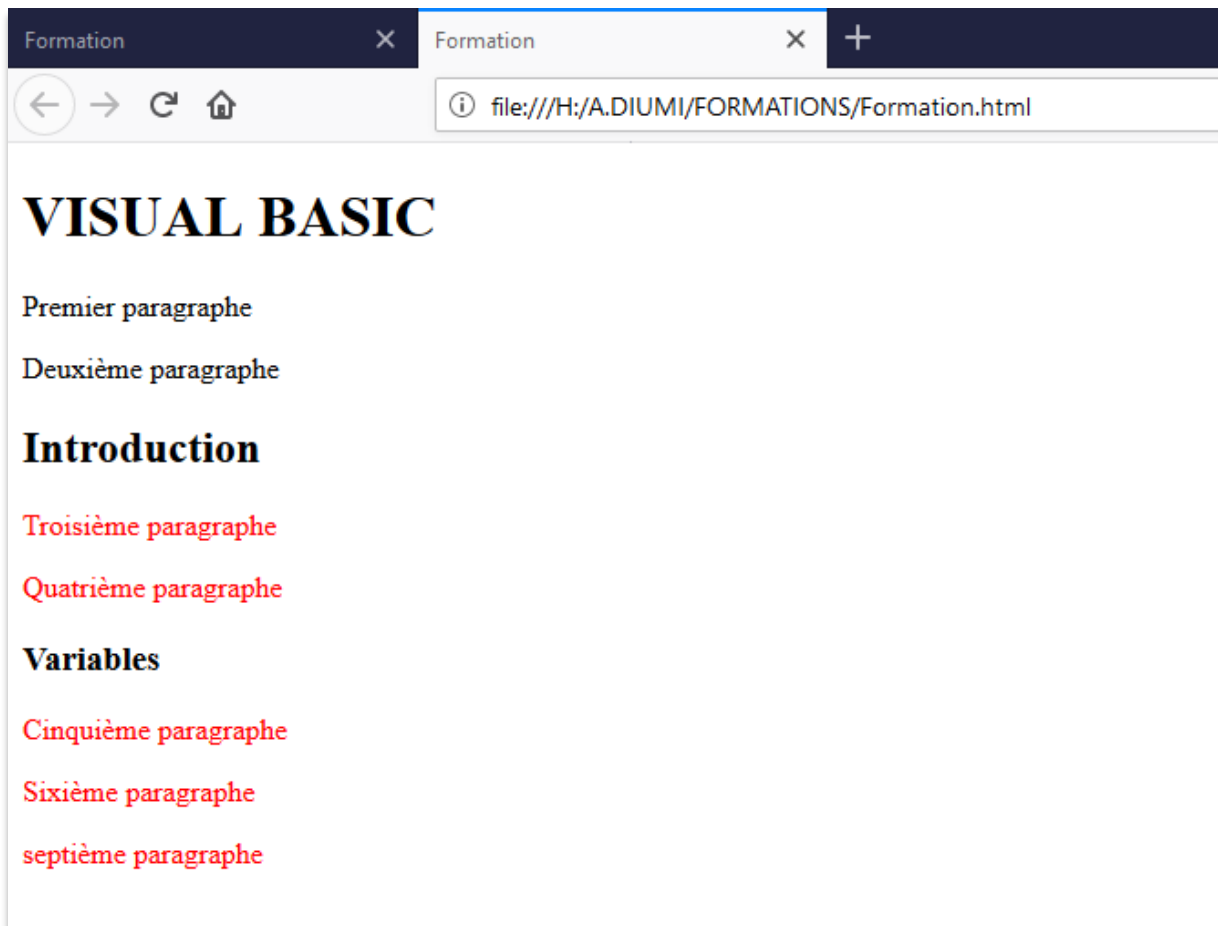
<h2> Introduction </h2>
<p> Troisième paragraphe </p>
  <p> Quatrième paragraphe </p>
<h3> Variables </h3>
<p> Cinquième paragraphe </p>
<p> Sixième paragraphe </p>
<p> septième paragraphe </p>

</body>
</html>
```

Forme.css

```
h2~p
{
color: red;
}
```

Résultat :



Le style n'est pas appliqué aux premiers et deuxièmes paragraphes car ils viennent avant h2.

## j. A[b^="c"]

Ce sélecteur permet de sélectionner tous les éléments A dont l'attribut b a une valeur commençant par "c".

Exemple

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>

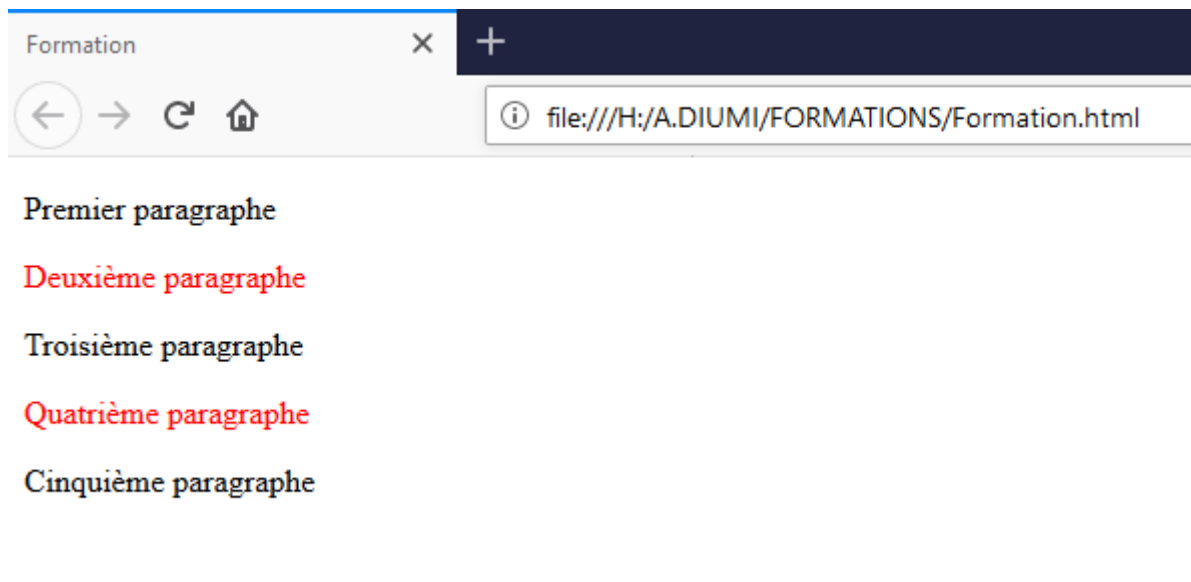
  <p class="para1"> Premier paragraphe </p>
  <p class="Important1"> Deuxième paragraphe </p>
  <p class="para2"> Troisième paragraphe </p>
  <p class="Important2"> Quatrième paragraphe </p>
```

```
<p id="para3"> Cinquième paragraphe </p>
</body>
</html>
```

Forme.css

```
p[class^="Important"]
{
    color: red;
}
```

Résultat :



Parmi les cinq paragraphes, il y a quatre qui ont l'attribut **class** et parmi les quatre, il n'y a que deux (le deuxième et quatrième) dont la valeur de l'attribut class commencent par "Important".

### k. A[b\$="c"]

Ce sélecteur permet de sélectionner tous les éléments A dont l'attribut b a une valeur finissant par "c".

### l. A[b\*="c"]

Ce sélecteur permet de sélectionner tous les éléments A dont l'attribut b a une valeur contenant "c". Notez que c représente une chaîne quelconque.

## II.3 NOTION D'HERITAGE EN CSS

En HTML si un élément B par exemple est contenu dans un autre élément A, l'élément A sera appelé l'élément parent et l'élément B est l'élément fils par rapport à l'élément A.

L'héritage est une notion extrêmement importante en CSS et cela veut simplement dire que si on applique un style à un élément parent, les éléments fils vont aussi hériter ce style : c'est le fameux « Cascading »(pour cascade).

Exemple

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel="stylesheet" href="forme.css" />

  </head>
  <body>
    <div>
      <h1> Visual Basic </h1>
      <p> Visual basic est un langage de programmation,...</p>
    </div>
  </body>
</html>
```

Forme.css

```
div
{
  color: red;
}
```

Nous avons appliqué le style à l'élément div mais voici le résultat :



Le style s'est appliqué au titre et au paragraphe car ils sont des enfants de l'élément div.

Si vous appliquez un autre style à l'élément enfant, ce style va dominer le style de l'élément parent.

Exemple

Laissons color : red pour div et ajoutons color : blue pour p :

Forme.css

```
div
{
  color: red;
}

p
{
  color: blue;
}
```

Résultat :



Le style de l'élément enfant (p) a dominé sur le style de l'élément parent(div). Notez au moins que cette domination concerne le cas où il y a contradiction entre le style de l'élément parent et l'élément enfant. Par exemple, dans notre cas, on a appliqué la couleur rouge à div (parent) et la couleur bleue à p(l'enfant). Au cas où il n'y a pas contradiction, l'élément enfant aura les styles définis pour lui ainsi que les styles définis au niveau de l'élément parent.

Exemple

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel="stylesheet" href="forme.css" />
  </head>
  <body>
    <div>
      Salut, bienvenu!!!
      <h1> Visual Basic </h1>
      <p> Visual basic est un langage de programmation,...</p>
    </div>
  </body>
</html>
```

Forme.css

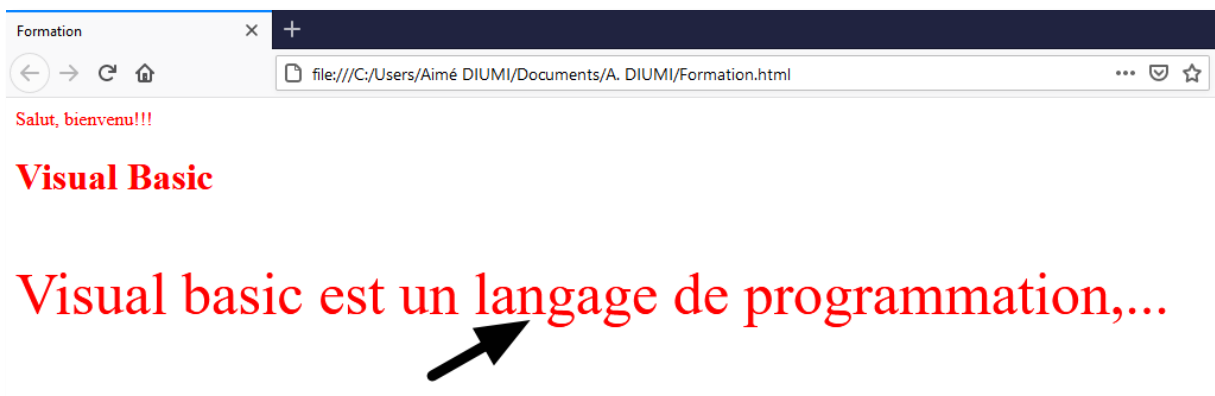
```
div
{
  color: red;
```



```
}  
  
p  
{  
font-size: 50px;  
}
```

Au niveau de div (parent), on a défini que le texte soit en rouge et au niveau de p (enfant), on a fixé la taille du texte à 50px.

Résultat :



Style de div (parent) et de p (enfant) sont appliqués à p (enfant) car il n'y a pas contradiction.

## II.4 MISE EN FORME DU TEXTE

### II.4.1 La taille du texte

On utilise la propriété **font-size** pour changer la taille du texte. On peut indiquer de façon précise (cm, mm ou px) ou de façon relative (en %, em ou ex). Cette dernière technique étant la plus souple car elle s'adapte facilement aux préférences des visiteurs.

Illustration

Formation.html

```
<!doctype html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <title> Page 1</title>  
    <link rel=stylesheet href="forme.css" />  
  </head>  
  <body>  
<h1> LES BASES DU LANGAGE ET MODE CONSOLE </h1>
```

```
<p class="paragraphe1">Wikipédia présente Visual basic comme un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, créé par Microsoft pour son modèle COM. <br/>Visual basic est directement dérivé du BASIC (acronyme pour Beginner's All purpose Symbolic Instruction Code ) et permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôle ou objets Active X. </p>
```

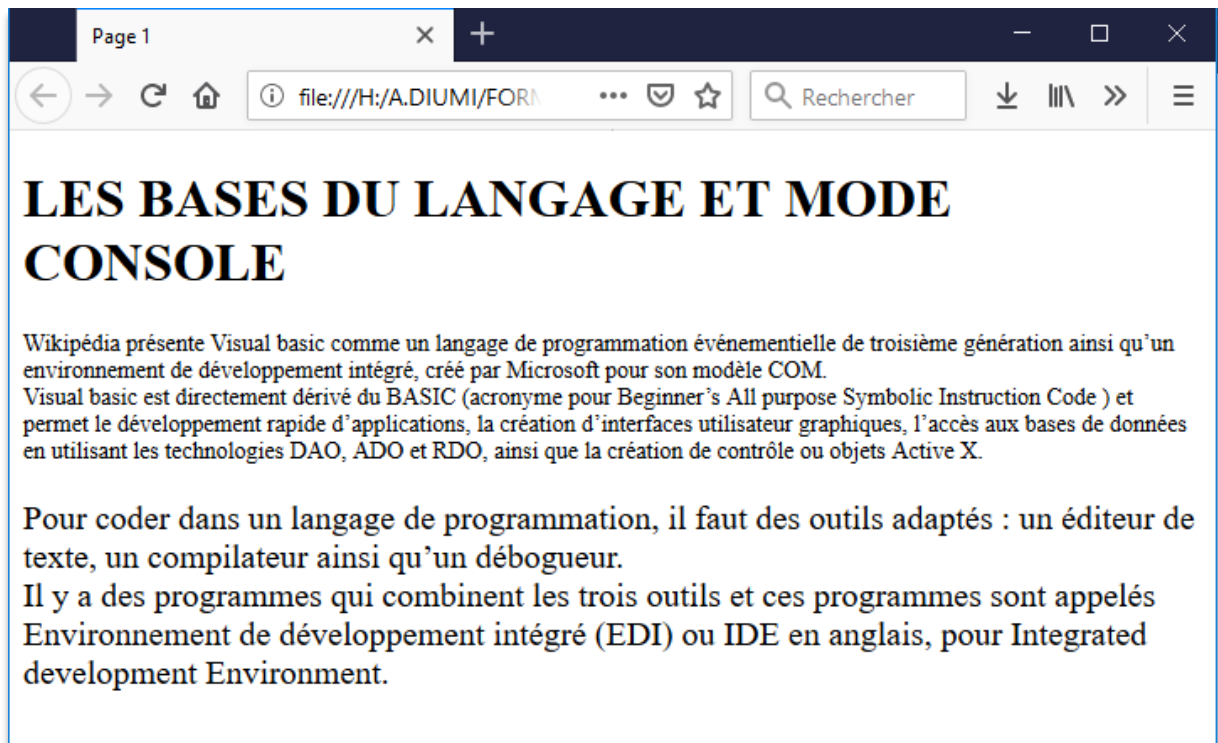
```
<p class="paragraphe2">Pour coder dans un langage de programmation, il faut des outils adaptés : un éditeur de texte, un compilateur ainsi qu'un débogueur.<br/> Il y a des programmes qui combinent les trois outils et ces programmes sont appelés Environnement de développement intégré (EDI) ou IDE en anglais, pour Integrated development Environment. </p>
```

```
</body>  
</html>
```

#### Forme.css

```
.paragraphe1  
{  
    font-size: 14px;  
}  
  
.paragraphe2  
{  
    font-size: 20px;  
}
```

Le résultat :



A part %, em et ex, il y a une autre méthode qui permet d'indiquer une valeur relative : écrire la taille avec des mots en anglais suivants :

xx-small : minuscule ;

x-small : très petit ;

small : petit ;

medium : moyen ;

large : grand ;

x-large : très grand ;

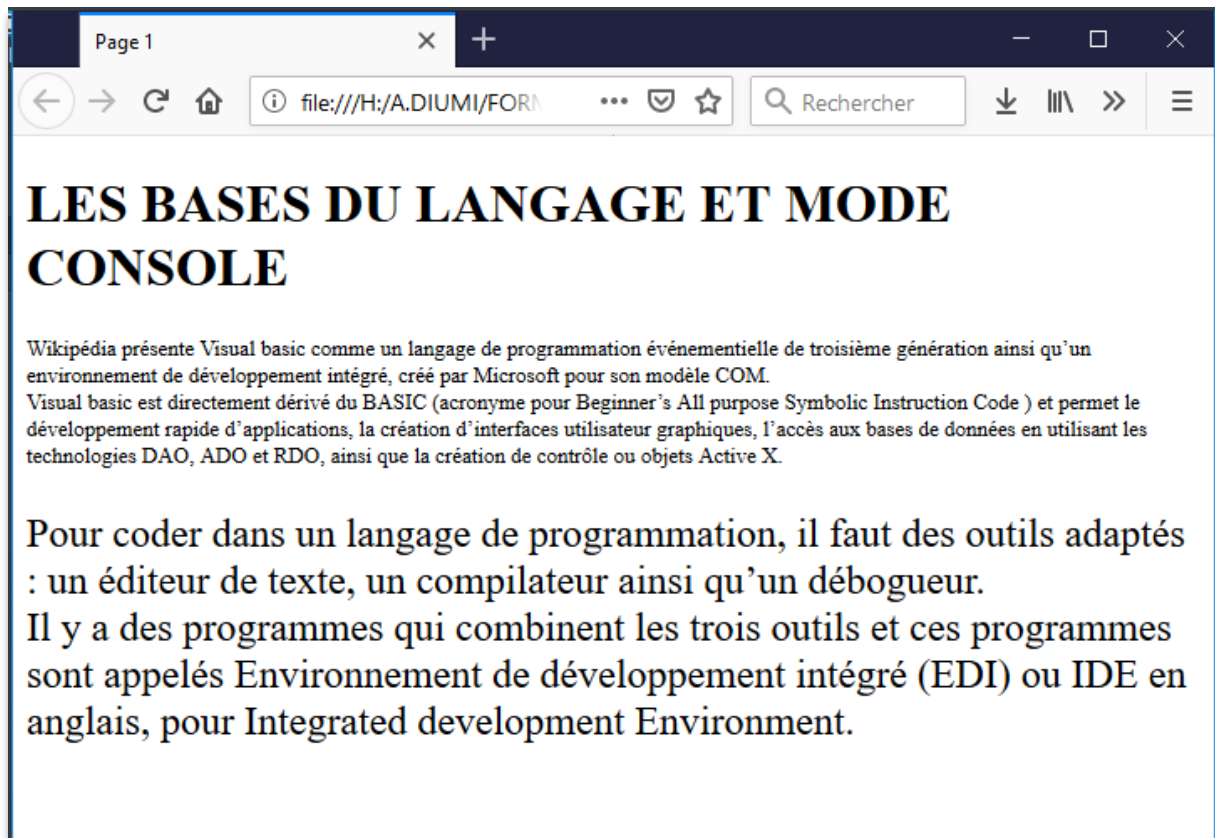
xx-large : gigantesque.

Modifions le code css précédent :

```
.paragraphe1
{
    font-size: small;
}

.paragraphe2
{
    font-size: x-large;
}
```

Le résultat :



### 11.4.2 La police

Pour définir la police, on utilise la propriété `font-family`. Ci-dessous, je vous présente quelques polices reconnus par la plupart des navigateurs.

ALGERIAN

Arial

Arial Black

Comic Sans MS

Courier New

Georgia

Impact Monotype

Symbol Monotype

Times New Roman

Trebuchet MS

Verdana

## Exemple

### Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
    <p class="p1"> Texte en ALGERIAN</p>
<p class="p2"> Texte en Arial </p>
<p class="p3"> Texte en Arial Black </p>
<P class="p0k"> Texte en Bookman Old</P>
<p class="p4"> Texte en Comic Sans MS</p>
<p class="p5"> Texte en Courier New</p>
<p class="p6"> Texte en Georgia</p>
<p class="p7"> Texte en Impact </p>
<p class="p8"> Texte en Symbol Monotype </p>
<p class="p9"> Texte en Times New Roman</p>
<p class="p10"> Texte en Trebuchet MS</p>
<p class="p11"> Texte en Verdana</p>

</body>
</html>
```

### Forme.css

```
.p1
{
  font-family: ALGERIAN;
}

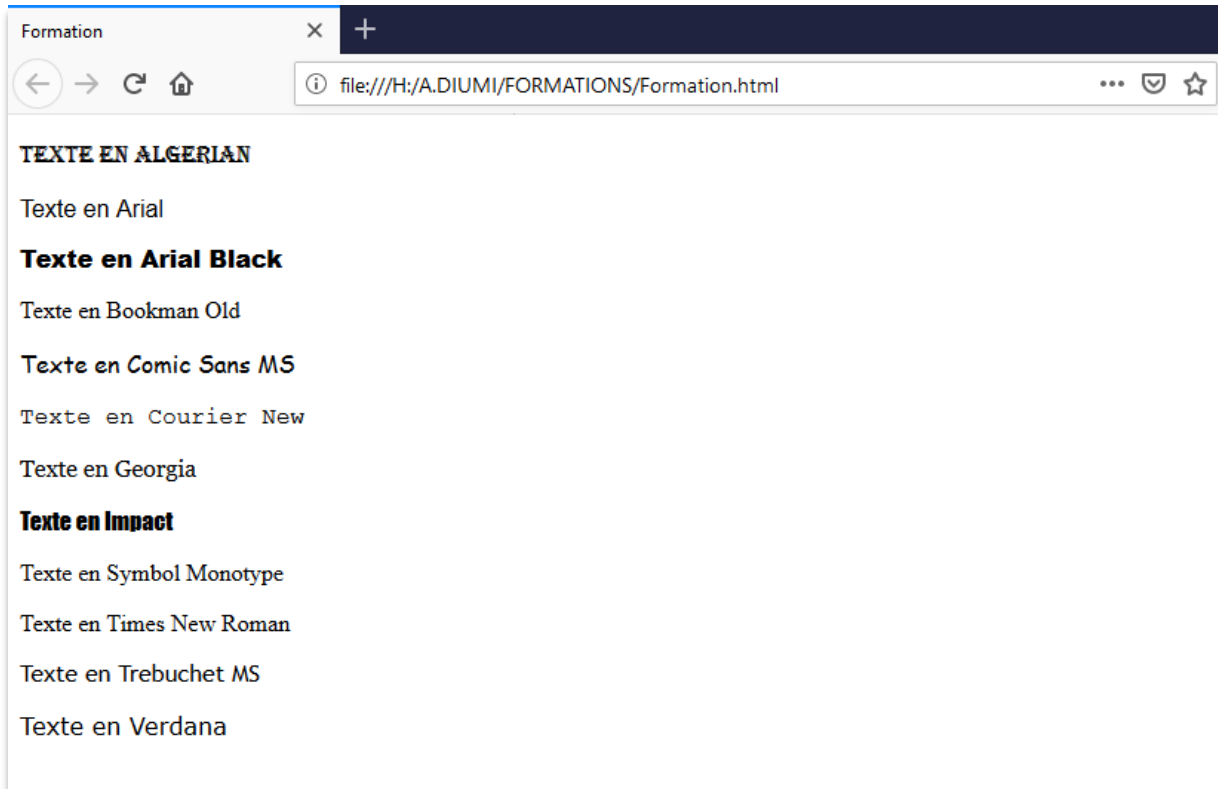
.p2
{
  font-family:arial;
}

.p3
{
  font-family: "Arial Black";
}

.p4
{
  font-family:"Comic Sans MS";
}
```

```
.p5
{
  font-family:"Courier New" ;
}
.p6
{
  font-family:Georgia;
}
.p7
{
  font-family:Impact ;
}
.p8
{
  font-family: "Symbol Monotype";
}
.p9
{
  font-family:"Times New Roman";
}
.p10
{
  font-family:"Trebuchet MS";
}
.p11
{
  font-family: Verdana;
}
.p12
{
  font-family:Webdings;
}
.p0k
{
  font-family: "Bookman Old";
}
```

Résultat :



Sur le Web, vous ne pouvez jamais être sûr des polices qu'auront vos lecteurs sur leurs ordinateurs, Il peut arriver que l'internaute n'ait pas la police que vous aviez définie, c'est pourquoi il est conseillé de préciser plusieurs polices séparées par des virgules :

```
sélecteur
{
    font-family: police1, police2, police3...
}
```

Nous avons ajouté des alternatives à la police1 : Le navigateur essaiera d'abord d'utiliser la police1. S'il ne l'a pas, il essaiera la police2. S'il ne l'a pas, il passera à la police3, et ainsi de suite.

En général, on indique en tout dernier serif, ce qui correspond à une police par défaut (qui ne s'applique que si aucune autre police n'a été trouvée).

## II.4.3 L'italique

On utilise la propriété **font-style** qui peut prendre une des trois valeurs suivantes :

- italic : le texte sera en italique.
- Oblique : les lettres seront penchées
- normal : le texte sera normal : c'est la valeur par défaut. Elle permet entre autres d'annuler la mise en forme en italique.

Exemple

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
    <h1> LES BASES DU LANGAGE ET MODE CONSOLE </h1>
    <p class="paragraphe1" > Wikipédia présente Visual basic comme un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, créé par Microsoft pour son modèle COM. <br/> Visual basic est directement dérivé du BASIC (acronyme pour Beginner's All purpose Symbolic Instruction Code ) et permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôle ou objets Active X. </p>
    <p class="paragraphe2" > Pour coder dans un langage de programmation, il faut des outils adaptés : un éditeur de texte, un compilateur ainsi qu'un débogueur. <br/> Il y a des programmes qui combinent les trois outils et ces programmes sont appelés Environnement de développement intégré (EDI) ou IDE en anglais, pour Integrated development Environment. </p>
  </body>
</html>
```

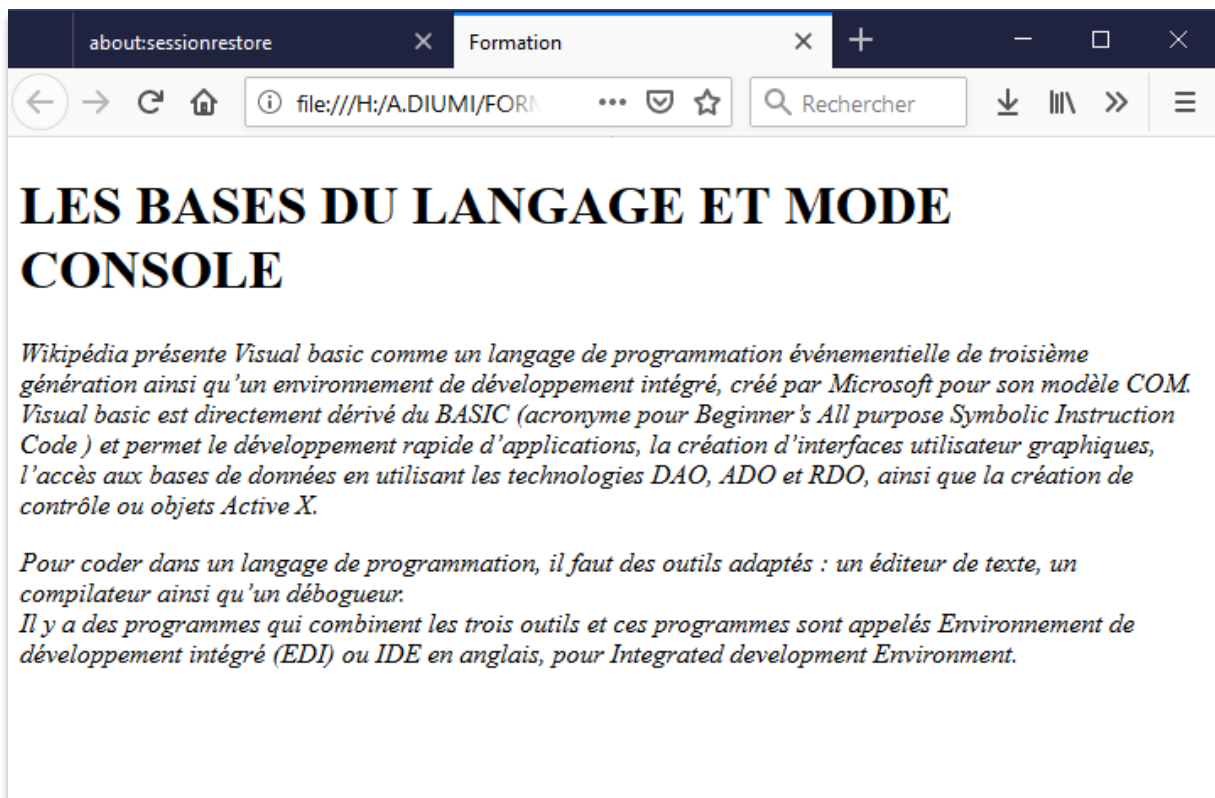


Forme.css

```
.paragraphe1
{
    font-style: italic;
}

.paragraphe2
{
    font-style: oblique;
}
```

Résultat :



Remarquez que le résultat de la valeur oblique est légèrement différente de la valeur italic.

## 11.4.4 Mettre le texte en gras

On utilise la propriété **font-weight** qui peut prendre les valeurs suivantes :

- bold : le texte sera en gras
- normal : le texte sera écrit normalement, c'est la valeur par défaut.
- Lighter (version allégée de la police) ;
- bolder le texte est encore plus en gras;
- Une centaine compris entre 100 et 900 (du plus léger au plus gras) ;
- Inherit (hérite des styles de ses parents).

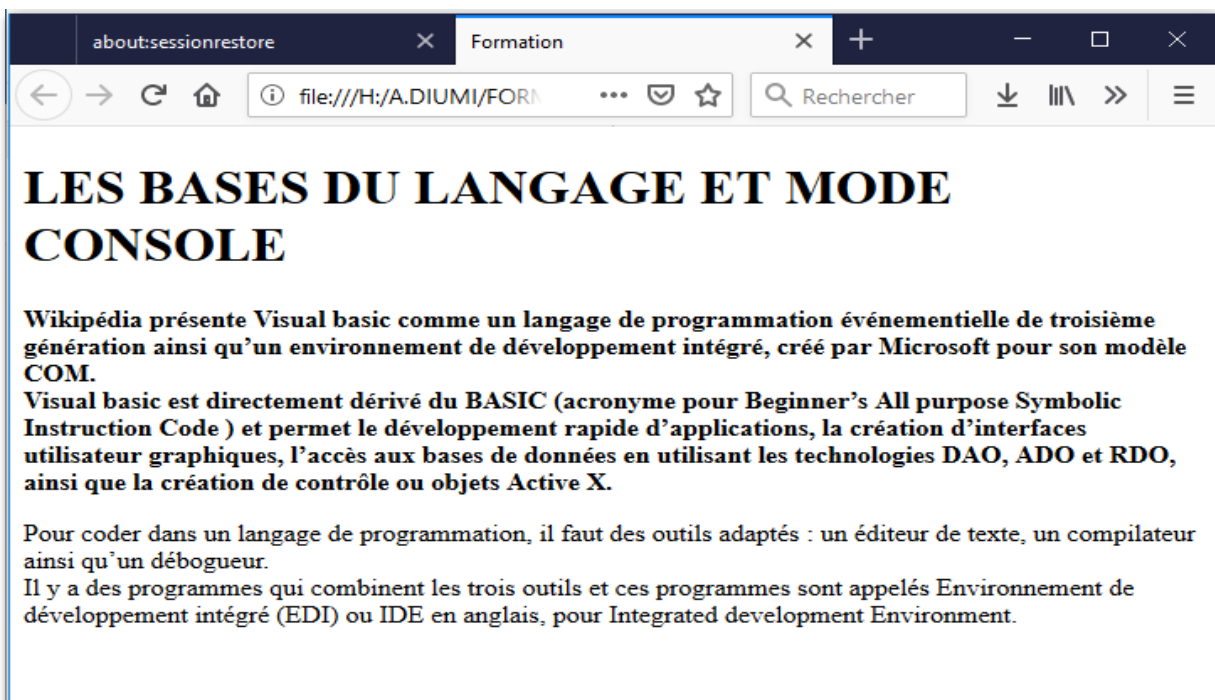
Reprenons le même exemple html, changeons seulement le code CSS

Forme.css

```
.paragraphe1
{
    font-weight: bold;
}

.paragraphe2
{
    font-style: normal;
}
```

Résultat :



## II.4. 5 Le soulignement du texte

On utilise la propriété **text-decoration** qui peut prendre l'une des valeurs suivantes :

- underline : Le texte sera souligné.
- line-through : Le texte sera barré.
- overline : ligne au-dessus du texte.
- blink : le texte va clignoter. Ne fonctionne pas sur tous les navigateurs (Internet Explorer et Google Chrome, notamment).
- none : normal : c'est la valeur par défaut.

Exemple :

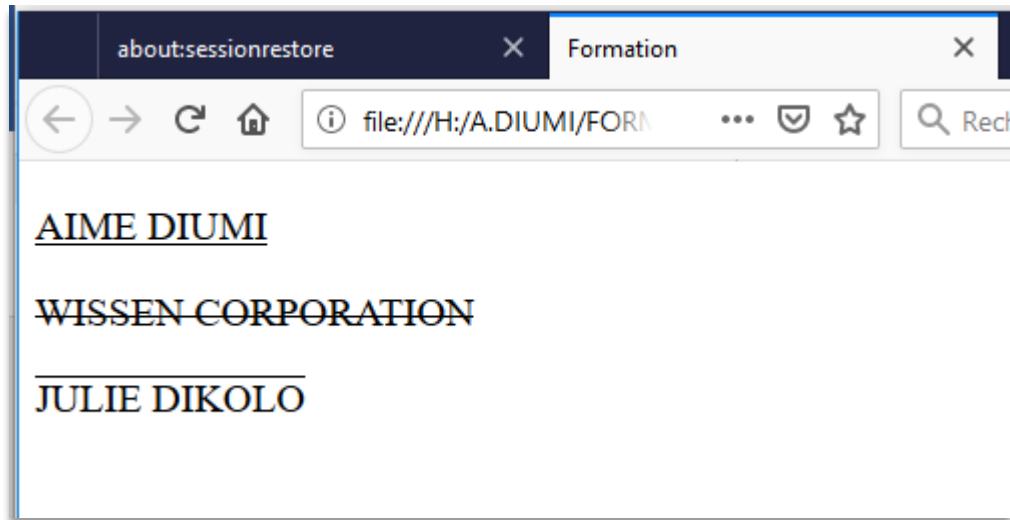
Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
    <p class="p1" > AIME DIUMI </p>
    <p class="p2" > WISSEN CORPORATION </p>
    <p class="p3" > JULIE DIKOLO </p>
  </body>
</html>
```

Forme.css

```
.p1
{
  text-decoration:underline;
}
.p2
{
  text-decoration: line-through;
}
.p3
{
  text-decoration: overline;
}
```

Aperçu :



## II.4.6 L’alignement du texte

### II.4.6.1 Alignement horizontal

On utilise la propriété **Text-align** qui peut prendre une des valeurs suivantes :

- left : le texte sera aligné à gauche (c'est le réglage par défaut).
- center : le texte sera centré.
- right : le texte sera aligné à droite.
- justify : le texte sera « justifié ». Justifier le texte permet de faire en sorte qu'il prenne toute la largeur possible sans laisser d'espace blanc à la fin des lignes.

Exemple :

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
    <h1> LES BASES DU LANGAGE ET MODE CONSOLE </h1>
    <p class="p1">Wikipédia présente Visual basic comme un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, créé par Microsoft pour son
```

modèle COM. Visual basic est directement dérivé du BASIC (acronyme pour Beginner's All purpose Symbolic Instruction Code ) et permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôle ou objets Active X. </p>

<p class="p2">Pour coder dans un langage de programmation, il faut des outils adaptés : un éditeur de texte, un compilateur ainsi qu'un débogueur. Il y a des programmes qui combinent les trois outils et ces programmes sont appelés Environnement de développement intégré (EDI) ou IDE en anglais, pour Integrated development Environment. </p>

<p class="p3">L'éditeur de texte nous permet d'écrire le code source du programme. Le compilateur (Interpréteur) pour transformer le code source en binaire qui est le seul langage compréhensible par la machine. Le débogueur permet de traquer les erreurs dans le programme. Pour Visual basic, nous pouvons utiliser : Visual studio, Visual studio Express, SharpDevelop, MonoDevelop etc. </p>

</body>  
</html>

## Forme.css

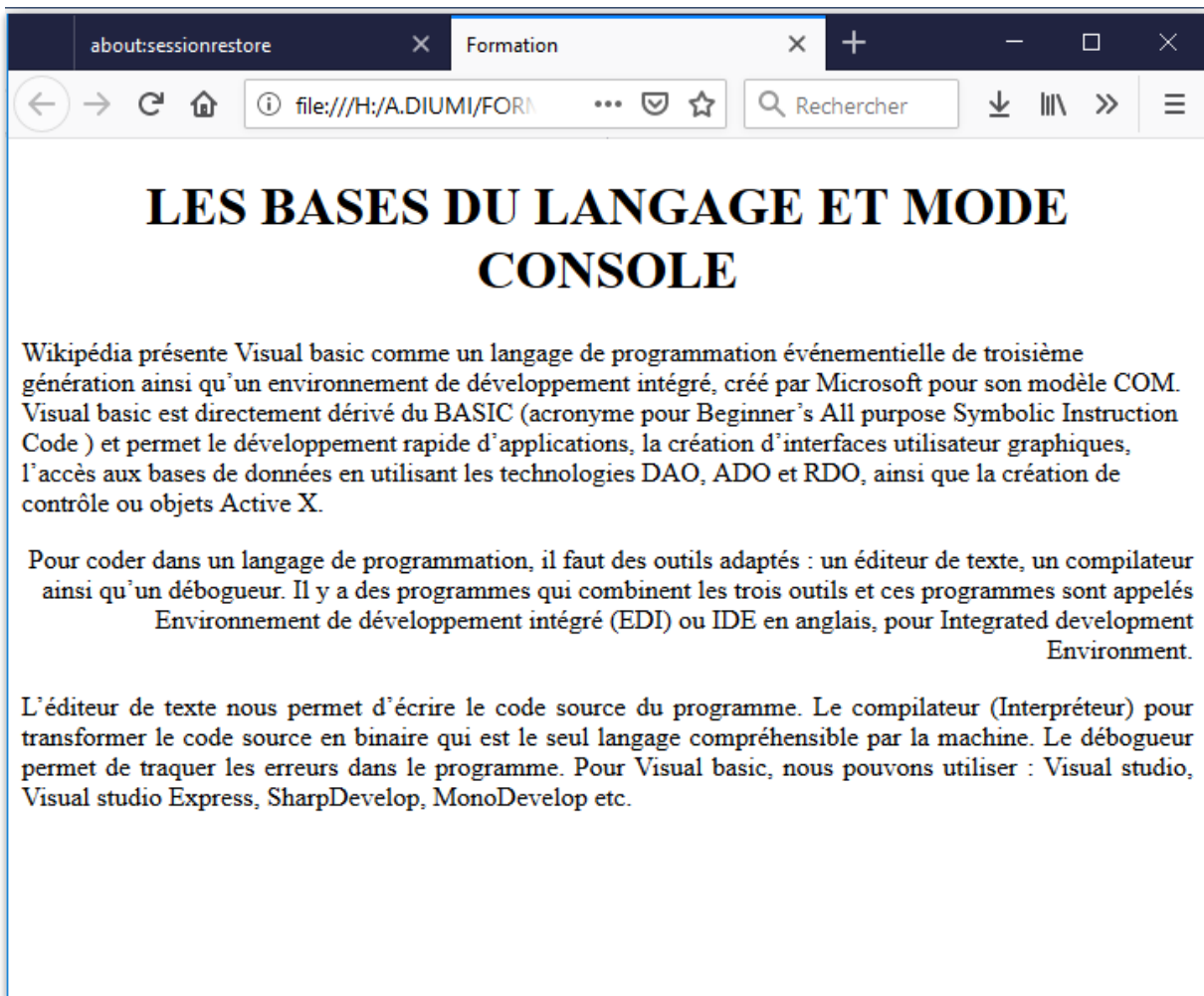
```
h1
{
text-align: center;
}

.p1
{
text-align:left;
}

.p2
{
text-align:right;
}

.p3
{
text-align: justify;
}
```

Résultat :



Il est bon de noter que le centrage ou l'alignement se fait toujours par rapport à l'élément parent le plus proche du texte.

#### *II.4.6.2 Alignement vertical*

On utilise la propriété **vertical-align** pour gérer l'alignement vertical d'un texte par rapport à un autre élément de la page. Les valeurs possibles sont :

- ❖ **baseline** : aligne le texte par rapport au bas de la ligne de texte. C'est la valeur par défaut.
- ❖ **sub** : cette valeur met le texte concernée en indice, en dessous de la ligne de base

## Exemple formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel="stylesheet" href="forme.css" />
  </head>
  <body>

  <p> H<span class="deux" >2</span>O</p>

</body>
</html>
```

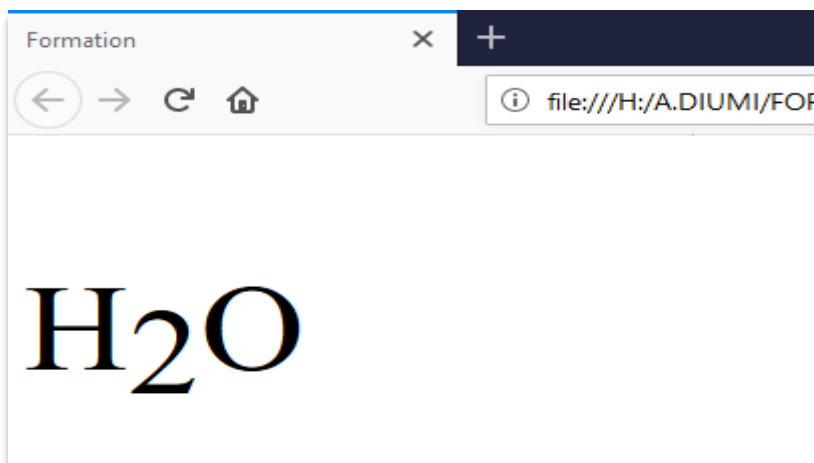
J'agrandis la taille de la police et place « 2 » en dessous de « H »

Forme.css

```
p
{
font-size: 4.5em;
}

.deux
{
vertical-align: sub;
}
```

Résultat :



- ❖ super : cette valeur met le texte concerné en exposant, soit au-dessus de la ligne de la base.

Exemple

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel="stylesheet" href="forme.css" />
  </head>
  <body>

  <p> L'équation du second degré a la forme : <br />
    <span class="equation"> ax<span class="exposant">2</span>+bx+c </span>
  </p>
</body>
</html>
```

Forme.css

```
.equation
{
font-size: 2.5em;
font-style: italic;
}
.exposant
{
vertical-align: super;
font-size: 0.8em;
}
```

Résultat :





Les trois valeurs baseline, sup et super alignent un élément par rapport à une ligne de texte, les valeurs suivantes alignent un élément relativement à l'élément parent.

- ❖ top : elle aligne par rapport au plus haut de l'élément parent.

Exemple

Gardons le même code html utilisé pour la valeur super, mettons cette fois ci top à la place et remarquez la petite différence :

Forme.css

```
.equation
{
font-size: 2.5em;
font-style: italic;
}

.exposant
{
vertical-align: top;
font-size: 0.8em;
}
```

Résultat :



- ❖ bottom : cette propriété aligne un élément au plus bas de l'élément parent.
- ❖ middle : aligne un élément par rapport au milieu d'un élément parent.

On peut aussi utiliser une valeur en pourcentage : un pourcentage positif aligne l'élément au-dessus par rapport à la ligne de base et un pourcentage négatif l'aligne en dessous par rapport à la ligne de base.

## II.5 LA COULEUR ET LE FOND

### II.5.1 La couleur du texte

Comme nous avons vu ci-haut, on utilise la propriété **Color**. Il y a plusieurs méthodes pour attribuer une valeur à cette propriété :

#### 1ere méthode :

Taper directement le nom de la couleur comme on le faisait au début.

Voici la liste de seize couleurs fondamentales :



#### 2e méthode : la notation RGB

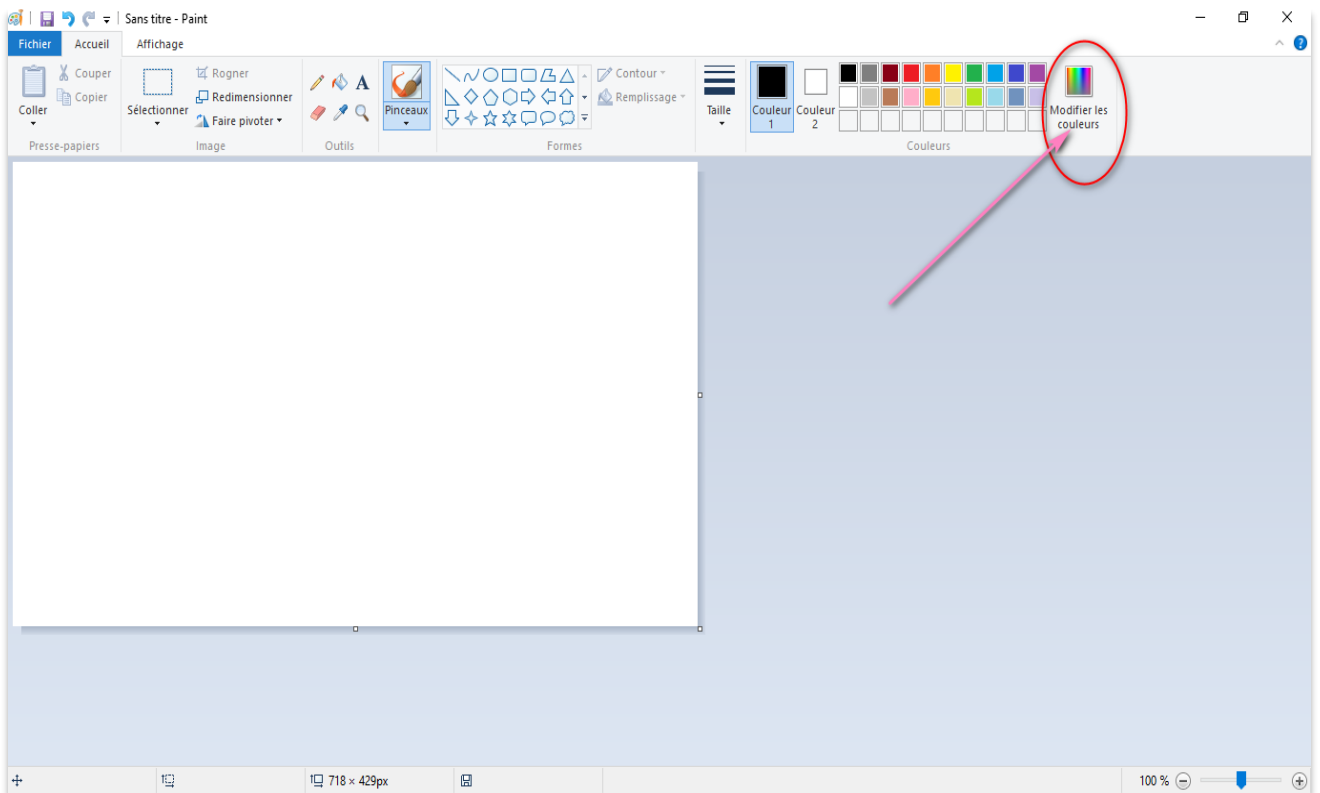
RGB signifie Red Green Blue (Rouge Vert Bleu). Avec cette méthode, on considère que les trois couleurs sont les couleurs fondamentales et toutes les autres couleurs peuvent être obtenues par la combinaison de ces trois couleurs. Sur ce, pour obtenir une couleur quelconque, il suffit de déterminer la quantité du rouge, du bleu et du vert.

Ces quantités sont des nombres allant de 0 (valeur la plus faible) à 255 (valeur la plus forte). Par exemple pour le blanc (considéré comme la présence de toutes les couleurs) on a : RGB (255,255,255) et pour le noir (absence de toutes les couleurs) : RGB (0,0,0)

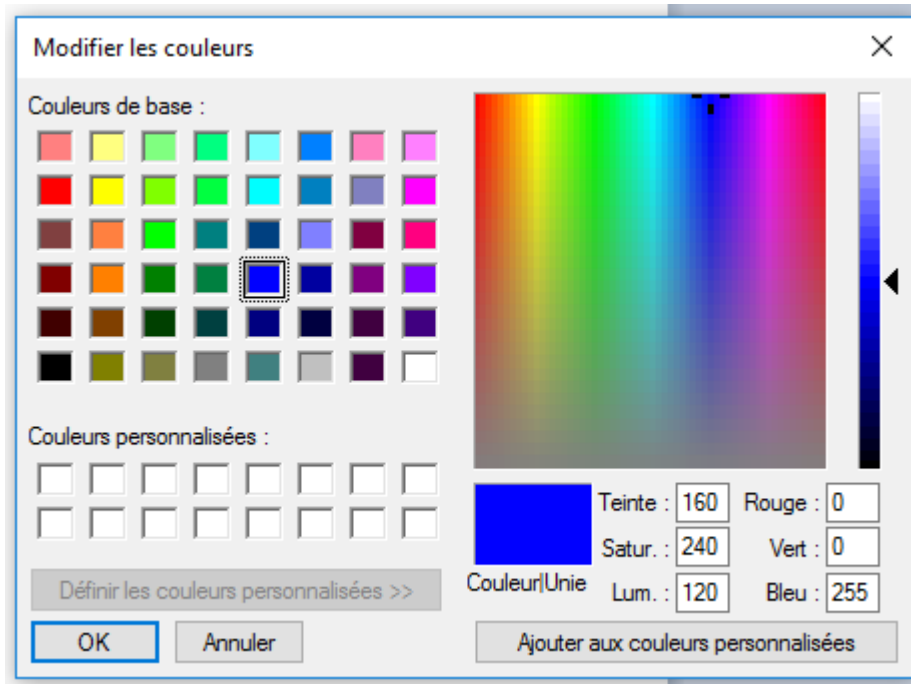
Parfois, il est difficile de connaître la combinaison RGB de la couleur souhaitée, heureusement qu'il a des logiciels qui nous permettent de trouver facilement la combinaison RGB d'une couleur. Par exemple Paint, Trac, etc.

Voici la procédure pour trouver la composition RGB d'une couleur avec le logiciel Paint (logiciel incorporé dans Windows)

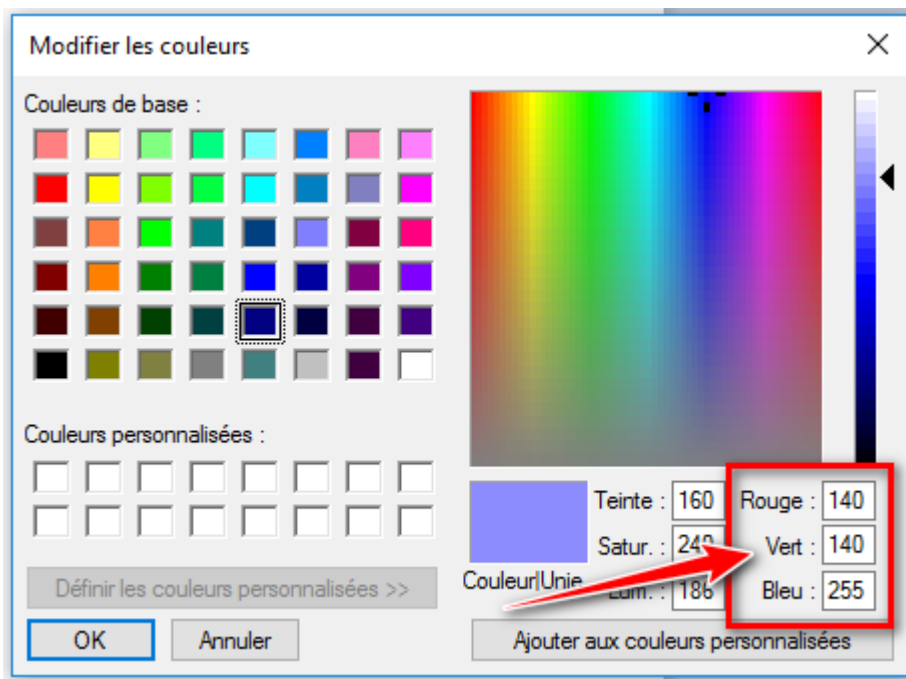
- Lancez Paint depuis le menu Démarrer
- Rendez-vous dans la section **Modifier les couleurs**



Une fenêtre s'ouvre : A gauche, vous avez les couleurs de base. Dans la zone qui apparaît à droite, faites bouger les curseurs pour sélectionner la couleur qui vous intéresse.



Pour chaque couleur sélectionnée, les quantités du Rouge-Vert-Bleu apparaissent en bas en droite :



Relevez les quantités de Rouge-Vert-Bleu correspondantes, indiquées en bas à droite de la fenêtre (ici 140,140,255). Recopiez ces valeurs dans cet ordre dans le fichier CSS, comme dans l'exemple ci-dessous.

```
sélecteur
{
    color: RGB(140,140,255);
}
```

### 3<sup>e</sup> méthode : La notation hexadécimale

Un nom de couleur en hexadécimal, cela ressemble à : #1A5A28. On doit toujours commencer par écrire un dièse (#), suivi de six lettres ou chiffres allant de 0 à 9 et de A à F.

Ces lettres ou chiffres fonctionnent deux par deux. Les deux premiers indiquent une quantité de rouge, les deux suivants une quantité de vert et les deux derniers une quantité de bleu. En mélangeant ces quantités (qui sont les composantes Rouge-Vert-Bleu de la couleur) on peut obtenir la couleur qu'on veut.

En pratique ces deux chiffres ou lettres représentent un nombre en hexadécimal, ce nombre varie de 00 (0 en décimal) à FF (correspond en 255 en décimal). Pour plus des détails sur la conversion entre système décimal et hexadécimal, voir mon cours sur [Les systèmes de numération](#)

Exemple : pour le blanc, on aura #FFFFFF, pour le noir #000000.

Certains logiciels de dessin, comme Photoshop et Gimp, vous indiquent les couleurs en hexadécimal. Il vous est alors facile de copier-coller le code hexadécimal d'une couleur dans votre fichier CSS. Pour ceux qui maîtrisent la conversion décimal-hexadécimal, vous pouvez récupérer les combinaisons Rouge-Vert-Bleu de la couleur souhaitée en décimal grâce au logiciel Paint et de les convertir en hexadécimal.

Sachez que les deux dernières méthodes (notation hexadécimale et la méthode RGB) sont presque identiques à une différence près : La notation hexadécimale donne la composition Rouge-Vert-Bleu en hexadécimal tandis que la méthode RGB la donne en décimal.

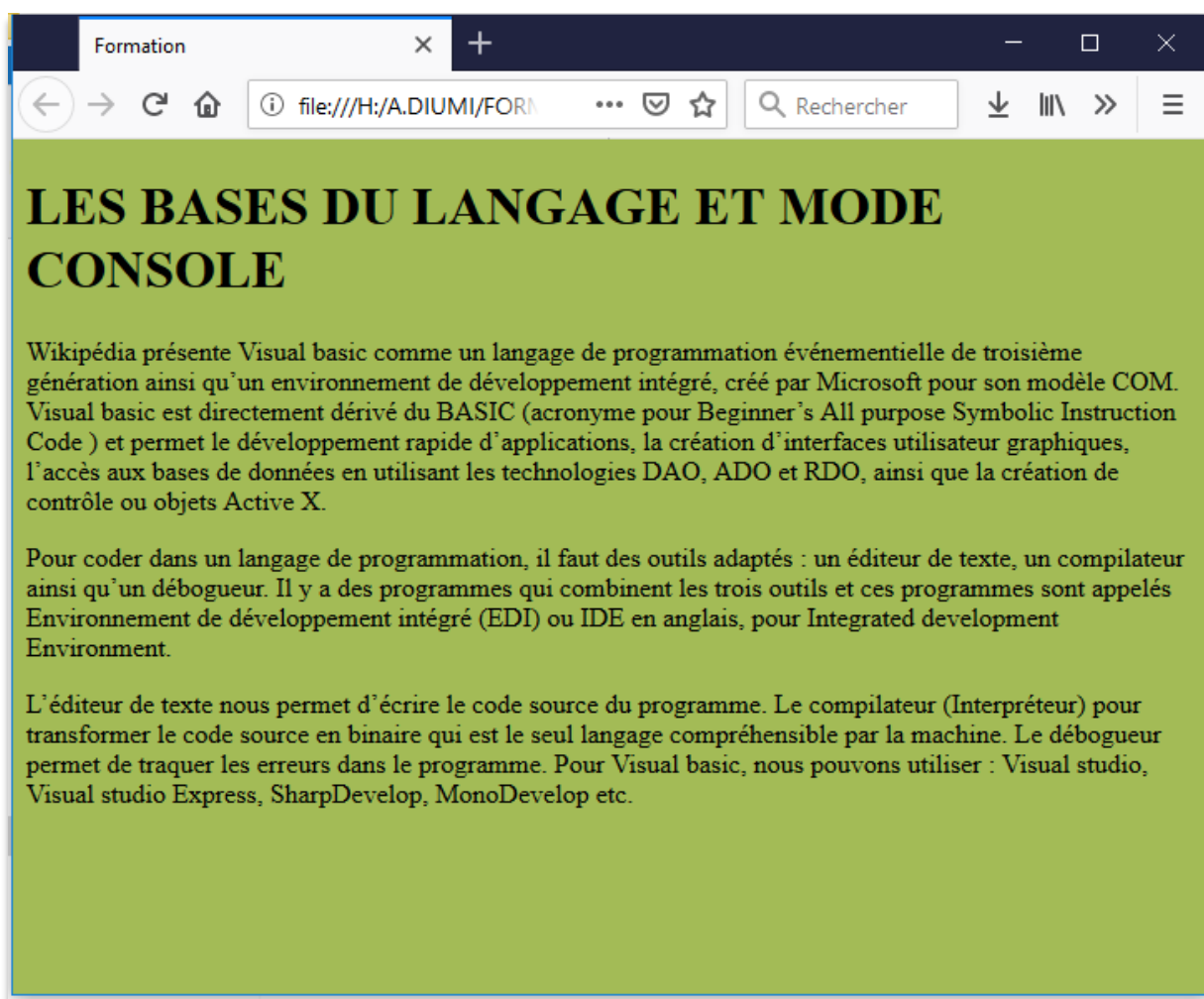
## II.5.2 La couleur de fond

On utilise la propriété **Background-color** qui s'utilise de la même façon que la propriété color.

Par exemple, pour ajouter une couleur de fond à la page, on écrit :

```
body
{
    background-color:#A3BB56;
}
```

Le résultat :



Remarquez que j'ai utilisé la balise body comme sélecteur car c'est elle qui englobe toute la page web.

## II.5.3 L'image de fond

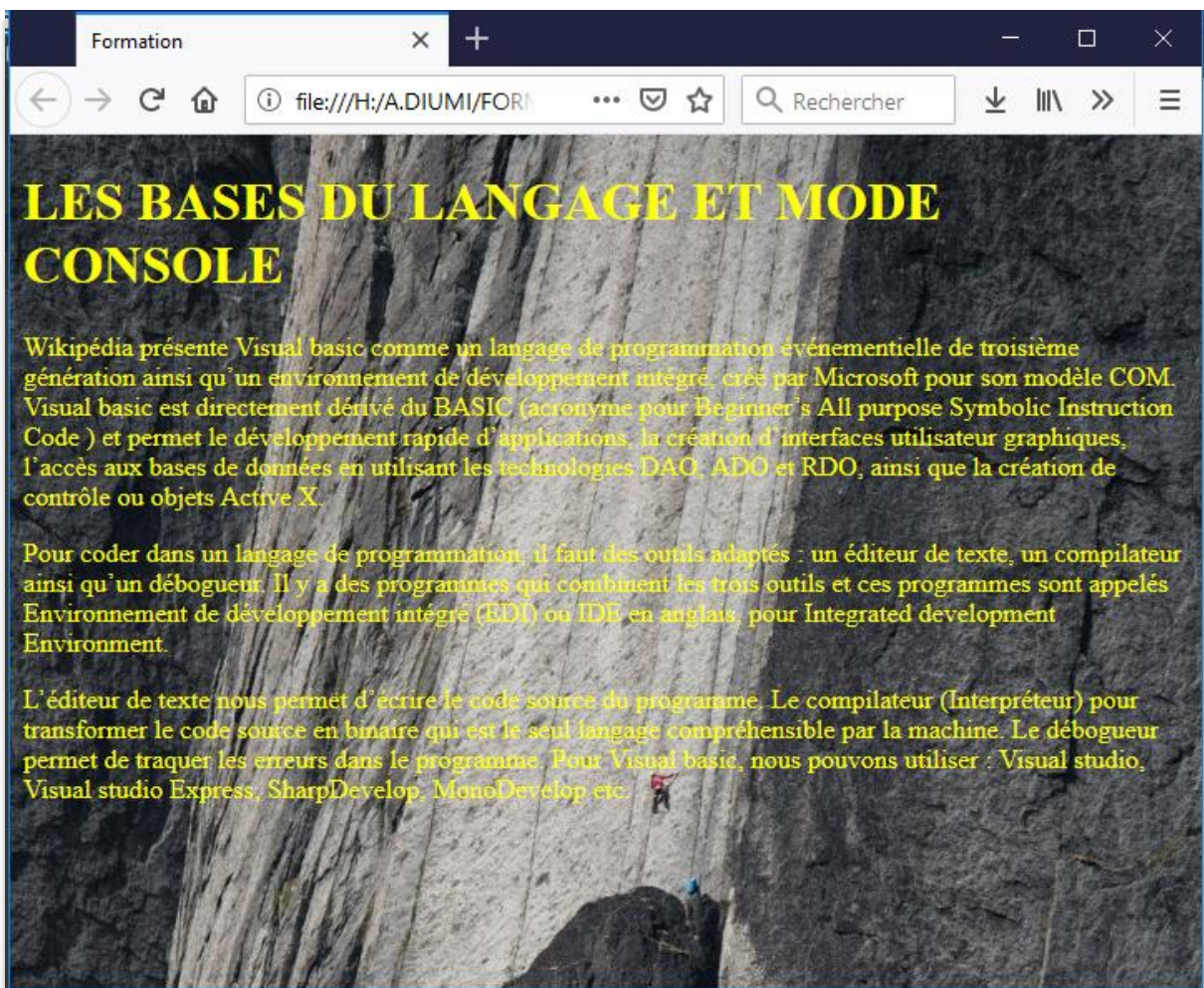
### a) Généralités

On utilise la propriété **Background-image**. Comme valeur à cette propriété, on met **url (" Nom\_de\_l'image")**. C'est mieux de mettre l'image dans le même dossier que votre fichier css.

Exemple :

```
body
{
    background-image:url("fond.png");
}
```

Rendu :



Il existe plusieurs autres propriétés qui viennent compléter ou changer le comportement de l'image de fond.

Nous allons utiliser le fichier html suivant :

## Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>

    <h1>AVANT PROPOS </h1>
    <h2>WISSEN CORPORATION </h2>

    <p> C'est un groupe formé essentiellement de meilleurs étudiants de nos groupes d'étude et d'encadrement qui se sont fixés plusieurs objectifs dont le premier est celui du partage de la connaissance en Informatique et en mathématique. <br/>
    Ainsi, pour atteindre notre premier objectif, nous organisons des formations essentiellement gratuites pour partager le peu que nous connaissons avec les autres. <br/> Car la connaissance, c'est comme un chandelier, sa lumière ne diminue pas s'il allume un autre chandelier.
    Albert Einstein a dit : « Il ne faut pas faire de l'intelligence un dieu, car si celle-ci est douée de muscles puissants, elle n'a pas de personnalité »
    ...Parfois, il n'existe pas de gens plus intelligents que les autres, mais c'est le travail qui nous différencie. Pour bâtir haut, il faut creuser profond et tu seras fort dans ce que tu te donnes. </p>

    <p>Il y a une naissance en toute connaissance, Thucydide a dit : « Avoir des connaissances sans les partager, c'est se mettre au niveau de celui qui n'a pas d'idées ». Wissen Corporation est là pour assurer votre formation, votre encadrement suivant vos désirs. </p>
    <h1> LES BASES DU LANGAGE ET MODE CONSOLE </h1>

    <p>Wikipédia présente Visual basic comme un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, créé par Microsoft pour son modèle COM. Visual basic est directement dérivé du BASIC (acronyme pour Beginner's All purpose Symbolic Instruction Code ) et permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôles ou objets Active X. </p>

    <p>Pour coder dans un langage de programmation, il faut des outils adaptés : un éditeur de texte, un compilateur ainsi qu'un débogueur. <br/>
```



Il y a des programmes qui combinent les trois outils et ces programmes sont appelés Environnement de développement intégré (EDI) ou IDE en anglais, pour Integrated development Environment. </p>

<p>L'éditeur de texte nous permet d'écrire le code source du programme. Le compilateur (Interpréteur) pour transformer le code source en binaire qui est le seul langage compréhensible par la machine. <br/> Le débogueur permet de traquer les erreurs dans le programme. Pour Visual basic, nous pouvons utiliser : Visual studio, Visual studio Express, SharpDevelop, MonoDevelop etc. </p>

## <h2> I. LES VARIABLES </h2>

<p> Visual Basic, comme la plupart des langages de programmation, utilise des variables pour stocker des valeurs. Elles servent à conserver momentanément des données en mémoire. <br/>

Pratiquement, nous pouvons supposer que déclarer des variables : c'est comme réserver des cases mémoires dans l'ordinateur et dans ces cases mémoires seront stockés momentanément les valeurs saisies par les utilisateurs ou les résultats de calculs de nos programmes. </p>

<p>Une variable a un nom (le mot que vous utilisez pour faire référence à la valeur que la variable contient).<br/> Elle a également un type de données (lequel détermine le genre des données que la variable peut stocker) et peut contenir à un instant donné une valeur.</p>

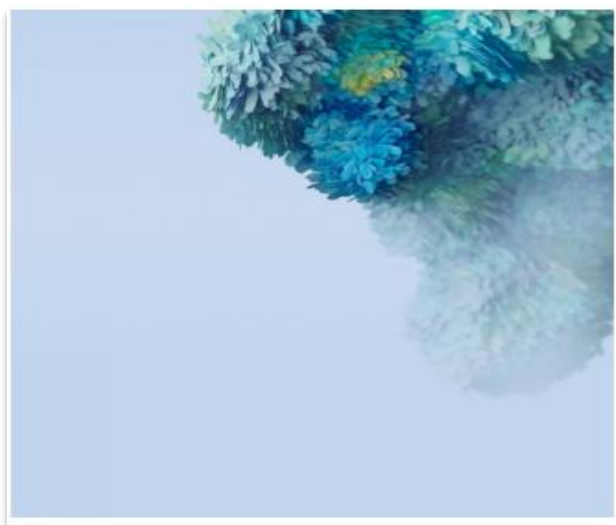
</body>

</html>

### *b) La répétition du fond*

Essayons d'appliquer une image de fond assez petite par rapport à notre page :

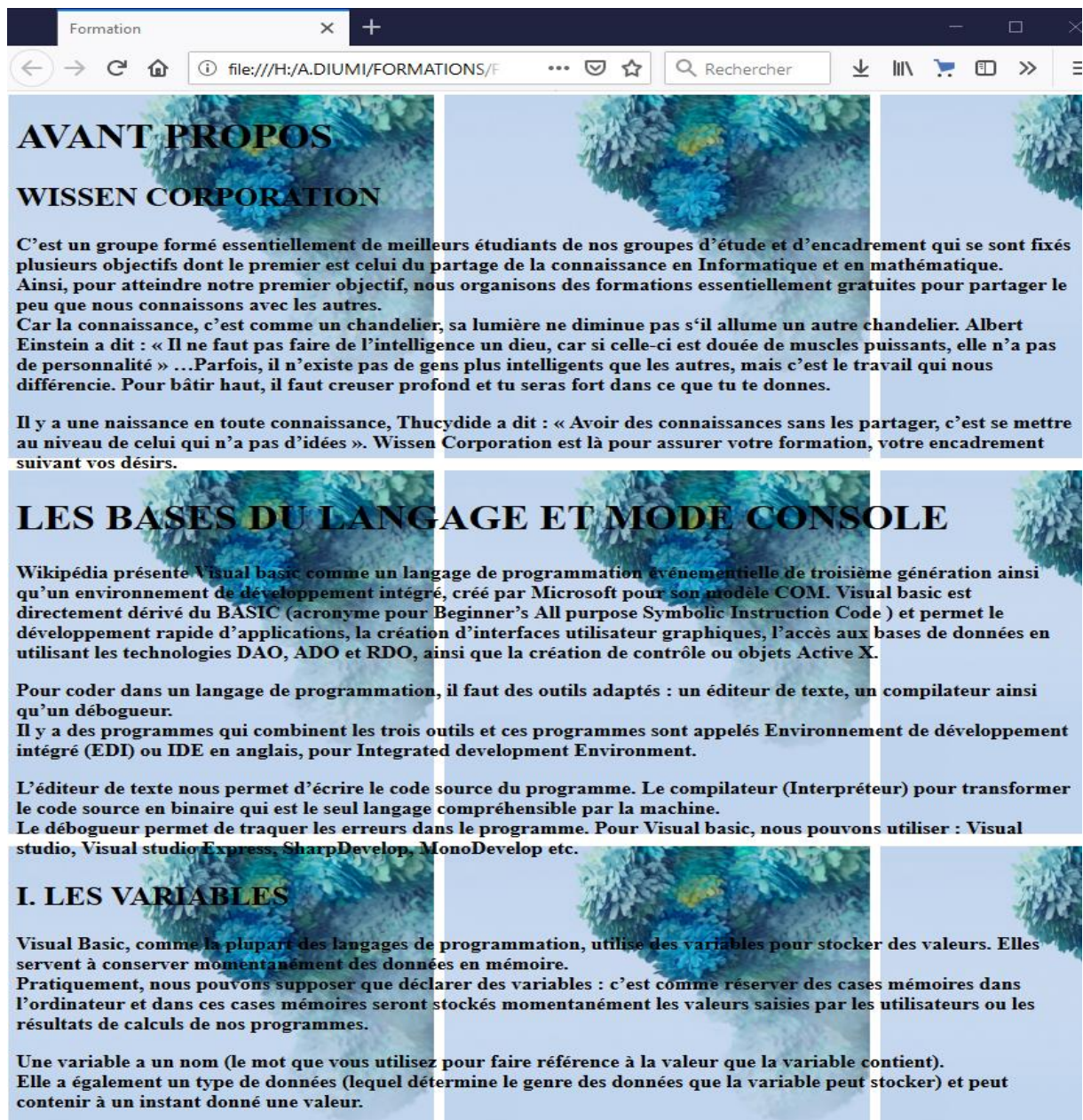
Voici l'image que je vais utiliser comme image de fond :



Forme.css

```
body
{
    background-image: url("fond.png");
}
```

Ça produit le résultat suivant :



Vous avez sans doute remarqué : l'image de fond se répète en mosaïque.

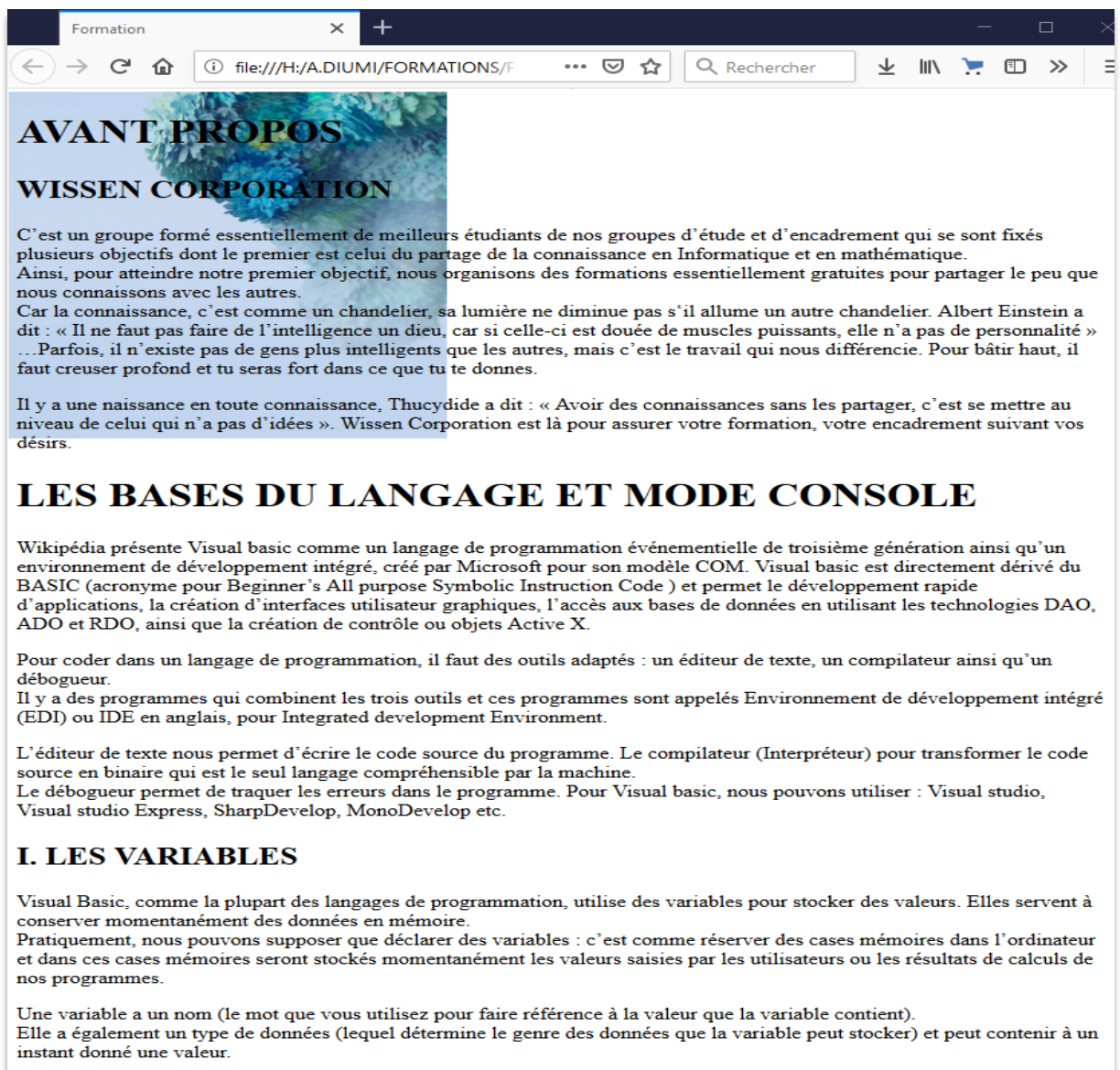
C'est la propriété **background-repeat** qui gère la répétition de l'image de fond au cas où celle-ci ne parvient pas à couvrir toute la page. Elle peut prendre l'une des valeurs suivantes :

- **no-repeat** : le fond ne sera pas répété. L'image sera donc unique sur la page.

Forme.css

```
body
{
    background-image: url("fond.png");
    background-repeat: no-repeat;
}
```

Résultat : l'image de fond ne se répète pas



- **repeat-x** : le fond sera répété uniquement sur la première ligne, horizontalement.

Forme.css

```
body
{
    background-image:url("fond.png");
background-repeat: x-repeat;
}
```

Résultat :

Formation

file:///H:/A.DIUMI/FORMATIONS/F

## AVANT PROPOS

### WISSEN CORPORATION

C'est un groupe formé essentiellement de meilleurs étudiants de nos groupes d'étude et d'encadrement qui se sont fixés plusieurs objectifs dont le premier est celui du partage de la connaissance en Informatique et en mathématique. Ainsi, pour atteindre notre premier objectif, nous organisons des formations essentiellement gratuites pour partager le peu que nous connaissons avec les autres.

Car la connaissance, c'est comme un chandelier, sa lumière ne diminue pas s'il allume un autre chandelier. Albert Einstein a dit : « Il ne faut pas faire de l'intelligence un dieu, car si celle-ci est douée de muscles puissants, elle n'a pas de personnalité » ... Parfois, il n'existe pas de gens plus intelligents que les autres, mais c'est le travail qui nous différencie. Pour bâtir haut, il faut creuser profond et tu seras fort dans ce que tu te donnes.

Il y a une naissance en toute connaissance, Thucydide a dit : « Avoir des connaissances sans les partager, c'est se mettre au niveau de celui qui n'a pas d'idées ». Wissen Corporation est là pour assurer votre formation, votre encadrement suivant vos désirs.

## LES BASES DU LANGAGE ET MODE CONSOLE

Wikipédia présente Visual basic comme un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, créé par Microsoft pour son modèle COM. Visual basic est directement dérivé du BASIC (acronyme pour Beginner's All purpose Symbolic Instruction Code ) et permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôle ou objets Active X.

Pour coder dans un langage de programmation, il faut des outils adaptés : un éditeur de texte, un compilateur ainsi qu'un débogueur.

Il y a des programmes qui combinent les trois outils et ces programmes sont appelés Environnement de développement intégré (EDI) ou IDE en anglais, pour Integrated development Environment.

L'éditeur de texte nous permet d'écrire le code source du programme. Le compilateur (Interpréteur) pour transformer le code source en binaire qui est le seul langage compréhensible par la machine.

Le débogueur permet de traquer les erreurs dans le programme. Pour Visual basic, nous pouvons utiliser : Visual studio, Visual studio Express, SharpDevelop, MonoDevelop etc.

### I. LES VARIABLES

Visual Basic, comme la plupart des langages de programmation, utilise des variables pour stocker des valeurs. Elles servent à conserver momentanément des données en mémoire.

Pratiquement, nous pouvons supposer que déclarer des variables : c'est comme réserver des cases mémoires dans l'ordinateur et dans ces cases mémoires seront stockés momentanément les valeurs saisies par les utilisateurs ou les résultats de calculs de nos programmes.

Une variable a un nom (le mot que vous utilisez pour faire référence à la valeur que la variable contient). Elle a également un type de données (lequel détermine le genre des données que la variable peut stocker) et peut contenir à un instant donné une valeur.

- **repeat-y** : le fond sera répété uniquement sur la première colonne, verticalement.

Forme.css

```
body
{
    background-image:url("fond.png");
background-repeat: repeat-y;
}
```

Résultat :

**AVANT PROPOS**

**WISSEN CORPORATION**

C'est un groupe formé essentiellement de meilleurs étudiants de nos groupes d'étude et d'encadrement qui se sont fixés plusieurs objectifs dont le premier est celui du partage de la connaissance en Informatique et en mathématique. Ainsi, pour atteindre notre premier objectif, nous organisons des formations essentiellement gratuites pour partager le peu que nous connaissons avec les autres.

Car la connaissance, c'est comme un chandelier, sa lumière ne diminue pas s'il allume un autre chandelier. Albert Einstein a dit : « Il ne faut pas faire de l'intelligence un dieu, car si celle-ci est douée de muscles puissants, elle n'a pas de personnalité » ...Parfois, il n'existe pas de gens plus intelligents que les autres, mais c'est le travail qui nous différencie. Pour bâtir haut, il faut creuser profond et tu seras fort dans ce que tu te donnes.

Il y a une naissance en toute connaissance, Thucydide a dit : « Avoir des connaissances sans les partager, c'est se mettre au niveau de celui qui n'a pas d'idées ». Wissen Corporation est là pour assurer votre formation, votre encadrement suivant vos désirs.

**LES BASES DU LANGAGE ET MODE CONSOLE**

Wikipédia présente Visual basic comme un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, créé par Microsoft pour son modèle COM. Visual basic est directement dérivé du BASIC (acronyme pour Beginner's All purpose Symbolic Instruction Code ) et permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôle ou objets Active X.

Pour coder dans un langage de programmation, il faut des outils adaptés : un éditeur de texte, un compilateur ainsi qu'un débogueur.

Il y a des programmes qui combinent les trois outils et ces programmes sont appelés Environnement de développement intégré (EDI) ou IDE en anglais, pour Integrated development Environment.

L'éditeur de texte nous permet d'écrire le code source du programme. Le compilateur (Interpréteur) pour transformer le code source en binaire qui est le seul langage compréhensible par la machine.

Le débogueur permet de traquer les erreurs dans le programme. Pour Visual basic, nous pouvons utiliser : Visual studio, Visual studio Express, SharpDevelop, MonoDevelop etc.

**I. LES VARIABLES**

Visual Basic, comme la plupart des langages de programmation, utilise des variables pour stocker des valeurs. Elles servent à conserver momentanément des données en mémoire.

Pratiquement, nous pouvons supposer que déclarer des variables : c'est comme réserver des cases mémoires dans l'ordinateur et dans ces cases mémoires seront stockés momentanément les valeurs saisies par les utilisateurs ou les résultats de calculs de nos programmes.

Une variable a un nom (le mot que vous utilisez pour faire référence à la valeur que la variable contient). Elle a également un type de données (lequel détermine le genre des données que la variable peut stocker) et peut contenir à un instant donné une valeur.

- **repeat** : le fond sera répété en mosaïque (par défaut).

### c) Fixation de l'image de fond

La propriété **background-attachment** permet de fixer l'image de fond. Deux valeurs sont disponibles :

**fixed** : l'image de fond reste fixe ;

**scroll** : l'image de fond défile avec le texte (par défaut).

### d) Positionnement de l'image de fond

La propriété **background-position** permet de choisir l'emplacement de l'image de fond. Cette propriété aura de l'effet si l'image est unique dans la page, c'est-à-dire qu'elle ne se répète pas. C'est pourquoi il faut la combiner avec **background-repeat : no-repeat**.

Vous devez donner à **background-position** deux valeurs en pixels pour indiquer la position du fond par rapport au coin supérieur gauche de l'élément auquel vous appliquez le fond.

Par exemple si j'écris ceci :

Forme.csss

```
body
{
  background-image:url("fond.png");
  background-repeat:no-repeat;
  background-position: 100px 150px;
}
```

Résultat :

about:sessionstore Formation

file:///H:/A.DIUMI/FORMATIONS/F

Rechercher

## AVANT PROPOS

150px

### WISSEN CORPORATION

C'est un groupe formé essentiellement de meilleurs étudiants de nos groupes d'étude et d'encadrement qui se sont fixés plusieurs objectifs dont le premier est celui du partage de la connaissance en Informatique et en mathématique. Afin d'atteindre notre premier objectif, nous organisons des formations essentiellement gratuites pour partager le peu que nous connaissons avec les autres.

100px

Car la connaissance, c'est comme un chandelier, sa lumière ne diminue pas s'il allume un autre chandelier. Albert Einstein a dit : « Il ne faut pas faire de l'intelligence un dieu, car si celle-ci est douée de muscles puissants, elle n'a pas de personnalité » ...Parfois, il n'existe pas de gens plus intelligents que les autres, mais c'est le travail qui nous différencie. Pour bâtir haut, il faut creuser profond et tu seras fort dans ce que tu te donnes.

Il y a une naissance en toute connaissance, Thucydide a dit : « Avoir des connaissances sans les partager, c'est se mettre au niveau de celui qui n'a pas d'idées ». Wissen Corporation est là pour assurer votre formation, votre encadrement suivant vos désirs.

## LES BASES DU LANGAGE ET MODE CONSOLE

Wikipédia présente Visual basic comme un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, créé par Microsoft pour son modèle COM. Visual basic est directement dérivé du BASIC (acronyme pour Beginner's All purpose Symbolic Instruction Code) et permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôle ou objets Active X.

Pour coder dans un langage de programmation, il faut des outils adaptés : un éditeur de texte, un compilateur ainsi qu'un débogueur.

Il y a des programmes qui combinent les trois outils et ces programmes sont appelés Environnement de développement intégré (EDI) ou IDE en anglais, pour Integrated development Environment.

L'éditeur de texte nous permet d'écrire le code source du programme. Le compilateur (Interpréteur) pour transformer le code source en binaire qui est le seul langage compréhensible par la machine.

Le débogueur permet de traquer les erreurs dans le programme. Pour Visual basic, nous pouvons utiliser : Visual studio, Visual studio Express, SharpDevelop, MonoDevelop etc.

### I. LES VARIABLES

Visual Basic, comme la plupart des langages de programmation, utilise des variables pour stocker des valeurs. Elles servent à conserver momentanément des données en mémoire.

Pratiquement, nous pouvons supposer que déclarer des variables : c'est comme réserver des cases mémoires dans l'ordinateur et dans ces cases mémoires seront stockés momentanément les valeurs saisies par les utilisateurs ou les résultats de calculs de nos programmes.

Une variable a un nom (le mot que vous utilisez pour faire référence à la valeur que la variable contient). Elle a également un type de données (lequel détermine le genre des données que la variable peut stocker) et peut contenir à un instant donné une valeur.

Il est aussi possible d'utiliser ces valeurs en anglais :

- **top** : en haut ;
- **bottom** : en bas ;
- **left** : à gauche ;
- **center** : centré ;
- **right** : à droite.

Il est possible de combiner ces mots. Par exemple, si je veux que le fond soit placé en bas à droite, j'écrirai :

Forme.css

```
body
{
    background-image:url("fond.png");
    background-repeat:no-repeat;
    background-position: bottom right;
}
```

Résultat :





### e) Combiner les propriétés relatives au fond

Il est possible de combiner toutes les propriétés relatives au fond (background-image, background-repeat, background-attachment, background-position) avec la super propriété **background**. Par exemple, pour le code précédent, je pouvais aussi écrire :

```
body
{
    background: url("fond.png") no-repeat bottom right;
}
```

Remarque :

- L'ordre des valeurs n'a pas d'importance. Vous pouvez combiner les valeurs dans n'importe quel ordre. On peut commencer par l'image ou la répétition, ça ne change rien.
- Vous n'êtes pas obligés de mettre toutes les valeurs.

### f) Plusieurs images de fond


Il est également possible de mettre plusieurs images de fond, pour cela, il suffit de séparer les déclarations de ces images par des virgules.

Par exemple, si je veux ajouter deux images de fond à ma page :

Forme.css

```
body
{
    background: url("fond.png") no-repeat bottom left,
                url("fond3.jpg") no-repeat top right;
}
```

Résultat :



The image is a screenshot of a web browser window. The address bar shows the file path: file:///H:/A.DIUMI/FORMATIONS/F. The browser has two tabs: 'about:sessionrestore' and 'Formation'. The page content is as follows:

## AVANT PROPOS

### WISSEN CORPORATION

C'est un groupe formé essentiellement de meilleurs étudiants de nos groupes d'étude et d'encadrement qui se sont fixés plusieurs objectifs dont le premier est celui de partage de la connaissance en Informatique et en mathématique. Ainsi, pour atteindre notre premier objectif, nous organisons des formations essentiellement gratuites pour partager le peu que nous connaissons avec les autres.

Car la connaissance, c'est comme un chandelier, sa lumière ne diminue pas s'il allume un autre chandelier. Albert Einstein a dit : « Il ne faut pas faire de l'intelligence un dieu, car si celle-ci est douée de muscles puissants, elle n'a pas de personnalité » ...Parfois, il n'existe pas de gens plus intelligents que les autres, mais c'est le travail qui nous différencie. Pour bâtir haut, il faut creuser profond et tu seras fort dans ce que tu te donnes.

Il y a une naissance en toute connaissance, Thucydide a dit : « Avoir des connaissances sans les partager, c'est se mettre au niveau de celui qui n'a pas d'idées ». Wissen Corporation est là pour assurer votre formation, votre encadrement suivant vos désirs.

## LES BASES DU LANGAGE ET MODE CONSOLE

Wikipédia présente Visual basic comme un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, créé par Microsoft pour son modèle COM. Visual basic est directement dérivé du BASIC (acronyme pour Beginner's All purpose Symbolic Instruction Code ) et permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôle ou objets Active X.

Pour coder dans un langage de programmation, il faut des outils adaptés : un éditeur de texte, un compilateur ainsi qu'un débogueur.

Il y a des programmes qui combinent les trois outils et ces programmes sont appelés Environnement de développement intégré (EDI) ou IDE en anglais, pour Integrated development Environment.

L'éditeur de texte nous permet d'écrire le code source du programme. Le compilateur (Interpréteur) pour transformer le code source en binaire qui est le seul langage compréhensible par la machine.

Le débogueur permet de traquer les erreurs dans le programme. Pour Visual basic, nous pouvons utiliser : Visual studio, Visual studio Express, SharpDevelop, MonoDevelop etc.

### I. LES VARIABLES

Visual Basic, comme la plupart des langages de programmation, utilise des variables pour stocker des valeurs. Elles servent à conserver momentanément des données en mémoire.

Pratiquement, nous pouvons supposer que déclarer des variables : c'est comme réserver des cases mémoires dans l'ordinateur et dans ces cases mémoires seront stockés momentanément les valeurs saisies par les utilisateurs ou les résultats de calculs de nos programmes.

Une variable a un nom (le mot que vous utilisez pour faire référence à la valeur que la variable contient). Elle a également un type de données (lequel détermine le genre des données que la variable peut stocker) et peut contenir à un instant donné une valeur.

## II.5.4 L'opacité d'un élément

La propriété **opacity** permet de déterminer le degré d'opacité (inverse de la transparence) d'un élément. Elle prend des valeurs de 0 (totalement transparent) à 1 (totalement opaque, c'est la valeur par défaut)

Exemple :

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel="stylesheet" href="forme.css" />
  </head>
  <body>

<h1> LES BASES DU LANGAGE ET MODE CONSOLE </h1>
<p class="p1" >Wikipédia présente Visual basic comme un langage de
programmation événementielle de troisième génération ainsi qu'un
environnement de développement intégré, créé par Microsoft pour son
modèle COM. <br/>Visual basic est directement dérivé du BASIC (acronyme pour B
eginner's All purpose Symbolic Instruction Code ) et permet le
développement rapide d'applications, la création d'interfaces utilisateur
graphiques, l'accès aux bases de données en utilisant les technologies DAO,
ADO et RDO, ainsi que la création de contrôle ou objets Active X. </p>

<p class="p2" >Pour coder dans un langage de programmation, il faut des outils
adaptés : un éditeur de texte, un compilateur ainsi qu'un débogueur.<br/> Il y
a
des programmes qui combinent les trois outils et ces programmes sont
appelés Environnement de développement intégré (EDI) ou IDE en anglais,
pour Integrated development Environment. </p>

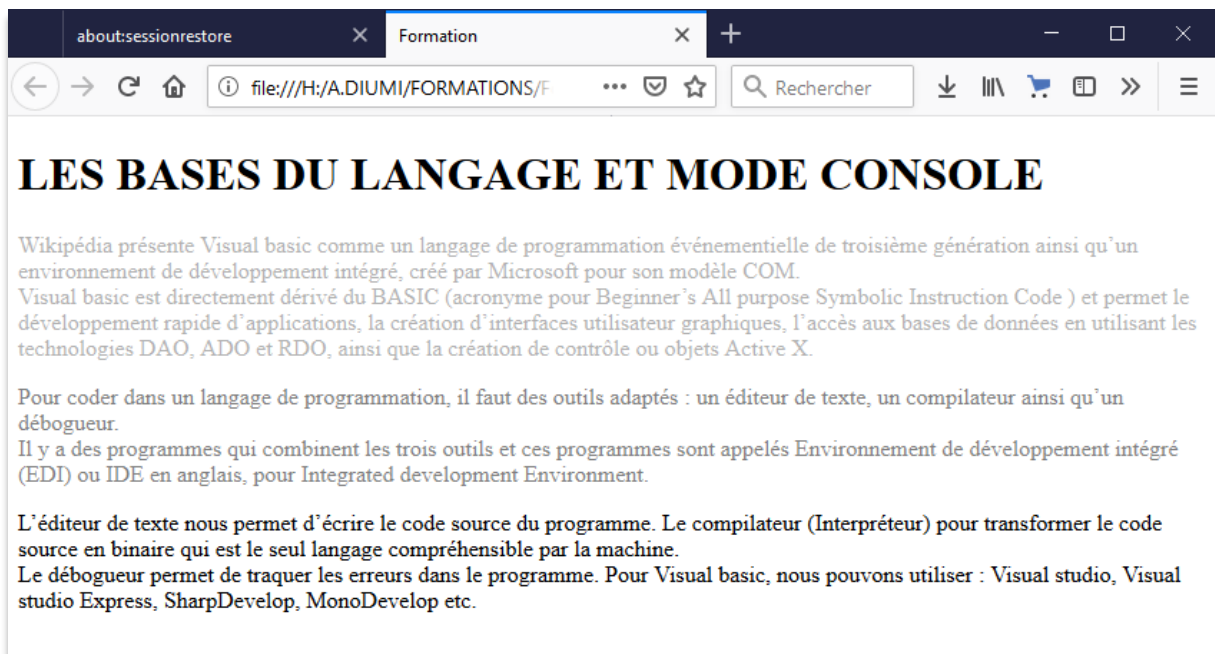
<p class="p3" >L'éditeur de texte nous permet d'écrire le code source du progr
amme.
Le compilateur (Interpréteur) pour transformer le code source en binaire qui
est le seul langage compréhensible par la machine.<br/>
Le débogueur permet de traquer les erreurs dans le programme.
Pour Visual basic, nous pouvons utiliser : Visual studio, Visual studio
Express, SharpDevelop, MonoDevelop etc. </p>
</body>
</html>
```

Forme.css

```
.p1
{
  opacity:0.3;
}

.p2
{
  opacity: 0.5;
}
```

Résultat :



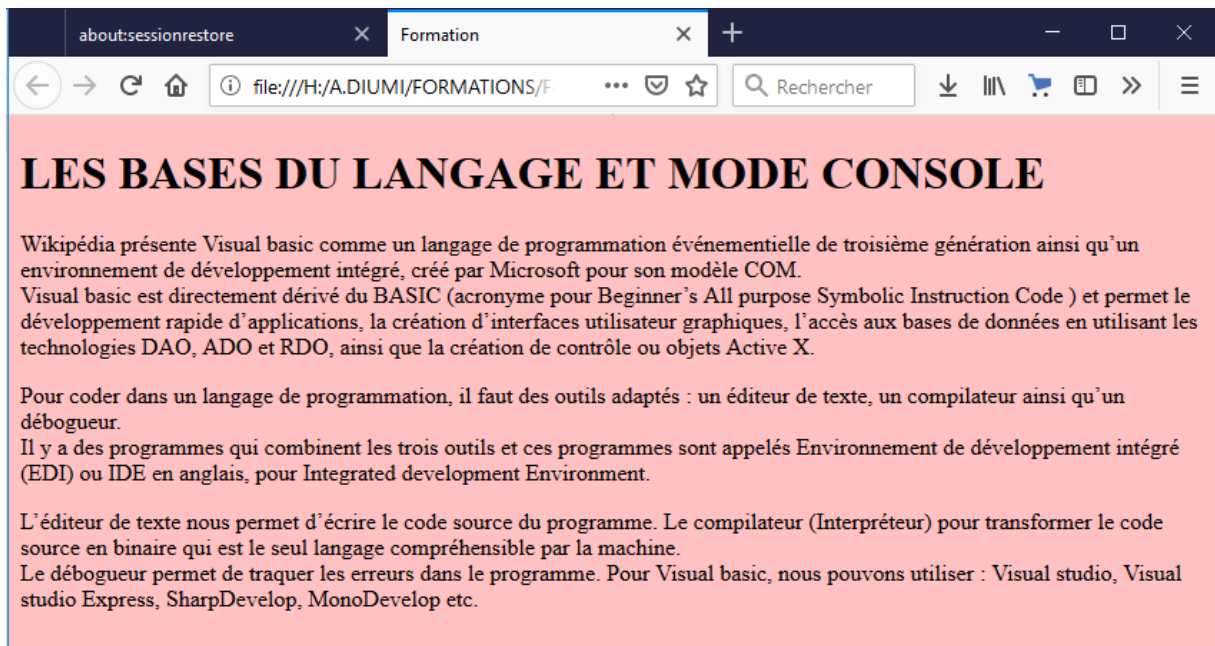
## II.5.5 La notation RGBA

Elle permet de définir la couleur (comme vu plus haut) mais avec un quatrième paramètre qui permet de définir le niveau d'opacité.

Exemple :

```
body
{
  background-color: rgba(255,50,50,0.3);
}
```

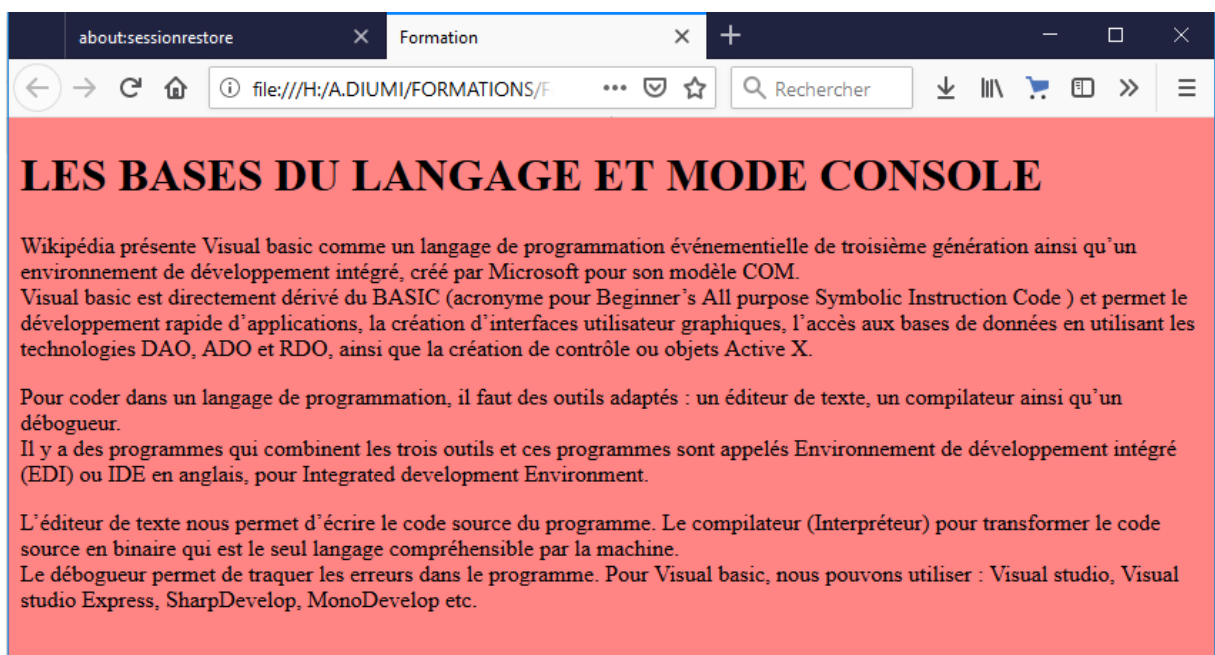
Résultat :



Essayons d'augmenter l'opacité :

```
body
{
    background-color:RGBA(255,50,50,0.6);
}
```

Résultat :



Remarque

Si vous appliquez la propriété **opacity** à un élément de la page, tout le contenu de cet élément sera rendu transparent (même les images, les autres blocs à l'intérieur, etc.). Si vous voulez juste rendre la couleur de fond transparente, utilisez plutôt la notation RGBA.

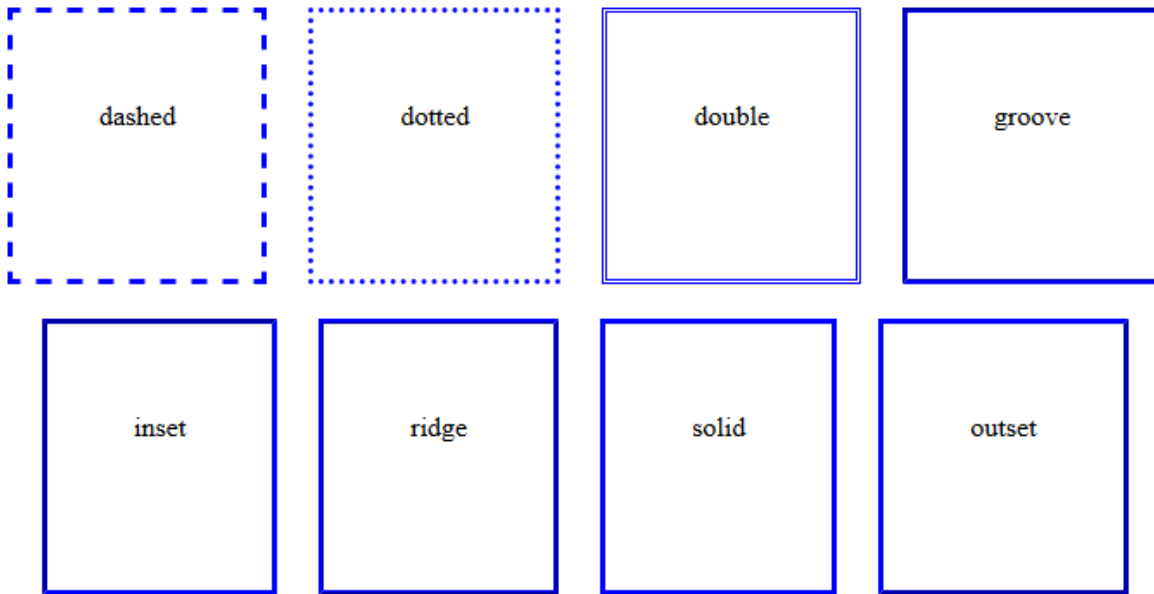
## II.6 LES BORDURES ET LES OMBRES

### II.6.1 Bordures standards

On utilise la propriété `border` qui englobe plusieurs propriétés relatives aux bordures (`border-style`, `border-color`, `border-width`). Elle peut prendre jusqu'à trois valeurs :

- **La largeur** : indiquez la largeur de votre bordure. Mettez une valeur en pixels (comme 3px).
- **La couleur** : c'est la couleur de votre bordure. On peut utiliser l'une des trois méthodes vues ci haut.
- **Le type de bordure** : là, vous avez le choix. Votre bordure peut être un simple trait, ou des pointillés, ou encore des tirets, etc. Voici les différentes valeurs disponibles :
  - `none` : pas de bordure (par défaut) ;
  - `solid` : un trait simple ;
  - `dotted` : pointillés ;
  - `dashed` : tirets ;
  - `double` : bordure double ;
  - `groove` : en relief ;
  - `ridge` : autre effet relief ;
  - `inset` : effet 3D global enfoncé ;
  - `outset` : effet 3D global surélevé.

Voici l'aperçu de différents types de bordures :



Par exemple, si je veux que le deuxième paragraphe soit encadré, j'écrirai :

Forme.css

```
.para2
{
  border: 2px blue solid;
}
```

Aperçu :



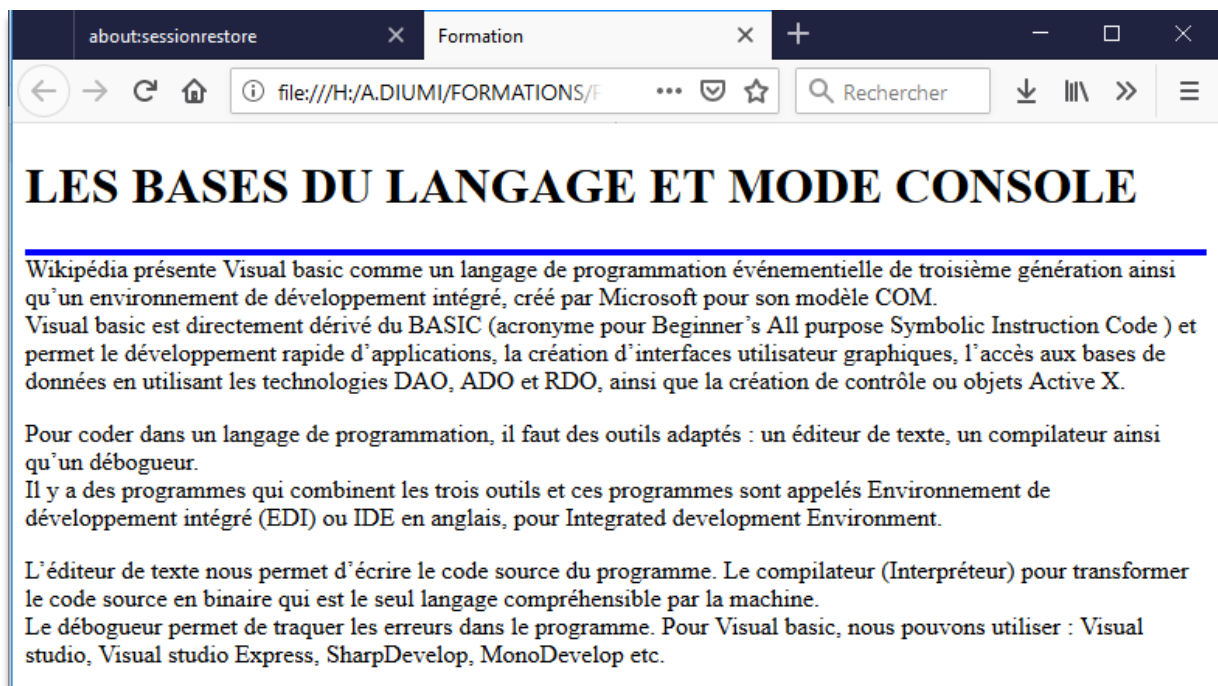
Vous n'êtes pas obligés d'utiliser le même type de bordure aux quatre côtés de l'élément (gauche, bas, haut et droite). Pour personnaliser les quatre côtés, on a quatre propriétés :

- ***border-top*** : bordure du haut ;
- ***border-bottom*** : bordure du bas ;
- ***border-left*** : bordure de gauche ;
- ***border-right*** : bordure de droite.

Si je veux ajouter une bordure seulement en haut du premier paragraphe, j'écrirai :

```
.para1
{
    border-top: 4px blue solid;
}
```

Aperçu :





## II.6.2 Les bordures arrondies

On utilise propriété **border-radius** pour arrondir facilement les angles de n'importe quel élément :

```
border-radius: x y;
```

Les deux valeurs (x et y) déterminent respectivement les rayons horizontaux et verticaux d'un quart d'une ellipse, ce qui induira la courbure de l'angle. Si les deux valeurs sont égales (x=y), on écrit seulement une valeur.

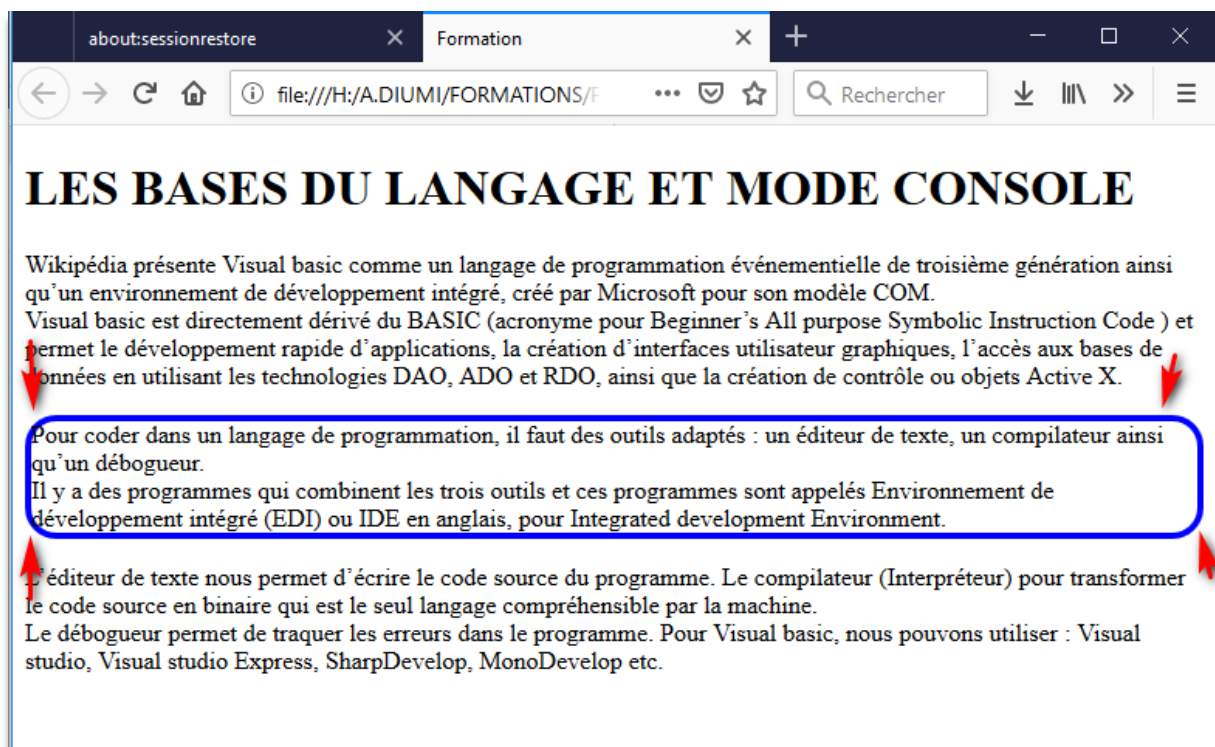
L'arrondi se voit notamment si l'élément a des bordures ou une couleur de fond.

Exemple

Forme.css

```
.para2
{
  border: 4px blue solid;
  border-radius: 20px;
}
```

Résultat :



On peut aussi préciser la forme de l'arrondi pour chaque coin. Dans ce cas, indiquez quatre valeurs :

**Border-radius** : *valeur1 valeur2 valeur3 valeur4* ;

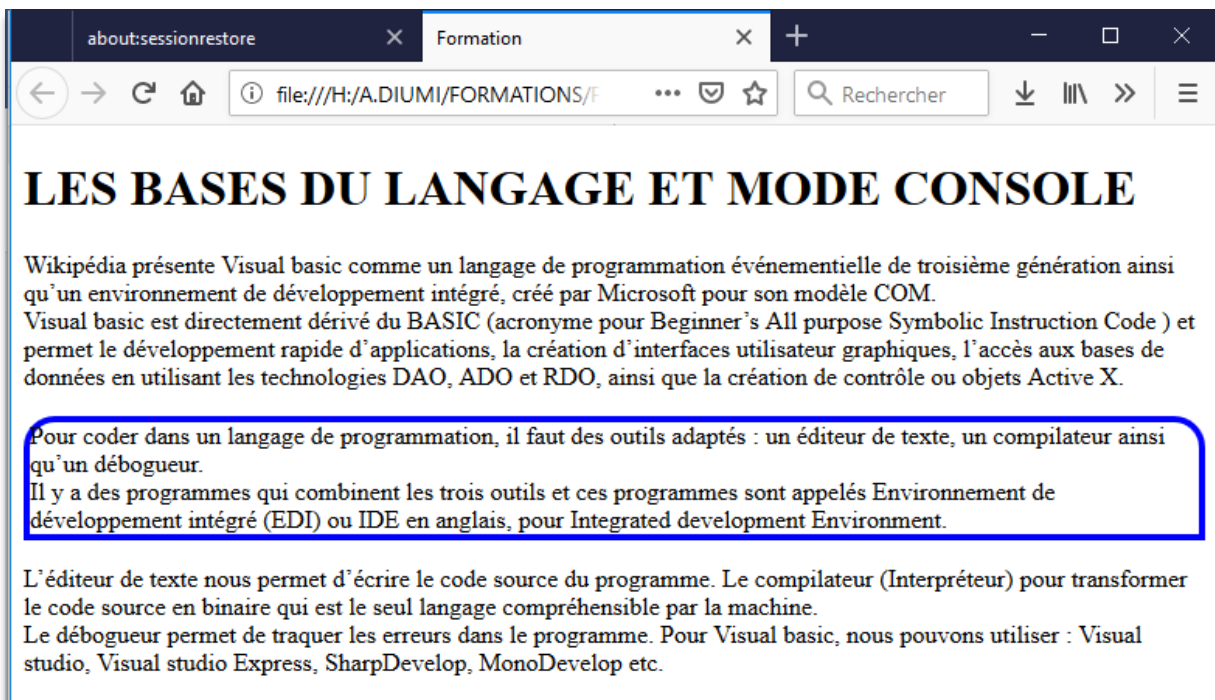
Les valeurs correspondent aux angles suivants dans cet ordre :

1. en haut à gauche ;
2. en haut à droite ;
3. en bas à droite ;
4. en bas à gauche.

Exemple

```
.para2
{
  border: 4px blue solid;
  border-radius: 20px 20px 0px 0px;
}
```

Résultat :



## II.6.3 Les bordures imagées

En CSS, il est possible d'utiliser une image comme bordure d'un élément de type bloc : on utilise la propriété `border-image`.

Elle prend en paramètre une image rectangulaire comme celle-ci par exemple :



Ensuite, divise l'image en neuf parties comme le montre la figure suivante :



Le centre de l'image (5) sera invisible pour permettre l'affichage du contenu, les autres parties seront utilisées comme bordure du contenu dans l'ordre suivant :

La partie 1 sera utilisée comme angle supérieur gauche ;

La partie 2 comme image de la bordure supérieure ;

La partie 3 comme image de l'angle supérieur droit ;

La partie 4 comme image de la bordure gauche ;

La partie 6 comme image de la bordure du côté droit ;

La partie 7 comme angle inférieur gauche ;

Partie 8 comme image de la bordure inférieure et

La partie 9 comme angle inférieur droit.

Cette propriété s'utilise comme suit :

`border-image:url(image) x1 x2 x3 x4 valeur1 valeur2;`

où :

- *image* représente le chemin de l'image à utiliser comme bordure ;
- *x1*, *x2*, *x3* et *x4* représentent respectivement la longueur du côté supérieur, droit, inférieur et gauche d'une partie de l'image.

On peut indiquer deux valeurs *x1 x2* au lieu de quatre valeurs, dans ce cas, ils représentent respectivement la longueur des côtés supérieur/inférieur et gauche/droite.

Si on indique seulement une valeur, elle va s'appliquer aux quatre côtés.

- Valeur peut prendre une des valeurs suivantes : repeat, round et stretch. Si on indique une seule valeur, elle va s'appliquer aux quatre coins de la bordure, mais si on indique deux valeurs, elles vont s'appliquer respectivement aux bordures supérieure/inférieure et gauche/droite.

Exemple

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel="stylesheet" href="forme.css" />
  </head>
  <body>

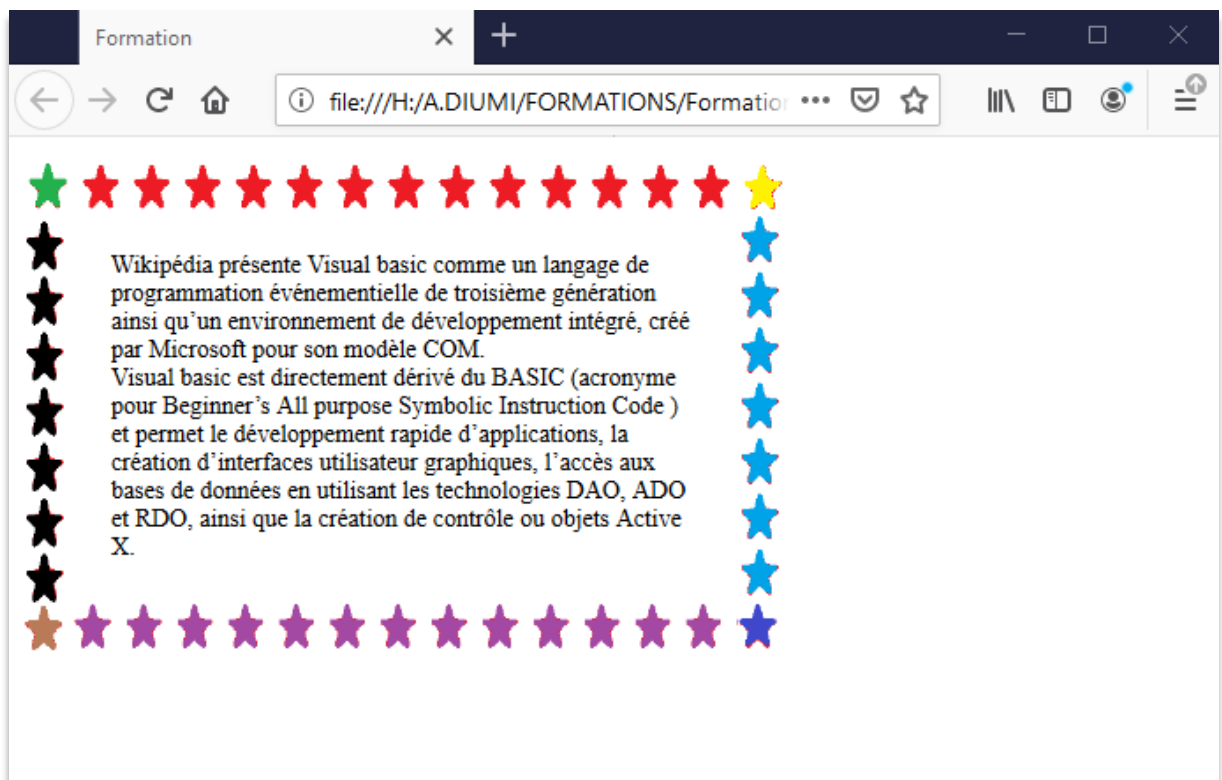
  <p>Wikipédia présente Visual basic comme un langage de
programmation événementielle de troisième génération ainsi qu'un
environnement de développement intégré, créé par Microsoft pour son
modèle COM. <br/>Visual basic est directement dérivé du BASIC (acronyme
pour Beginner's All purpose Symbolic Instruction Code ) et permet le
développement rapide d'applications, la création d'interfaces utilisateur
graphiques, l'accès aux bases de données en utilisant les technologies DAO,
ADO et RDO, ainsi que la création de contrôle ou objets Active X. </p>

  </body>
</html>
```

## Forme.css

```
p
{
border:red solid;
border-image:url(bordure_image.png) 50 round;
border-width:30px;
width:50%;
height: 180px;
padding:20px;
}
```

Résultat :



## 11.6.4 Les ombres

Il y a deux types d'ombres :

- Les ombres de boîtes et
- Les ombres de texte

### a) Les ombres de boîtes

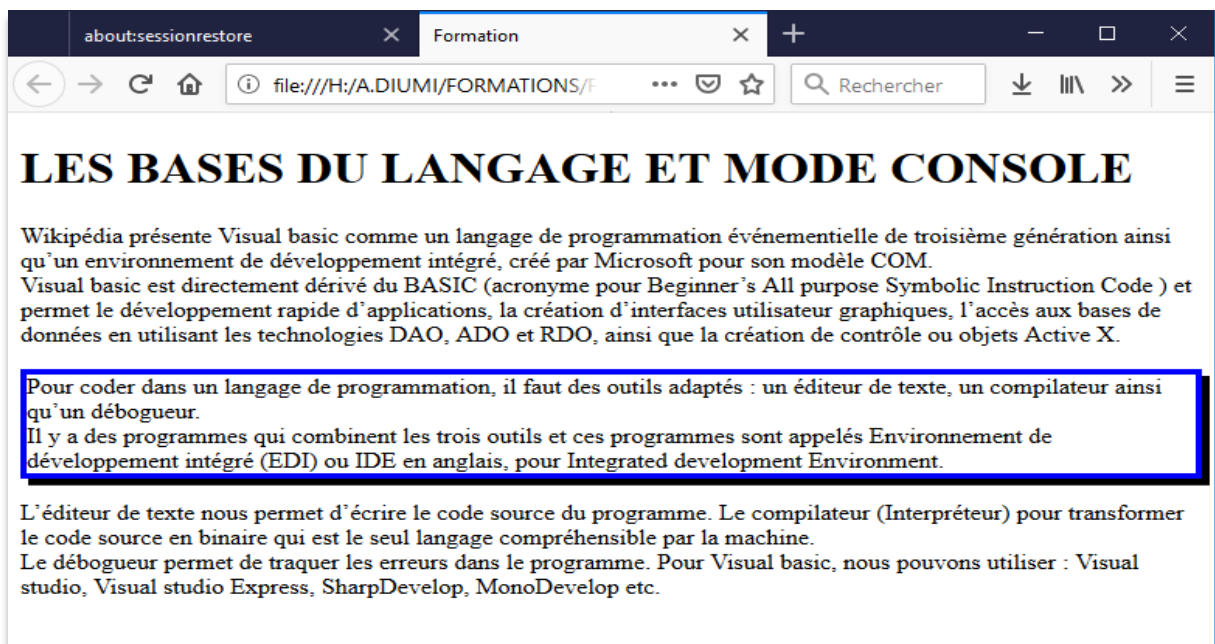
On utilise la propriété **box-shadow** s'applique à tout le bloc et prend quatre valeurs dans l'ordre suivant :

1. le décalage horizontal de l'ombre ;
2. le décalage vertical de l'ombre ;
3. l'adoucissement du dégradé ou rayon de propagation (appelé aussi flou gaussien) de l'ombre. En d'autres termes, elle désigne l'intensité du dégradé ou du flou. Elle est facultative et par défaut, elle vaut 0.
4. la couleur de l'ombre.

Exemple

```
.para2
{
    border: 4px blue solid;
    box-shadow: 5px 5px 0px black;
}
```

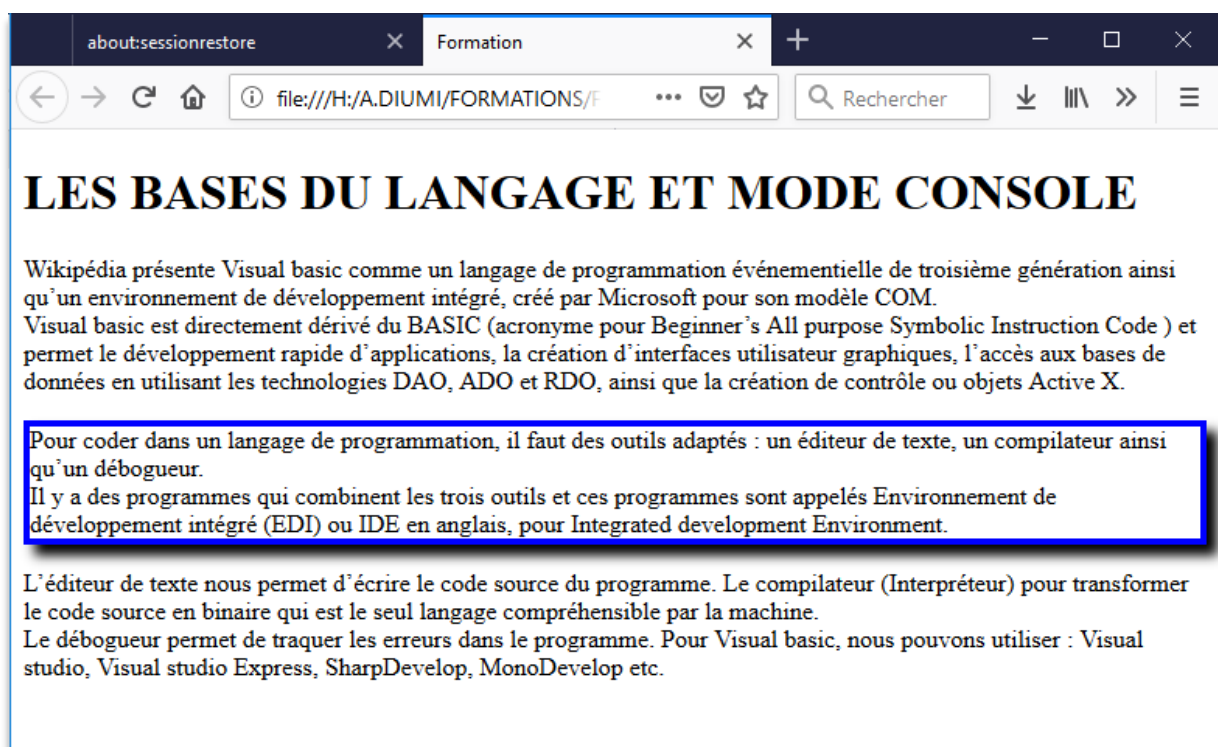
Aperçu :



Cette fois-ci, ajoutons l'adoucissement (troisième paramètre). Il peut être faible (inférieur au décalage), normal (égal au décalage) ou élevé (supérieur au décalage).

```
.para2
{
  border: 4px blue solid;
  box-shadow: 7px 7px 7px black;
}
```

Résultat :



### b) Ombre du texte

On peut ajouter une ombre sur les lettres du texte : on utilise la propriété **text-shadow**. Les valeurs fonctionnent exactement de la même façon que box-shadow : décalage, adoucissement et couleur.

Exemple :

```
.para2
{
  text-shadow: 3px 3px 3px black ;
}

.para1
```

```
{  
  
text-shadow: 1px 1px 0px blue ;  
  
}
```

Aperçu :



Avant d'utiliser ça, songez à la visibilité de votre texte, utilisez ça si besoin pour les titres.



## II.7 PSEUDOS CLASSES ET PSEUDO ELEMENTS

Les pseudos classes et pseudos éléments nous permettent d'affiner le style appliqué à un certain nombre d'éléments en définissant une réaction à un événement ou bien à la position relative de l'élément au sein des autres éléments.

Il existe plusieurs pseudos classes et pseudo éléments, nous allons essayer d'étudier quelques-uns.

### II.7.1 Pseudo-classes dynamiques (Aspect dynamique)

CSS nous permet aussi de modifier l'apparence des éléments de façon dynamique, c'est-à-dire que des éléments peuvent changer de forme une fois que la page a été chargée.

#### *a) Au survol*

Pour définir l'apparence d'un élément lors du survol de la souris, on écrit :

```
sélecteur: hover
{
  propriété1: valeur;
  propriété2: valeur2;
  propriété3: valeur;
}
```

Les propriétés indiquées s'appliqueront au sélecteur seulement si la souris le survole.

Exemple

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
<h1> A PROPOS DU DEVELOPPEUR </h1>
<p> Ce site a été développé par Aimé DIUMI DIKOLO .<br/>
Il est co-fondateur et coordonnateur de Wissen Corporation</p>
<p>  </p>
</body>
</html>
```

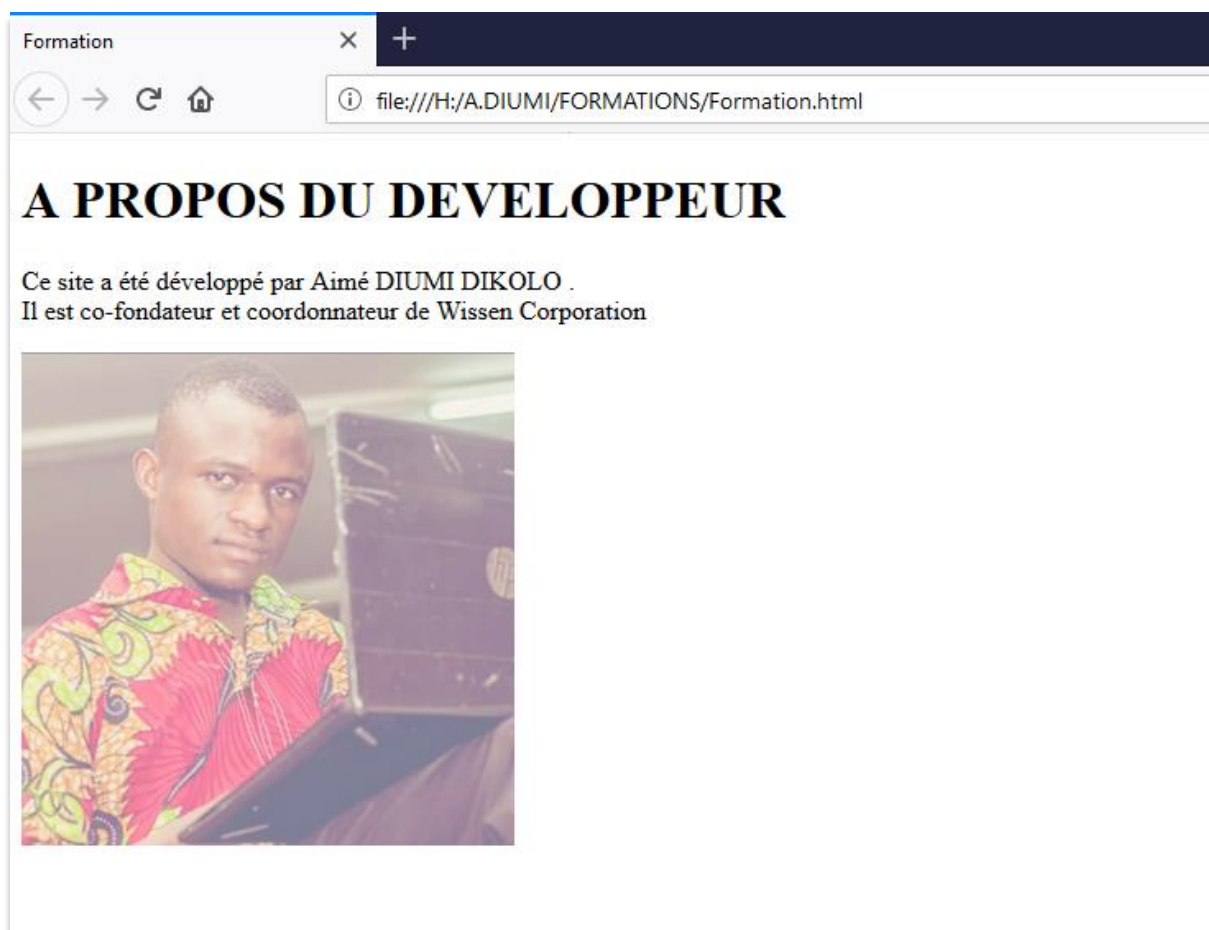
J'aimerais que la photo soit un peu transparente et devienne opaque au survol de la souris, j'écrirai :

Forme.css

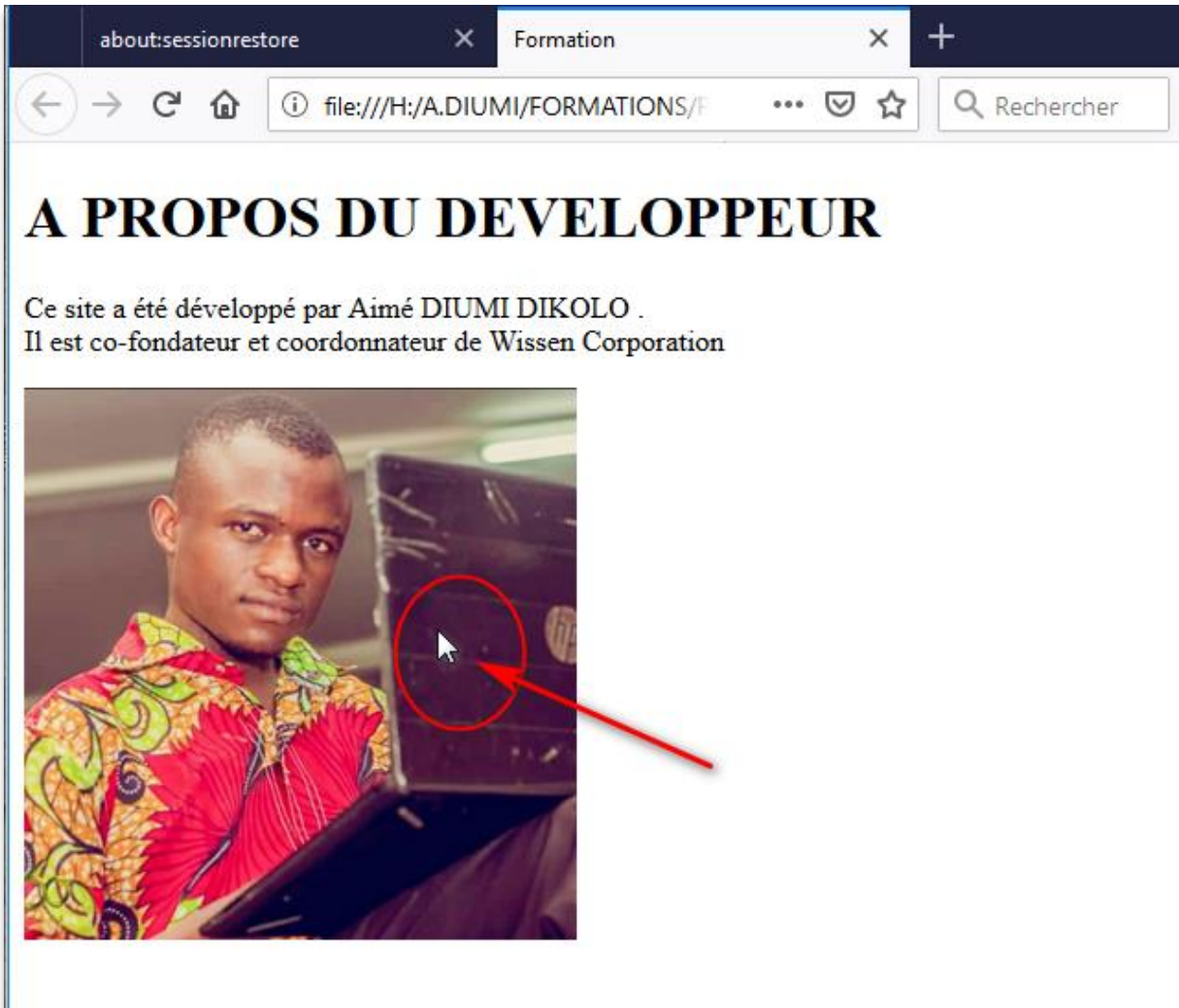
```
img
{
  opacity: 0.5;
}

img: hover
{
  opacity: 1;
}
```

Résultat avant survol de la souris sur l'image :



Résultat au survol de la souris sur l'image :



Cette technique (: hover) est souvent utilisée pour les liens, on utilise un style donné pour les liens par défaut (non survolés) et on change d'aspect lors du survol de la souris sur les liens (liens survolés).

Exemple :

Formation.html

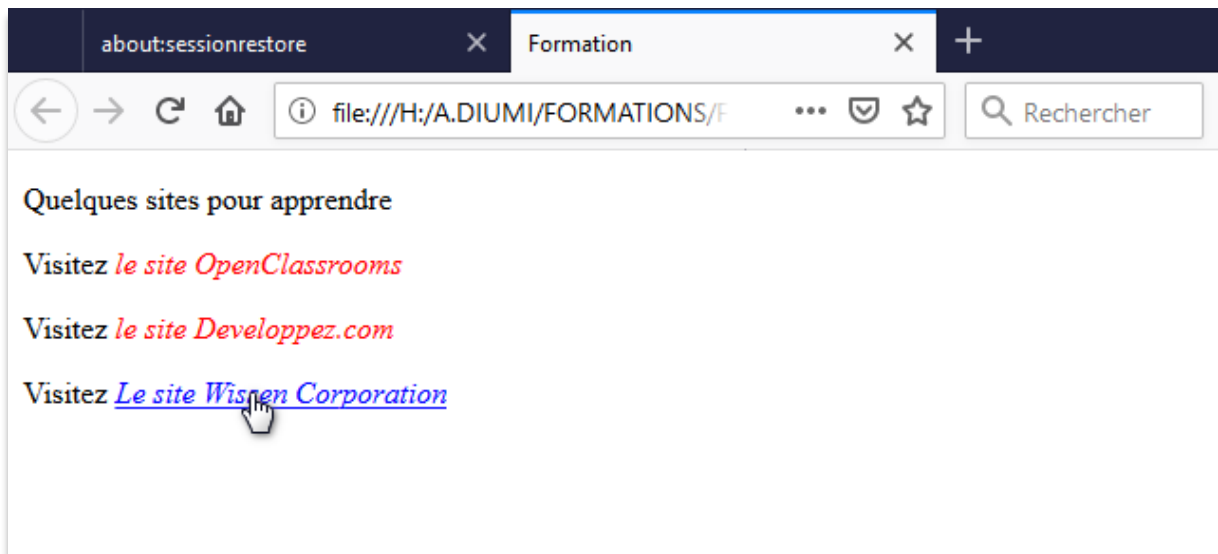
```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
<p> Quelques sites pour apprendre </p>
<p> Visitez <a href="www.OpenClassrooms.com" > le site OpenClassrooms </a></p>
<p> Visitez <a href="www.Developpez.com"> le site Developpez.com </a></p>
<p> Visitez <a href="www.wissenCorporation"> Le site Wissen Corporation </a></p>
```

```
</body>
</html>
```

## Forme.css

```
a /* Liens par défaut (non survolés) */
{
    color: red;
    font-style: italic;
    text-decoration: none;
}
a:hover /* Apparence au survol des liens */
{
    text-decoration: underline;
    color: green;
}
```

Résultat :



### *b) Lors d'un clic sur un élément*

On peut appliquer un style particulier à un élément au moment du clic en utilisant le pseudo-format : active

```
selecteur :active
{
}
```

En pratique, il n'est utilisé que sur les liens. Le lien gardera cette apparence très peu de temps : en fait, le changement intervient lorsque le bouton de la souris est enfoncé. En clair, ce n'est pas forcément toujours bien visible.

#### **Exemple :**

Reprenons l'exemple html ci-dessus, changeons la couleur du lien lors du clic :

Forme.css

```
a /* Liens par défaut (non survolés) */
{
    color: red;
    font-style: italic;
    text-decoration: none;
}

a:active /* Quand le visiteur clique sur le lien */
{
    color: yellow;
    text-decoration: decoration;
}
```

### *c) Lorsque l'élément est sélectionné*

Pour appliquer un style particulier à un élément lorsqu'il est sélectionné, on utilise le pseudo-format : focus

```
a:focus /* Quand le visiteur sélectionne le lien */
{
}
}
```

Ce pseudo-format s'applique aussi sur les formulaires (voir plus loin).

#### d) Lorsque le lien a déjà été consulté

Par défaut, le navigateur colore le lien déjà consulté (lien vers une page déjà vue) en violet, mais il est possible d'appliquer un style particulier à ce lien grâce au pseudo-modèle : visited

```
a:visited /* Quand le visiteur a déjà vu la page concernée */
{
}
}
```

### II.7.2 Pseudo-classe descendante

Permet d'appliquer un style à un descendant particulier d'un sélecteur donné, le premier ou le dernier enfant par exemple. Je ne veux pas beaucoup parler sur cette catégorie. A titre d'informations sachez que pour cette catégorie, on retrouve :

: first-child, concerne le premier enfant d'un sélecteur même si celui-ci dispose de sélecteurs frères.

### II.7.3 Les pseudo-éléments de texte

Ils permettent d'appliquer un style à une partie du texte délimitée par les balises auxquelles ils s'appliquent. Dans cette catégorie, on retrouve :

#### a) :first-line

Agit sur la première ligne d'un paragraphe.

Exemple

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel="stylesheet" href="forme.css" />
  </head>
  <body>
    <p> C'est un groupe formé essentiellement de meilleurs étudiants de nos groupes d'étude et d'encadrement qui se sont fixés plusieurs objectifs dont le premier est celui du partage de la connaissance en Informatique et en mathématique.
```

```
Il y a une naissance en toute connaissance, Thucydide a dit : « Avoir des connaissances sans les partager, c'est se mettre au niveau de celui qui n'a pas d'idées ». Wissen Corporation est là pour assurer votre formation, votre encadrement suivant vos désirs. </p>
```

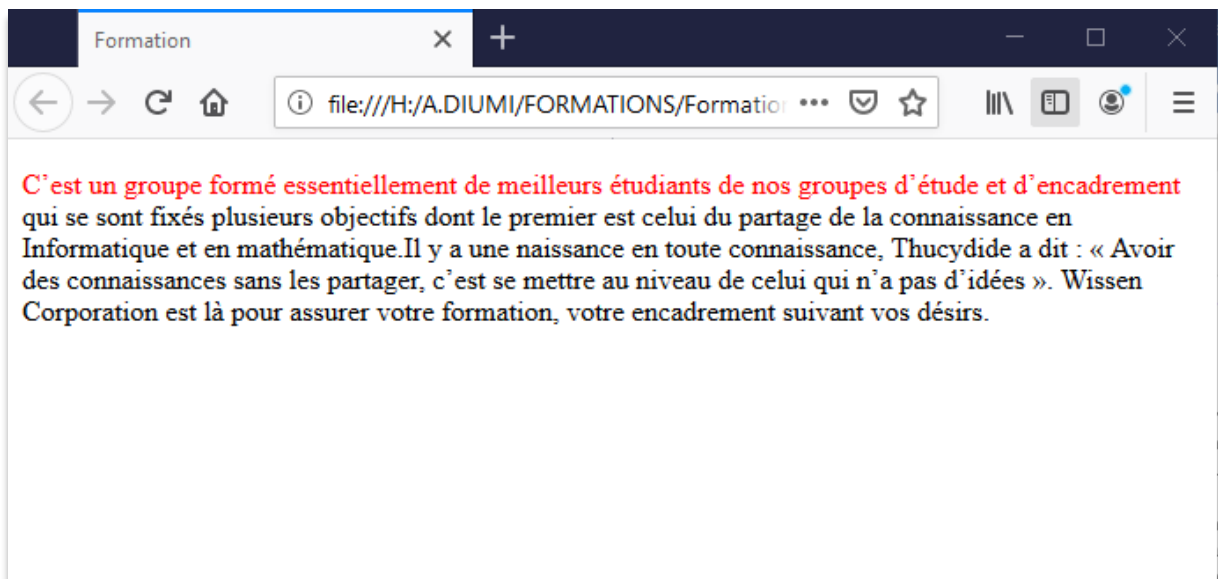
```
</body>  
</html>
```

Si je veux que la première ligne du paragraphe soit en rouge, j'écrirai :

Forme.css

```
p:first-line  
{  
  color: red;  
}
```

Résultat :



## b) :First-letter

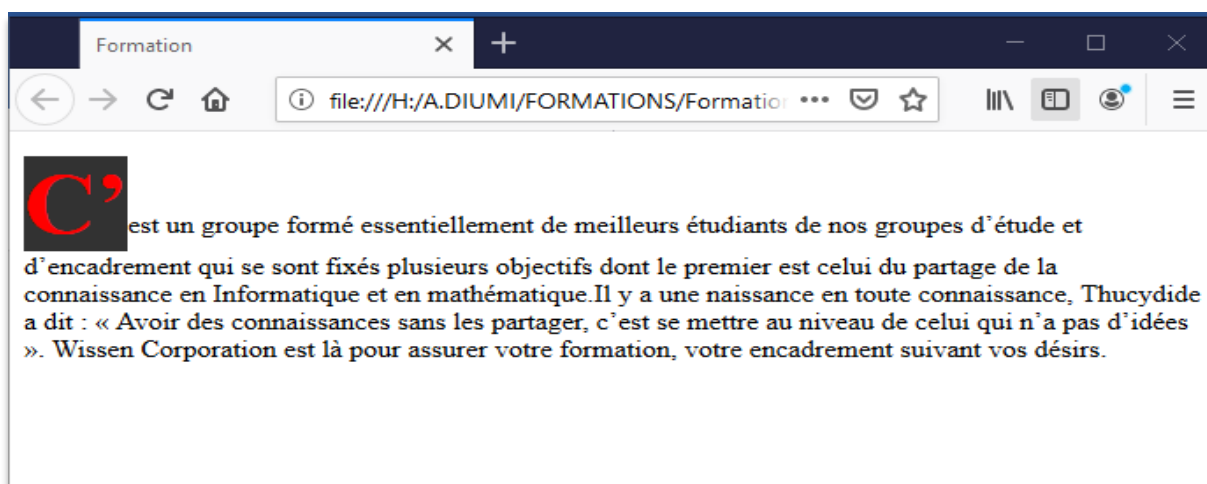
Agit sur la première lettre d'un paragraphe ou d'un bloc.

Reprenons le même exemple et je veux cette fois ci ajouter une couleur de fond à la première lettre du paragraphe, la mettre en gras et aussi agrandir la taille de la police :

Forme.css

```
p:first-letter
{
  background-color: rgb(50,50,50);
  font-size:3.5em;
  font-weight:bold;
  color:red;
}
```

Résultat :





## 11.7.4 Les pseudos éléments : after et : before

Ces pseudos éléments permettent de rajouter du texte, une image et des caractères avant et après un sélecteur donné. Ils sont associés à une propriété spécifique – « content » comme le montre la figure suivante :

```
selecteur:after
{
    content: valeur;
}

selecteur:before
{
    content: valeur;
}
```

Prenons l'exemple d'une citation dans l'exemple suivant :

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel="stylesheet" href="forme.css" />
  </head>
  <body>

  <blockquote>Petit poisson deviendra grand, pourvu que Dieu lui prête vie</blockquote>

</body>
</html>
```

Résultat :



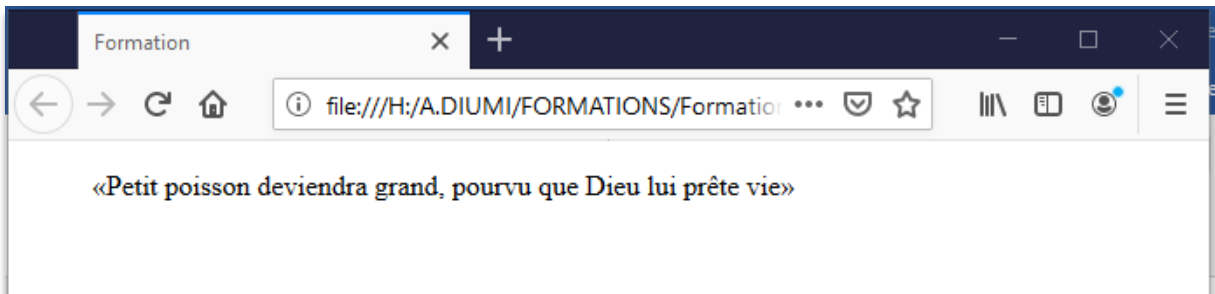
Si on souhaite que la citation soit entourée des guillemets (« ») ; c'est-à-dire qu'on ajoute des guillemets au début et à la fin de la citation, il suffit d'écrire :

Forme.css

```
blockquote:before{
  content:"«";
}

blockquote:after {
  content:"»";
}
```

Résultat :



## II.8 LES MARQUEURS DES LISTES

Dans cette partie, nous allons voir quelques options sur les listes ordonnées ou non ordonnées.

### II.8.1 La propriété list-style-type

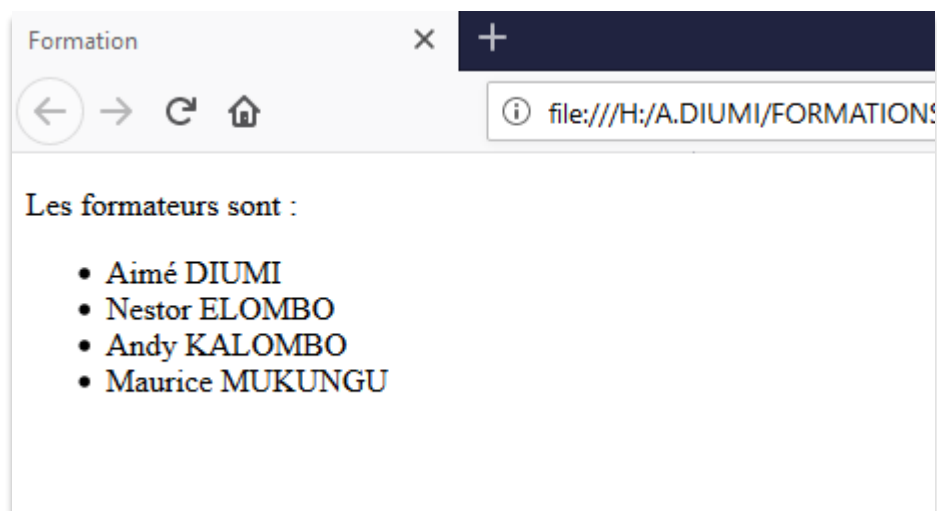
Cette propriété permet de déterminer l'apparence de la puce ou le style de numérotation de la liste. Les valeurs possibles sont : disc, circle, square, decimal, decimal-leading-zero, upper-roman, lower-roman, upper-alpha, lower-alpha, armenian, georgian, lower-greek. Nous allons essayer d'étudier le comportement de chaque valeur.

Prenons l'exemple suivant :

Formation.html

```
<p> Les formateurs sont :</p>
<ul>
  <li> Aimé DIUMI </li>
  <li> Nestor ELOMBO </li>
  <li> Andy KALOMBO </li>
  <li> Maurice MUKUNGU </li>
</ul>
```

Résultat :



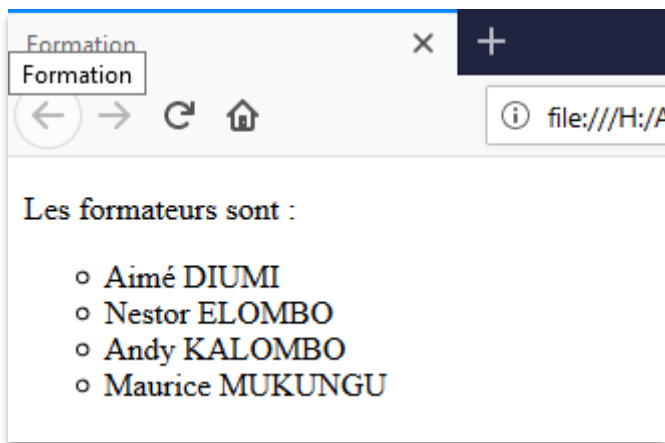
Par défaut, la puce utilisée pour la liste non ordonnée dépend du navigateur, mais avec la propriété list-style-type, il est possible de choisir le type de puce à afficher devant les éléments de la liste.

- ❖ Avec la valeur **circle** : cercle vide

Forme.css

```
li
{
  list-style-type: circle;
}
```

Résultat :

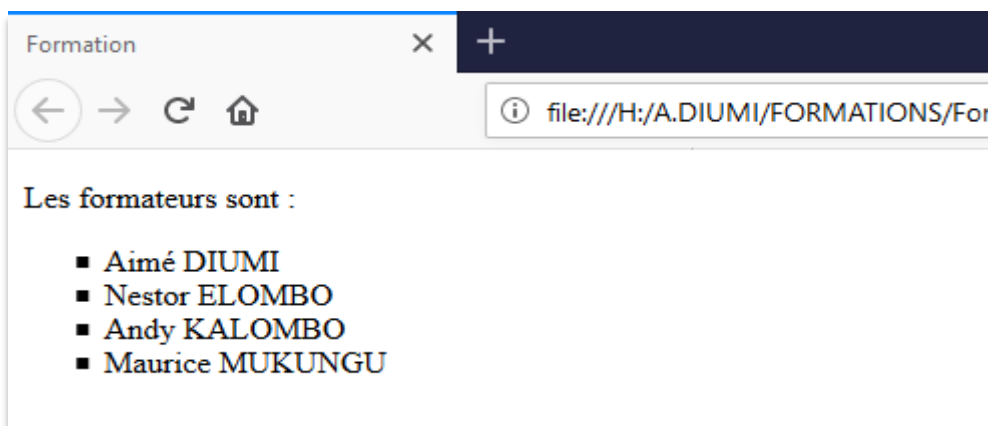


- ❖ Avec la valeur **disc** : cercle plein (voir exemple ci-haut)
- ❖ Avec la valeur **square** : carré

Forme.css

```
li
{
  list-style-type: square;
}
```

Résultat :

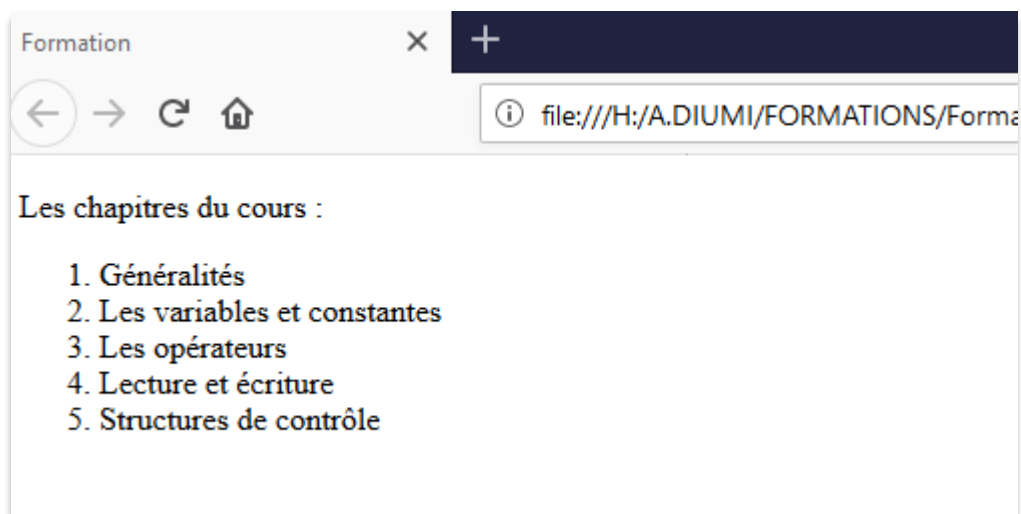


Pour illustrer les autres valeurs, utilisons la liste ordonnée ci-après :

Formation.html

```
<p> Les chapitres du cours :</p>
<ol>
  <li> Généralités </li>
  <li> Les variables et constantes </li>
  <li> Les opérateurs </li>
  <li> Lecture et écriture </li>
  <li> Structures de contrôle </li>
</ol>
```

Résultat :



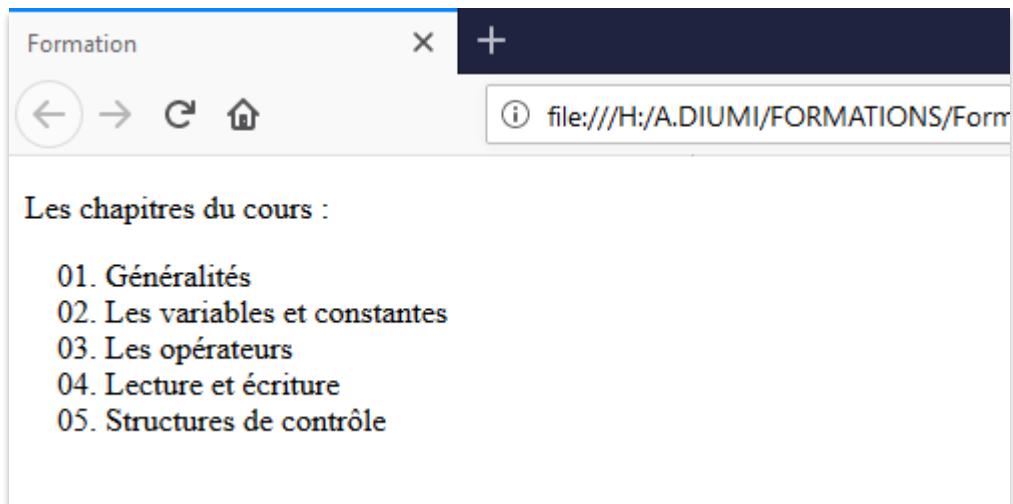
Pour la liste ordonnée, la notation par défaut est **decimal**, faisons le tour d'horizon des autres valeurs de la propriété list-style-type :

❖ Avec la valeur **decimal-leading-zero** :

Forme.css

```
li
{
  list-style-type: decimal-leading-zero;
}
```

Résultat :

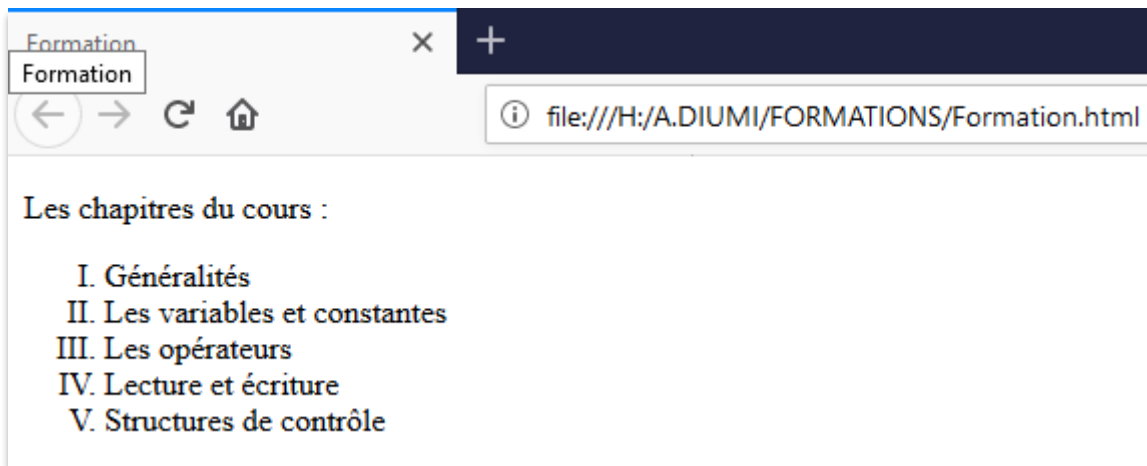


❖ Avec la valeur upper-roman : chiffres romains (majuscules)

Forme.css

```
li
{
  list-style-type: upper-roman;
}
```

Résultat :

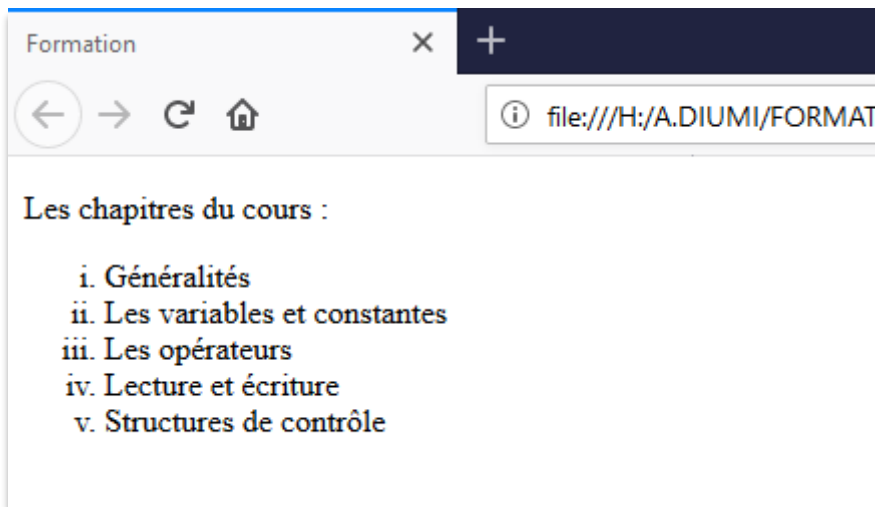


❖ Avec la valeur lower-roman : chiffres romains (minuscules)

Forme.css

```
li
{
  list-style-type: lower-roman;
}
```

Résultat :

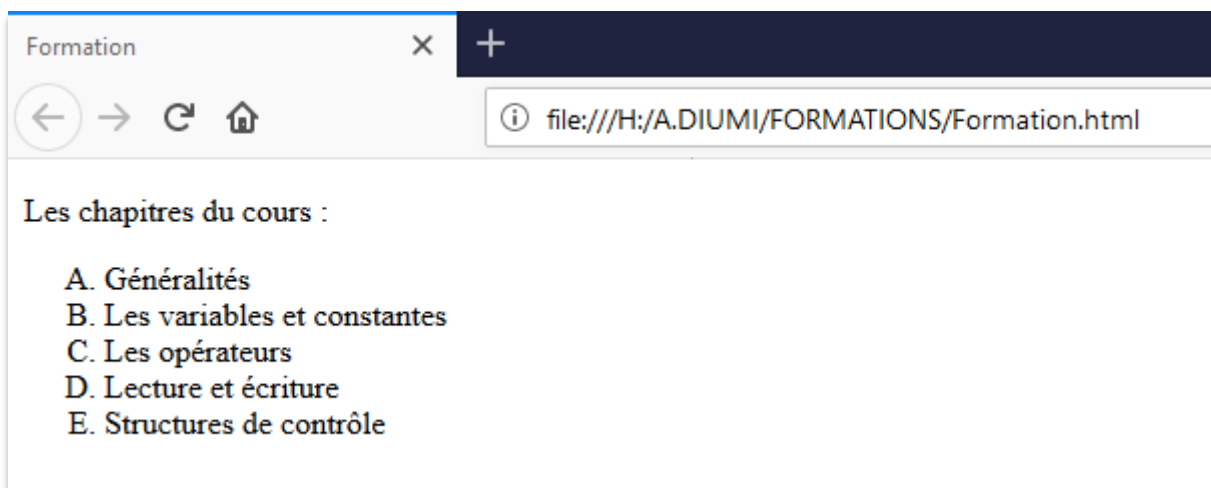


❖ Avec la valeur upper-alpha : lettres majuscules

Forme.css

```
li
{
  list-style-type:upper-alpha;
}
```

Résultat :

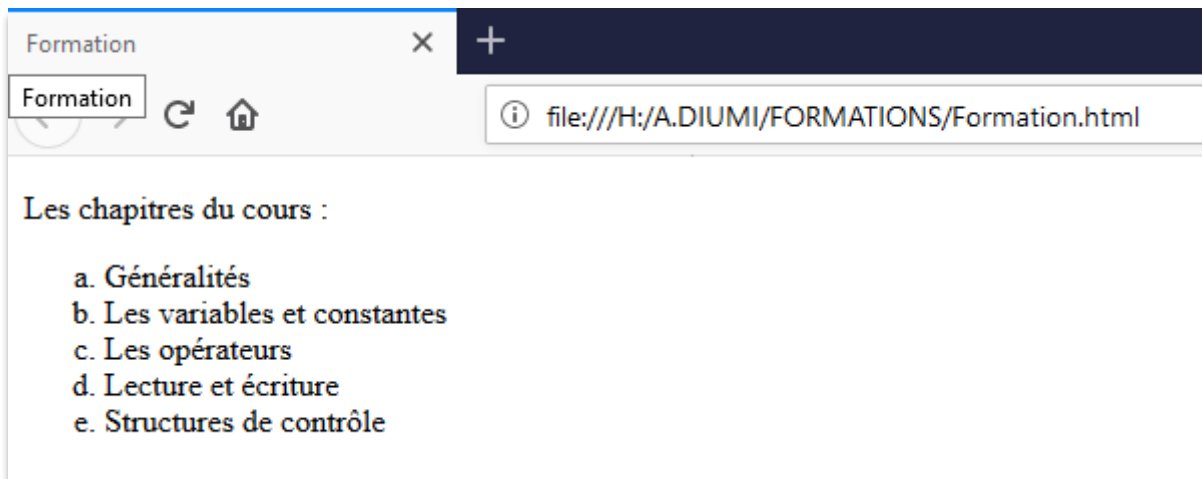


❖ Avec la valeur lower-alpha : lettres minuscules

Forme.css

```
li
{
  list-style-type:lower-alpha;
}
```

Résultat :

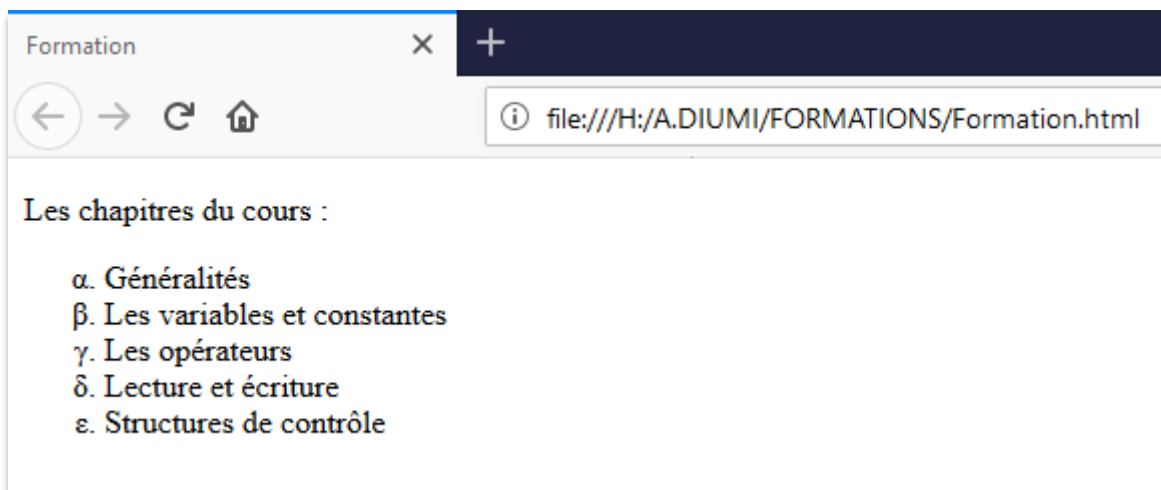


❖ Avec la valeur lower-greek : minuscules des lettres grecques

Forme.css

```
li
{
  list-style-type:lower-greek;
}
```

Résultat :





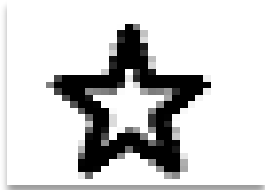
## II.8.2 La propriété list-style-image

Elle permet d'utiliser un marqueur graphique (image) comme puce aux éléments de la liste. Elle s'utilise comme suit :

```
list-style-image:url("chemin_de_l'image");
```

Exemple

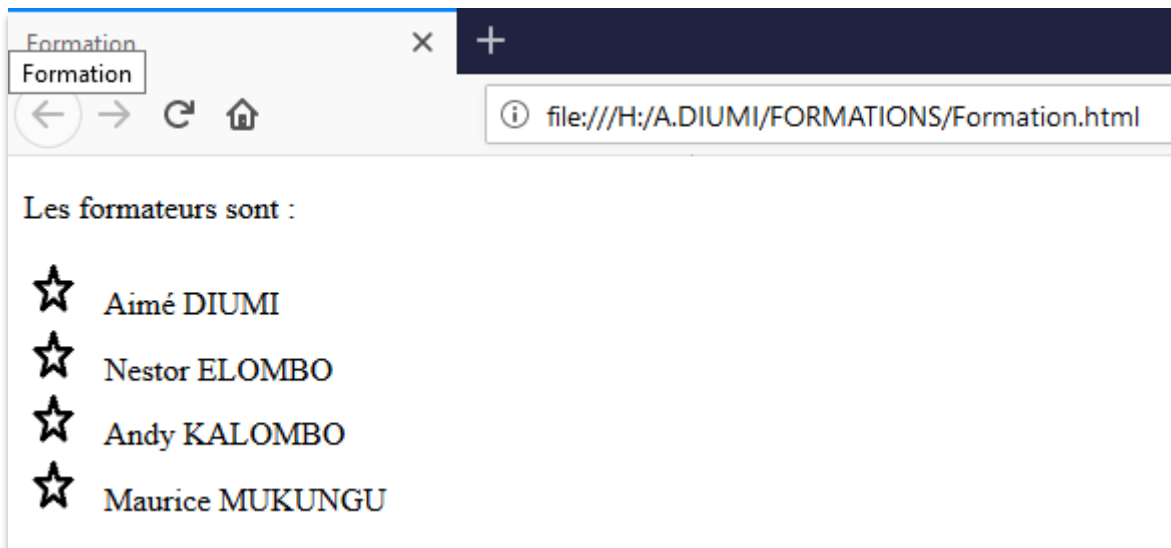
Considérons la liste non ordonnée créée ci-dessus, on veut utiliser l'image ci-après (puce.png) située dans le même dossier que mon fichier html :



Forme.css

```
li
{
  list-style-image:url("puce.png");
}
```

Résultat :



## II.8.3 La propriété list-style-position

Elle permet de déterminer la position du deuxième élément (et les suivants) de la liste par rapport à la puce ou de la numérotation. Elle peut prendre l'une des valeurs suivantes :

- ❖ La valeur **outside** : Tous éléments de la liste sont en retrait par rapport à la puce ou au numéro (valeur par défaut).
- ❖ La valeur **inside** : il y a retrait seulement pour le premier élément, les autres se s'alignent sur la puce.

Remarque

Il est possible de combiner les trois propriétés (list-style-type, list-style-image et list-style-position) dans la super propriété list-style comme le montre l'exemple suivant :

```
list-style:url("puce.png") inside circle;
```

Vous n'êtes pas obligés de donner une valeur pour chaque propriété et l'ordre des valeurs n'a pas d'importance.

## II.9 LES MODELES DE BOITES

### II.9.1 Introduction

La plupart des balises en html sont souvent dans l'une des catégories suivantes :

- inline : une balise de type inline se trouve à l'intérieur d'une balise block. Une balise inline ne crée pas de retour à la ligne, le texte qui se trouve à l'intérieur s'écrit donc à la suite du texte précédent, sur la même ligne. Elle occupe juste la largeur nécessaire.
- block : une balise de type block sur votre page web crée automatiquement un retour à la ligne avant et après. Votre page web sera en fait constituée d'une série de blocs les uns à la suite des autres. Une balise de type block occupe toute la largeur disponible.

On peut résumer tout ça avec un exemple

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
<p>Wikipédia présente Visual basic comme un langage de
programmation événementielle de troisième génération ainsi qu'un
environnement de développement intégré, créé par Microsoft pour son
modèle COM.<a href="http://www.wikipedia.com" > Voir l'article </a> (Pour plus
d'informations, cliquez sur le lien)
  <br/>Visual basic est directement dérivé du BASIC (acronyme
pour Beginner's All purpose Symbolic Instruction Code ) et permet le
développement rapide d'applications, la création d'interfaces utilisateur
graphiques, l'accès aux bases de données en utilisant les technologies DAO,
ADO et RDO, ainsi que la création de contrôle ou objets Active X. </p>
</body>
</html>
```

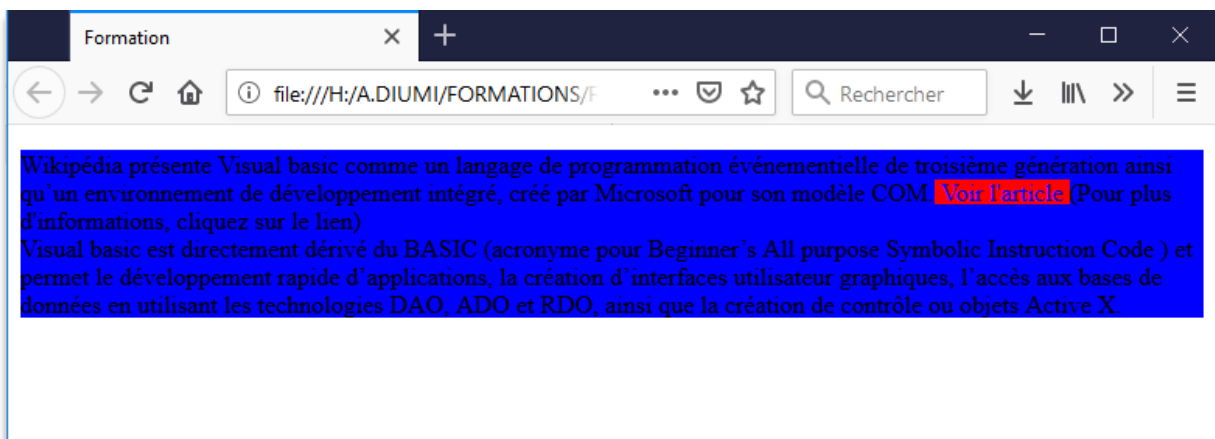
J'ai une balise <p> qui est de type block et <a> qui est de type inline, ajoutons une couleur de fond à ces deux balises pour voir la différence

## Forme.css

```
p
{
  background-color: blue;
}

a
{
  background-color: red;
}
```

Résultat :



Vous avez sans doute remarqué que le lien occupe jusque l'espace nécessaire.

Les blocs se placent les uns en dessous des autres comme le montre l'exemple suivant :

## Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
    <h1> LES BASES DU LANGAGE ET MODE CONSOLE </h1>
```

```

<p class="para1">Wikipédia présente Visual basic comme un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, créé par Microsoft pour son modèle COM. Pour plus d'informations, cliquez sur le lien
<br/>Visual basic est directement dérivé du BASIC (acronyme pour Beginner's All purpose Symbolic Instruction Code ) et permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôle ou objets Active X. </p>

<p class="para2">Pour coder dans un langage de programmation, il faut des outils adaptés : un éditeur de texte, un compilateur ainsi qu'un débogueur.<br/> Il y a des programmes qui combinent les trois outils et ces programmes sont appelés Environnement de développement intégré (EDI) ou IDE en anglais, pour Integrated development Environment. </p>
</body>
</html>

```

## Forme.css

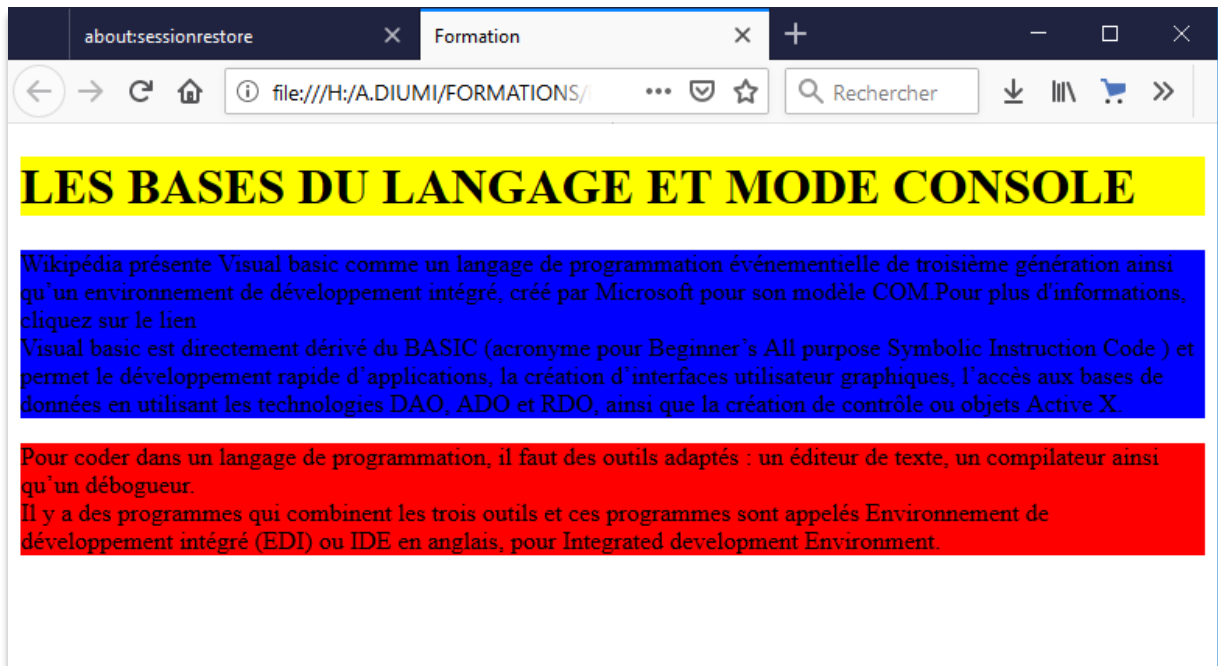
```

.para1
{
  background-color: blue;
}

.para2
{
  background-color: red;
}
h1
{
  background-color: yellow;
}

```

Résultat :



Sachez qu'il est aussi possible d'imbriquer un bloc à l'intérieur d'un autre bloc.

Exemple

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
    <section >
      <h1> LES BASES DU LANGAGE ET MODE CONSOLE </h1>
      <p class="para1">Wikipédia présente Visual basic comme un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, créé par Microsoft pour son modèle COM. Pour plus d'informations, cliquez sur le lien
      <br/>Visual basic est directement dérivé du BASIC (acronyme pour Beginner's All purpose Symbolic Instruction Code) et permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôle ou objets Active X. </p>
      <p class="para2">Pour coder dans un langage de programmation, il faut des outils adaptés : un éditeur de texte, un compilateur ainsi qu'un débogueur.<br/> Il y a des programmes qui combinent les trois outils et ces programmes sont app
```

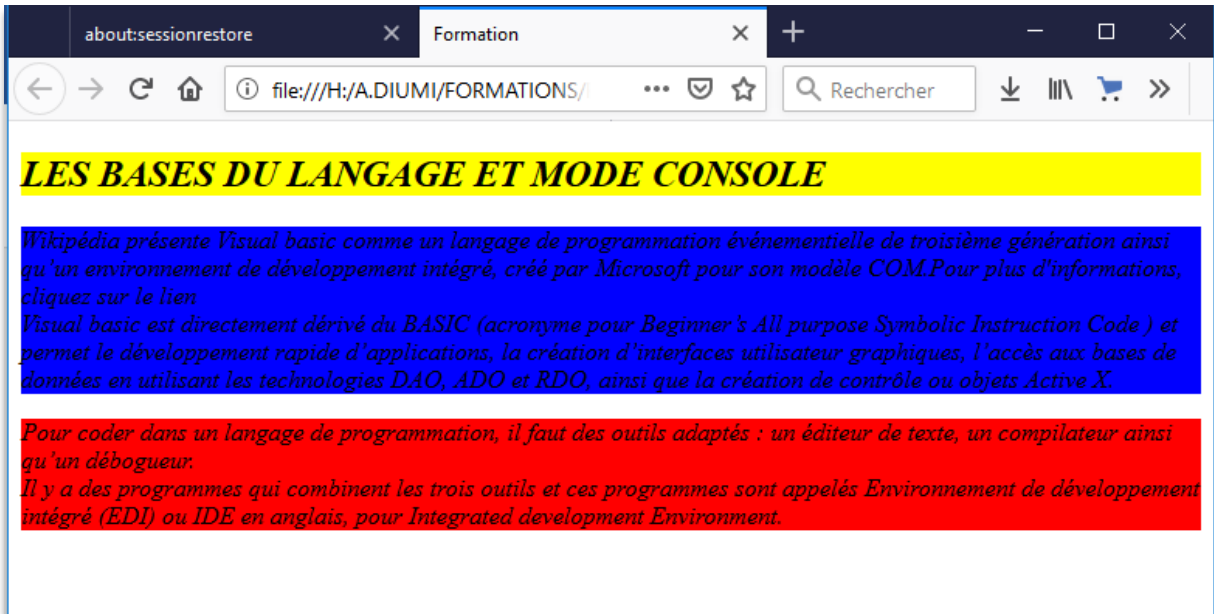
```
elés Environnement de développement intégré (EDI) ou IDE en anglais, pour Inte  
grated development Environment. </p>  
</section>  
</body>  
</html>
```

La balise <section> qui est de type block contient les balises <h1> et <p> qui sont aussi de type block. Dans le code css suivant, j'ai mis le texte de <section> en italique, cela prendra effet sur <h1> et <p> car elles sont contenues dans <section>

Forme.css

```
.para1  
{  
  background-color: blue;  
}  
  
.para2  
{  
  background-color: red;  
}  
h1  
{  
  background-color: yellow;  
}  
  
section  
{  
  font-style: oblique;  
}
```

On peut voir le résultat :



## II.9.2 Les dimensions

Chaque bloc a une taille précise, il a une largeur et une hauteur. Par défaut, un bloc prend 100% de la largeur disponible et la hauteur dépend de son contenu. Pour modifier la largeur ou la hauteur, on a deux propriétés en CSS :

- **width** : c'est la largeur du bloc. À exprimer en pixels (px) ou en pourcentage (%).
- **height** : c'est la hauteur du bloc. A exprimer aussi en px ou en %.

Exemple :

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
    <h1> LES BASES DU LANGAGE ET MODE CONSOLE </h1>
    <p class="para1">Wikipédia présente Visual basic comme un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, créé par Microsoft pour son modèle COM.<br/>
    Visual basic est directement dérivé du BASIC (acronyme pour Beginner's All purpose Symbolic Instruction Code) et permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôle ou objets Active X. </p>
```



```
<p class="para2">Pour coder dans un langage de programmation, il faut des outils adaptés : un éditeur de texte, un compilateur ainsi qu'un débogueur.<br/> Il y a des programmes qui combinent les trois outils et ces programmes sont appelés Environnement de développement intégré (EDI) ou IDE en anglais, pour Integrated development Environment. </p>

</body>
</html>
```

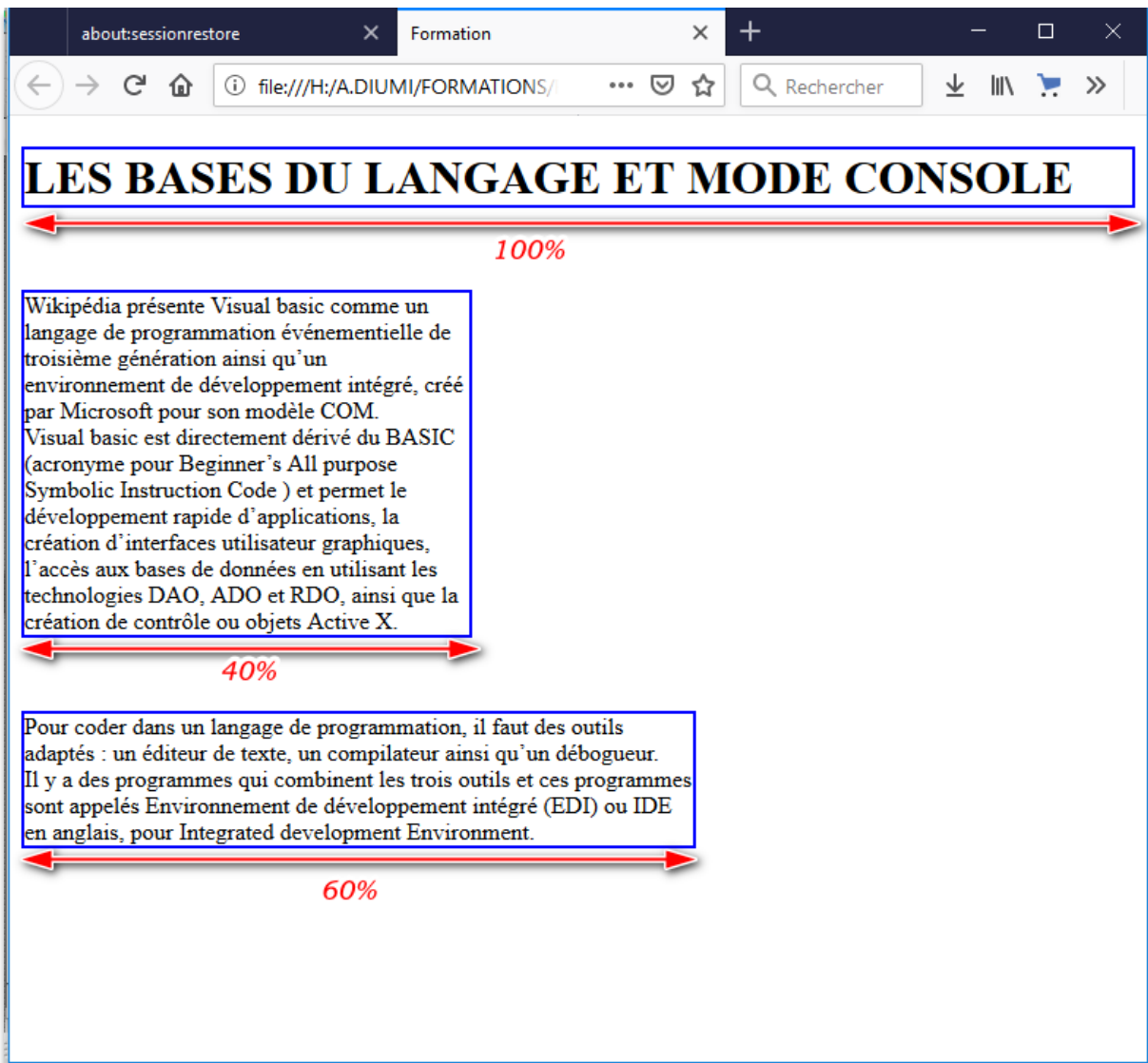
J'ajoute une bordure à chaque élément pour bien voir le résultat. Pour le premier paragraphe, je définis la largeur à 40% (c'est-à-dire que le bloc occupera 40% de la largeur disponible du navigateur) et pour le deuxième paragraphe, je mets 60%.

Forme.css

```
.para1
{
  border: 2px blue solid;
width: 40%;
}

.para2
{
  border: 2px blue solid;
width: 60%;
}
h1
{
  border: 2px blue solid;
}
```

Résultat :



On peut aussi indiquer la valeur précise en pixels, mais il est recommandé de la mettre en pourcentage car la largeur va s'adapter automatiquement à la taille de l'écran du visiteur.

On peut demander à ce qu'un bloc ait des dimensions minimales et maximales. C'est très pratique car cela nous permet de définir des dimensions « limites » pour que notre site s'adapte aux différentes résolutions d'écran de nos visiteurs :

- **min-width** : largeur minimale ;
- **min-height** : hauteur minimale ;
- **max-width** : largeur maximale ;
- **max-height** : hauteur maximale.

## Exemple

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
    <h1> LES BASES DU LANGAGE ET MODE CONSOLE </h1>

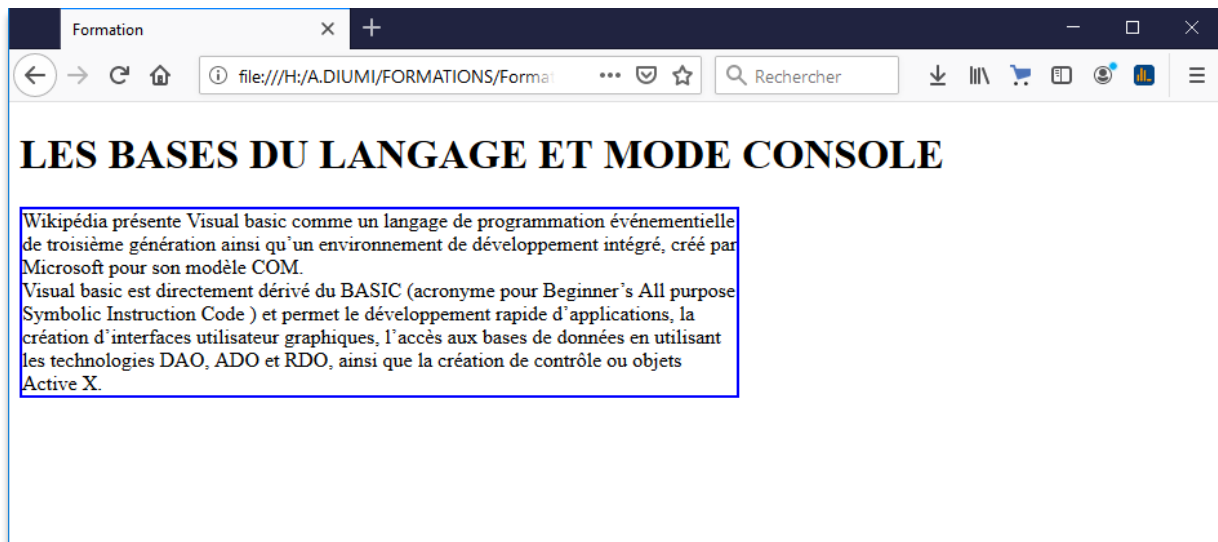
    <p >Wikipédia présente Visual basic comme un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, créé par Microsoft pour son modèle COM.<br/>
    Visual basic est directement dérivé du BASIC (acronyme pour Beginner's All purpose Symbolic Instruction Code) et permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôle ou objets Active X. </p>
  </body>
</html>
```

Je veux que mon paragraphe occupe 60% de la largeur disponible, mais que cette largeur ne soit pas inférieure à 500px :

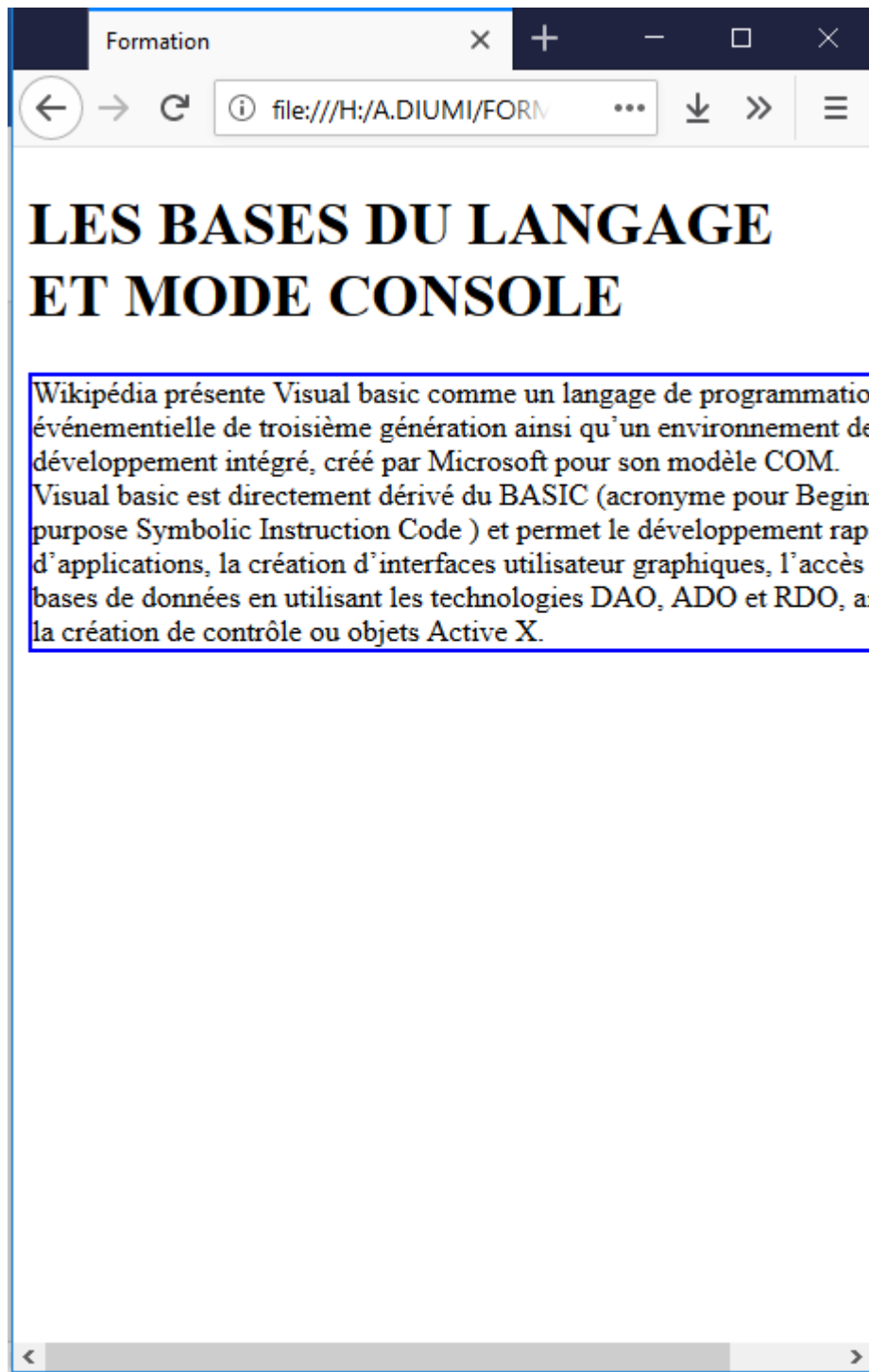
Forme.css

```
p
{
  border: 2px blue solid;
width: 60%;
min-width: 200px;
}
```

Résultat :



Si j'essaie de réduire la largeur du navigateur et que celle-ci devient inférieure à 500px (la valeur minimale de la largeur de mon paragraphe), voici ce qui va se passer :

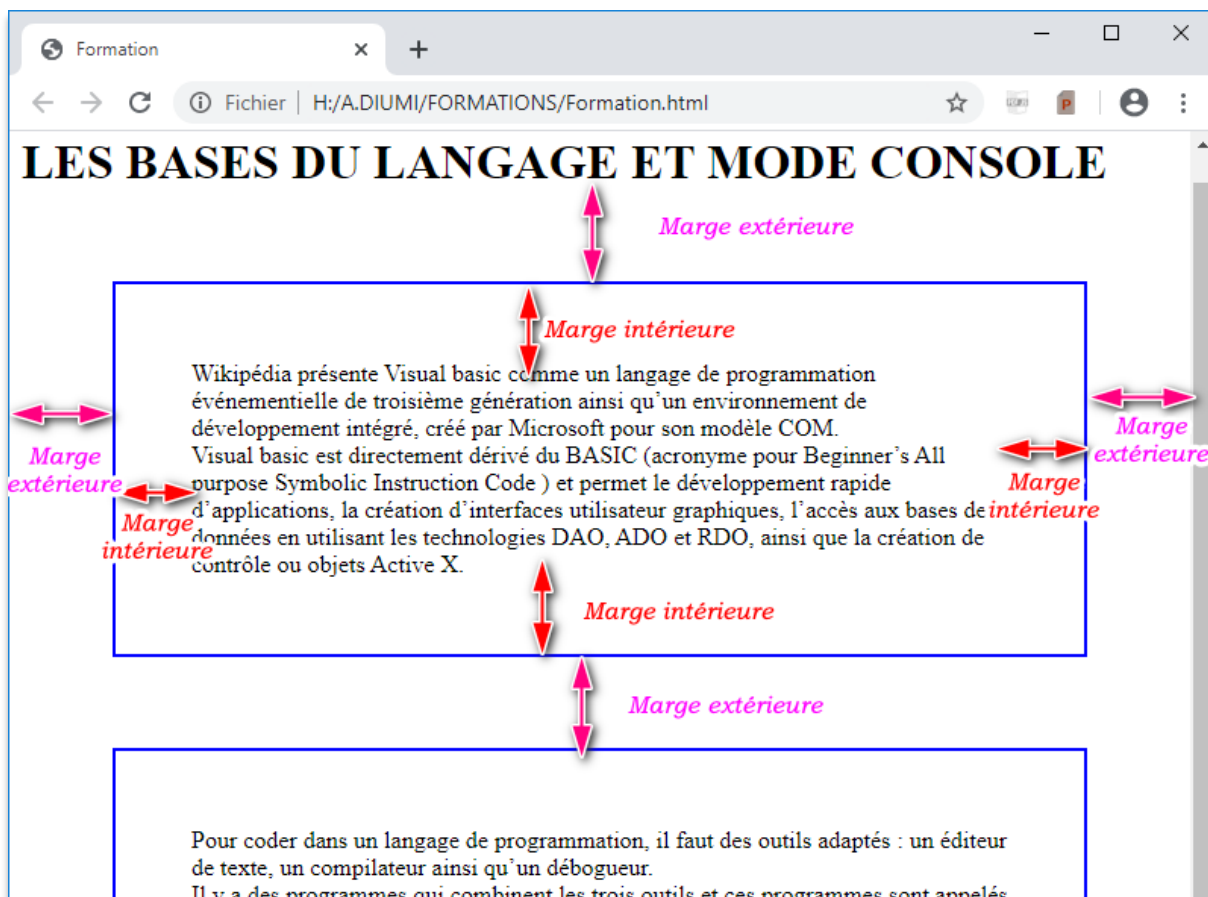


Le paragraphe se force à occuper 500px et le navigateur a ajouté une barre de défilement car la largeur du paragraphe est maintenant supérieure à celle du navigateur.

## II.9.3 Les marges

Chaque bloc possède deux types de marges :

- Marge intérieure : l'espace entre le contenu et la limite (frontière) du block.
- Marge extérieure : l'espace entre le block et les blocks voisins



Exemple

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
    <h1> LES BASES DU LANGAGE ET MODE CONSOLE </h1>

    <p >Wikipédia présente Visual basic comme un langage de
    programmation événementielle de troisième génération ainsi qu'un
    environnement de développement intégré, créé par Microsoft pour son
```

```

modèle COM.<br/>
Visual basic est directement dérivé du BASIC (acronyme
pour Beginner's All purpose Symbolic Instruction Code ) et permet le
développement rapide d'applications, la création d'interfaces utilisateur
graphiques, l'accès aux bases de données en utilisant les technologies DAO,
ADO et RDO, ainsi que la création de contrôle ou objets Active X. </p>

<p>Pour coder dans un langage de programmation, il faut des outils
adaptés : un éditeur de texte, un compilateur ainsi qu'un débogueur.<br/> Il y
a
des programmes qui combinent les trois outils et ces programmes sont
appelés Environnement de développement intégré (EDI) ou IDE en anglais,
pour Integrated development Environment. </p>

<div class="editeur">L'éditeur de texte nous permet d'écrire le code source du
programme.
Le compilateur (Interpréteur) pour transformer le code source en binaire qui
est le seul langage compréhensible par la machine.<br/>
Le débogueur permet de traquer les erreurs dans le programme.
Pour Visual basic, nous pouvons utiliser : Visual studio, Visual studio
Express, SharpDevelop, MonoDevelop etc. </div>

<div class="variables">I. LES VARIABLES </div>
</body>
</html>

```

Forme.css

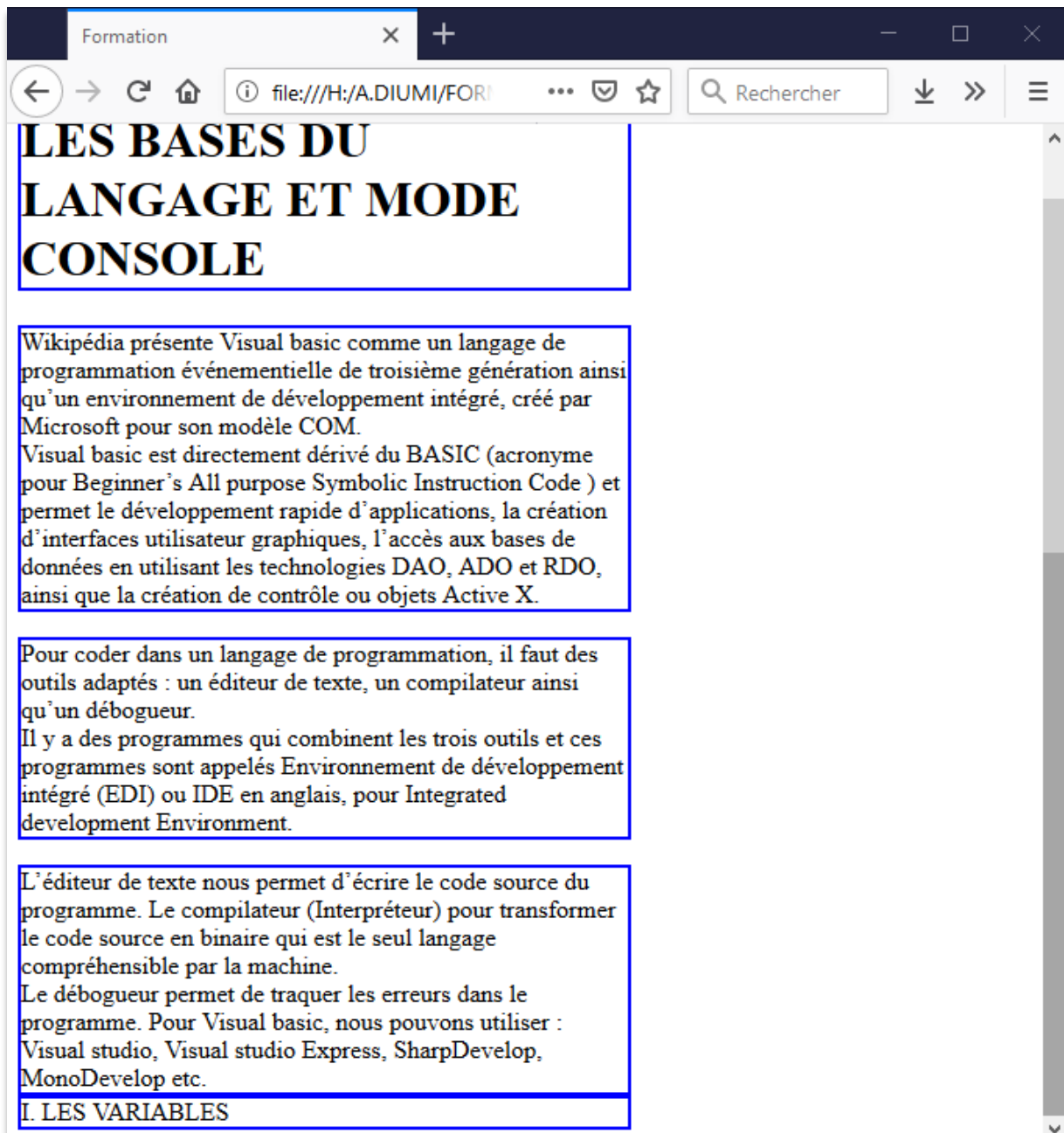
J'ai ajouté une bordure pour chaque bloc :

```

p,h1, .editeur, .variables
{
border: 2px solid blue;
width: 60%;
}

```

Résultat :



Vous avez sans doute remarqué qu'il y a des balises qui possèdent par défaut des marges extérieures et pas de marges intérieures (cas des balises `p` et `h1` pour notre exemple) et d'autres qui ne possèdent par défaut pas de marges intérieures et extérieures (cas de la balises `div`). Retenez que la valeur par défaut de la marge intérieure et extérieure n'est pas la même pour toutes les balises.

En CSS, on peut modifier la taille des marges avec les deux propriétés suivantes :

- ***padding*** : indique la taille de la marge intérieure. Cette taille s'exprime généralement en pixels (px)



- **margin** : indique la taille de la marge extérieure. On l'exprime aussi souvent en px.

Sachez que ces deux propriétés s'appliquent aux quatre coins du bloc, pour appliquer une marge différente à l'un de côté du bloc, on a le choix entre :

#### *Pour la propriété margin*

- **margin-top** : marge extérieure en haut ;
- **margin-bottom** : marge extérieure en bas ;
- **margin-right** : marge extérieure à droite.
- **margin-left** : marge extérieure à gauche ;

#### *Pour la propriété padding*

- **padding-top** : marge intérieure en haut ;
- **padding-bottom** : marge intérieure en bas ;
- **padding-left** : marge intérieure à gauche ;
- **padding-right** : marge intérieure à droite.

Exemple :

Formation.html

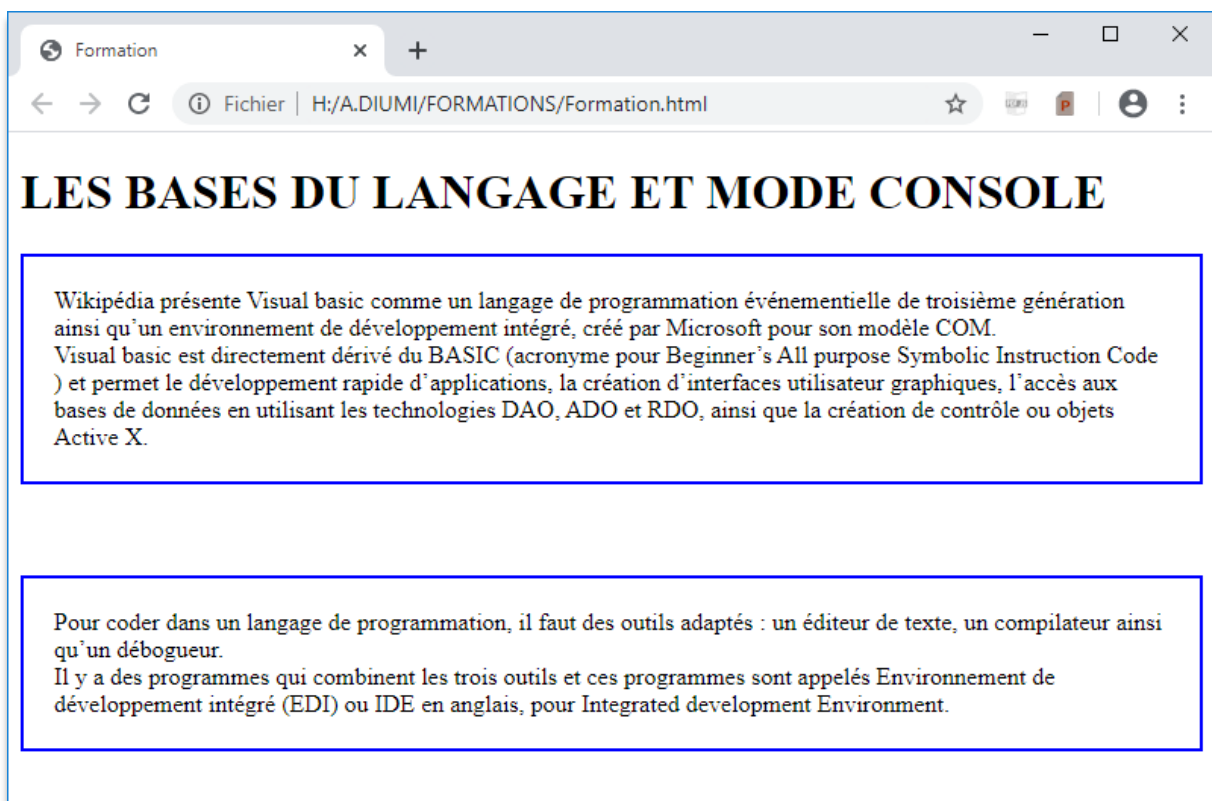
```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
    <h1> LES BASES DU LANGAGE ET MODE CONSOLE </h1>
    <p class="para1">Wikipédia présente Visual basic comme un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, créé par Microsoft pour son modèle COM.<br/>
    Visual basic est directement dérivé du BASIC (acronyme pour Beginner's All purpose Symbolic Instruction Code) et permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôle ou objets Active X. </p>
    <p class="para2">Pour coder dans un langage de programmation, il faut des outils adaptés : un éditeur de texte, un compilateur ainsi qu'un débogueur.<br/>
    Il y a des programmes qui combinent les trois outils et ces programmes sont appelés Environnement de développement intégré (EDI) ou IDE en anglais, pour Integrated development Environment. </p>
  </body>
</html>
```

Je veux modifier la marge extérieure du premier paragraphe, qu'elle soit petite entre le premier paragraphe et le titre et assez grande entre les deux paragraphes.

Forme.css

```
.para1
{
border: 2px blue solid;
margin-top:20px;
margin-bottom: 60px;
padding: 20px;
}
.para2
{
border: 2px blue solid;
padding:20px
}
```

Résultat :



Il y a d'autres façons de spécifier les marges avec les propriétés margin et padding.

### 1<sup>er</sup> raccourci :

On donne quatre valeurs à la propriété dans l'ordre suivant : la taille en haut, à droite, en bas et à gauche.

Exemple

```
padding: 10px 12px 5px 30px;
```

».

Signifie « 10 px de marge en haut, 12px à droite, 5 px en bas, 30 px à gauche ».

### 2<sup>e</sup> raccourci :

On donne une valeur pour la taille en haut et en bas et une autre valeur pour la taille en gauche et en droite :

Exemple :

```
margin: 10px 12px;
```

Signifie « 10 px de marge en haut et en bas, 12 px de marge à gauche et à droite ».

## II.9.4 Centrer les blocs

Avec Css, nous avons la possibilité de centrer des blocs. C'est même très pratique pour réaliser un design centré quand on ne connaît pas la résolution de l'écran du visiteur.

Voici les règles à respecter pour centrer un bloc :

- Donnez une largeur au bloc. Veillez à ce que le block n'ait pas la même largeur que le navigateur (c'est-à-dire que la largeur du bloc doit être inférieure à 100%) . Attribuez à la propriété **width** une valeur inférieure à la largeur du navigateur.
- Indiquez que vous voulez des marges extérieures automatiques, comme

ceci : 

```
margin: auto;
```

Exemple :

Reprenons le même code html ci-dessus. J'aimerais que les trois blocs (les deux paragraphes et le titre) soient centrés.

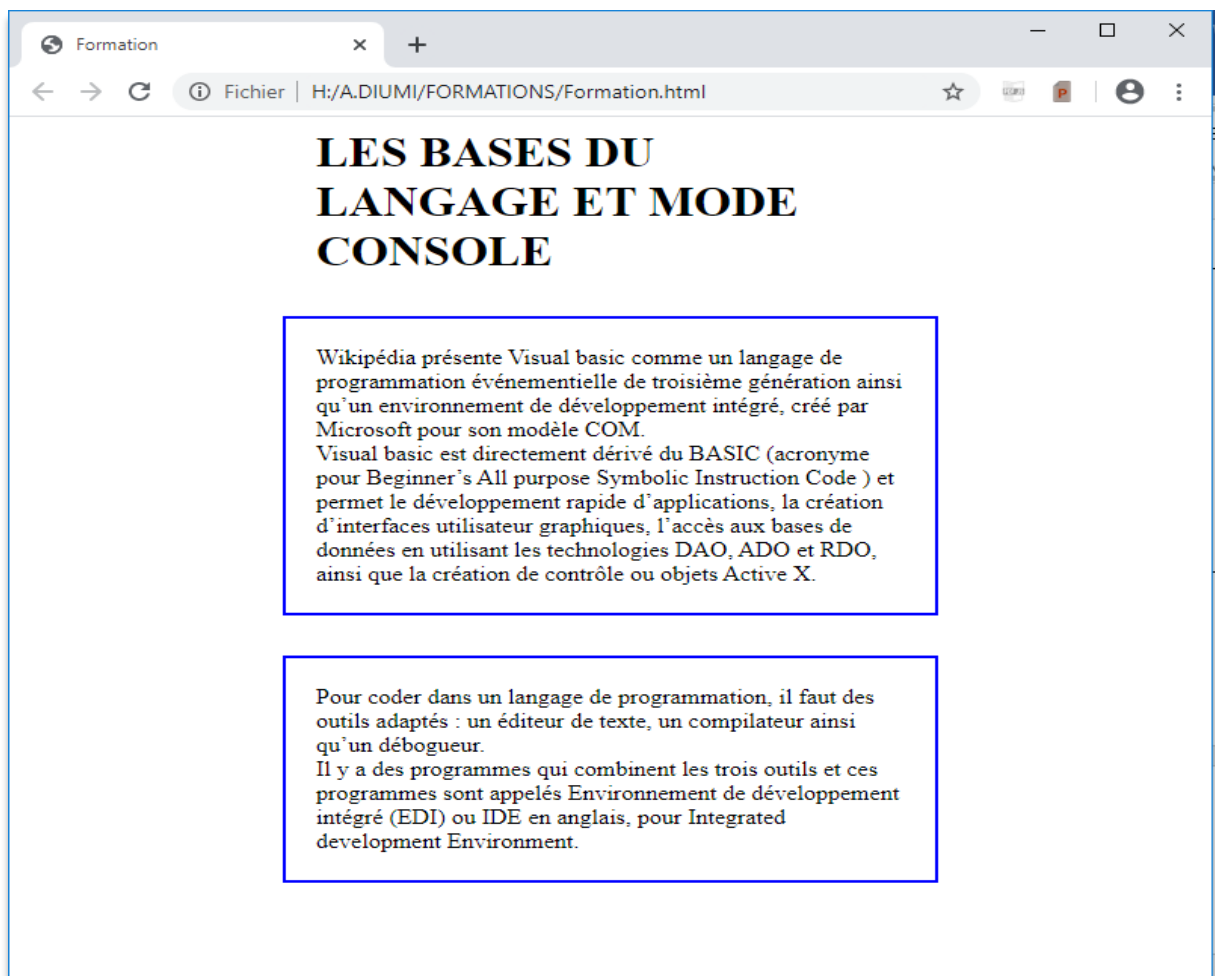
Forme.css

```
{  
  width: 50%;  
  margin: auto;  
  border: 2px blue solid;  
  padding: 20px;  
  margin-top: 30px;  
}
```

```
h1
{
width: 50%;
margin: auto;
}
```

Pour les paragraphes, j'ai ajouté une bordure.

Résultat :



## II.9.5 Dépassement du texte d'un bloc

En donnant des dimensions précises à un bloc, il peut arriver que le bloc devienne petit par rapport au texte qu'il contient. Heureusement qu'il existe des propriétés CSS qui permettent de gérer ce dépassement.

### a) La propriété *Overflow*

Commençons par un exemple

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
    <h1> LES BASES DU LANGAGE ET MODE CONSOLE </h1>

    <p class="para1">Wikipédia présente Visual basic comme un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, créé par Microsoft pour son modèle COM.<br/>
    Visual basic est directement dérivé du BASIC (acronyme pour Beginner's All purpose Symbolic Instruction Code) et permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôle ou objets Active X. <br/>
    Pour coder dans un langage de programmation, il faut des outils adaptés : un éditeur de texte, un compilateur ainsi qu'un débogueur. <br/>
    Il y a des programmes qui combinent les trois outils et ces programmes sont appelés Environnement de développement intégré (EDI) ou IDE en anglais, pour Integrated development Environment. </p>

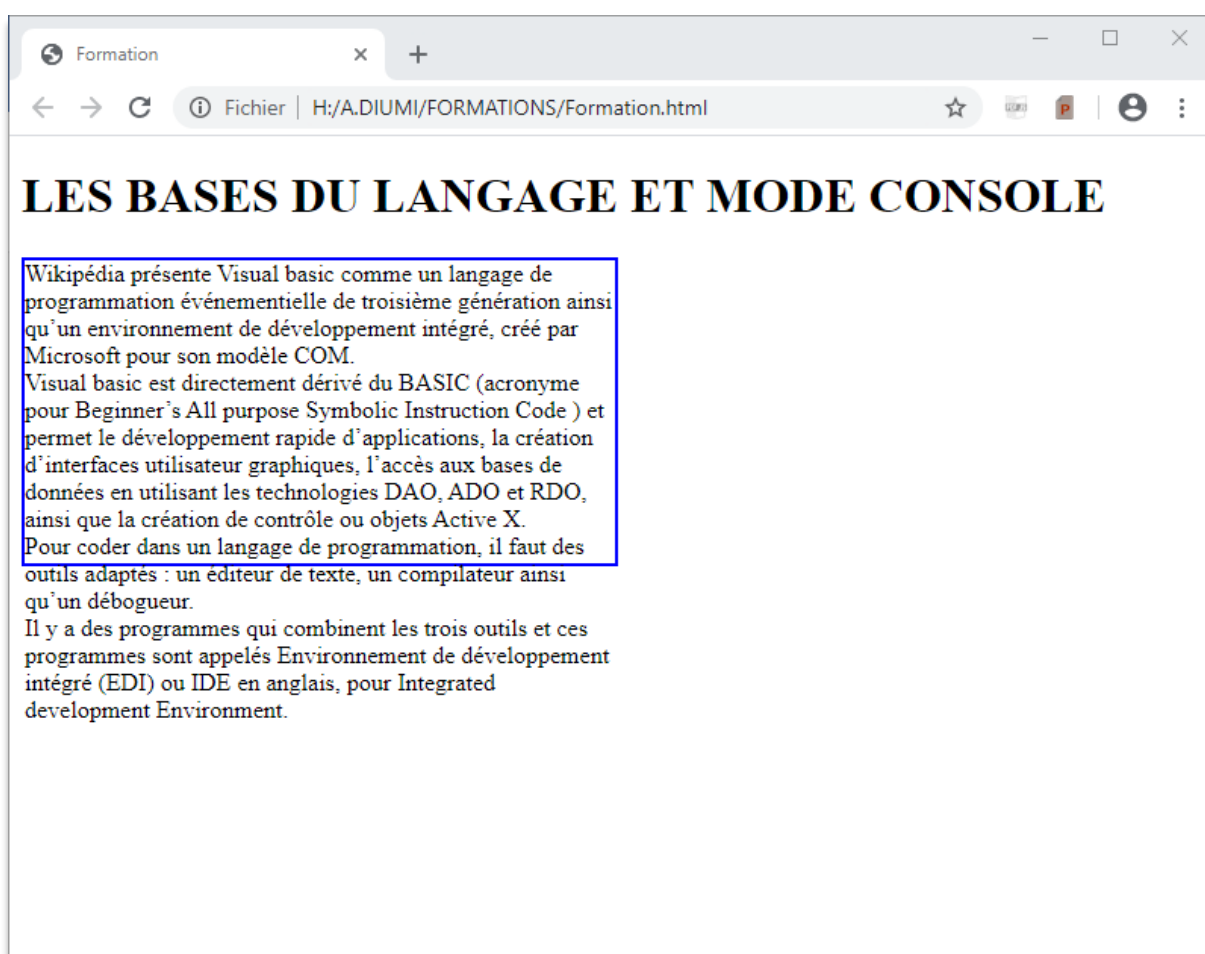
  </body>
</html>
```

J'aimerais que le paragraphe occupe 50% de la largeur disponible et 200 px de hauteur (j'ai ajouté une bordure et une marge intérieure) :

Forme.css

```
p
{
width: 50%;
height: 200px;
border: 2px blue solid;
padding: 20px;
}
```

Résultat :



Vous avez sans doute remarqué que le bloc ne peut plus contenir tout le texte car ses dimensions sont petites. C'est dans de telles situations qu'intervient la propriété **Overflow**. Avec cette propriété, on peut gérer le texte qui dépasse le bloc. Voici les valeurs possibles de cette propriété :

- **visible** : avec cette valeur, si le texte dépasse les limites du bloc, il reste visible et sort du bloc. C'est la valeur par défaut. C'est ce qui s'est appliqué à l'exemple précédent.

- **hidden** : si le texte dépasse les limites, il sera tout simplement coupé.

Exemple

Forme.css

```
p
{
width: 50%;
height: 200px;
border: 2px blue solid;
padding: 20px;

overflow: hidden;
}
```

Résultat :



Le texte est coupé et on ne pourra pas voir tout le texte.

- **scroll** : S'il y a dépassement, le texte sera coupé. Sauf que cette fois, le navigateur mettra en place des barres de défilement pour qu'on puisse lire l'ensemble du texte.

Exemple

Forme.css

```
p
{
width: 50%;
height: 200px;
border: 2px blue solid;
padding: 20px;

overflow: scroll;
}
```

Résultat :



- **auto** : C'est le navigateur qui décide de mettre ou non des barres de défilement (il n'en mettra que si c'est nécessaire).



## b) La propriété Word-wrap

Dans un texte de contenu, lorsqu'un mot sans espace ni trait d'union est plus large que la dimension définie pour son parent, le comportement normal consiste à faire déborder le texte au-delà de la largeur normale du cadre, sans retour à la ligne et sans que la dimension du conteneur ne soit affectée (sauf sur IE6).

C'est dans de telles situations qu'intervient la propriété Word-wrap, elle permet de gérer des mots très longs placés dans les blocs qui ne tiennent pas dans la largeur. Cette propriété permet de forcer la césure des très longs mots (généralement des adresses un peu longues).

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>

  <p> Les fichiers se trouvent à l'adresse H:\A.DIUMI\FORMATIONS\COURS\INFO\PRO
  GRAMMATION WEB\HTML ET CSS </p>

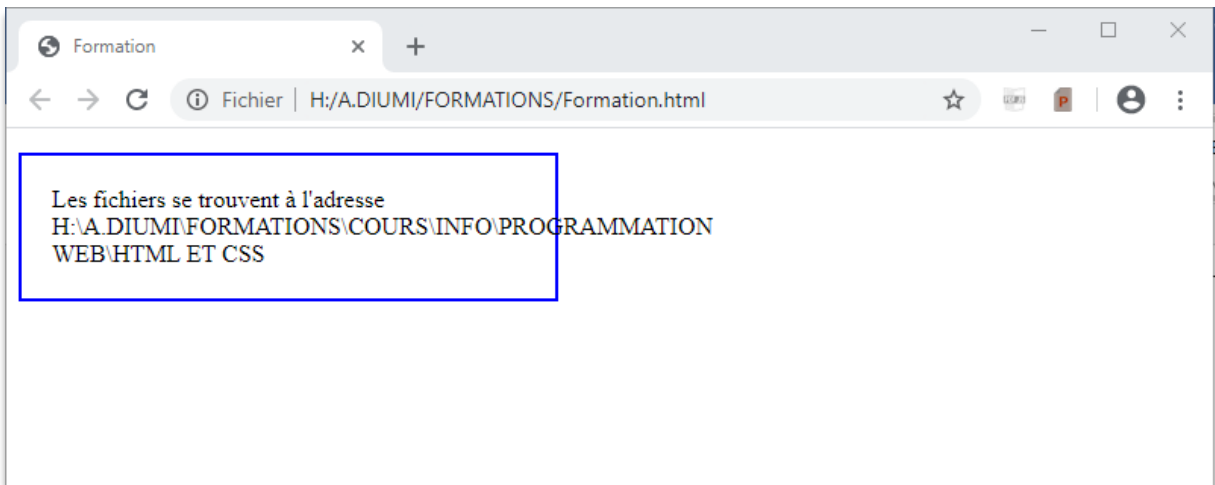
  </body>
</html>
```

Ajouter une bordure et des dimensions au paragraphe :

Forme.css

```
p
{
  width: 40%;
  border: 2px blue solid;
  padding: 20px;
}
```

Résultat :



Par défaut, le mot long n'est pas coupé et dépasse même le bloc, si on veut forcer la césure, il suffit d'écrire :

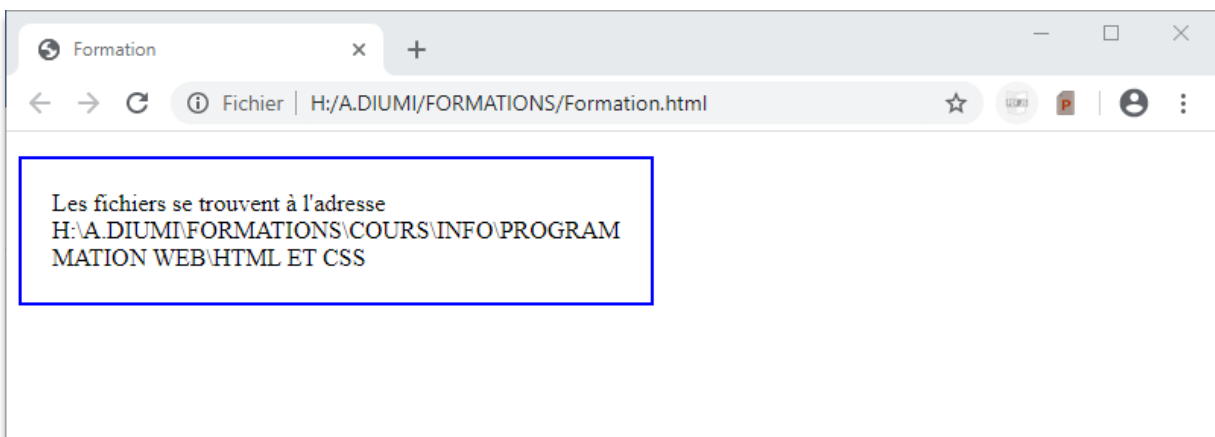
```
word-wrap: break-word;
```

Pour notre exemple :

Forme.css

```
p
{
width: 48%;
border: 2px blue solid;
padding: 20px;
word-wrap: break-word;
}
```

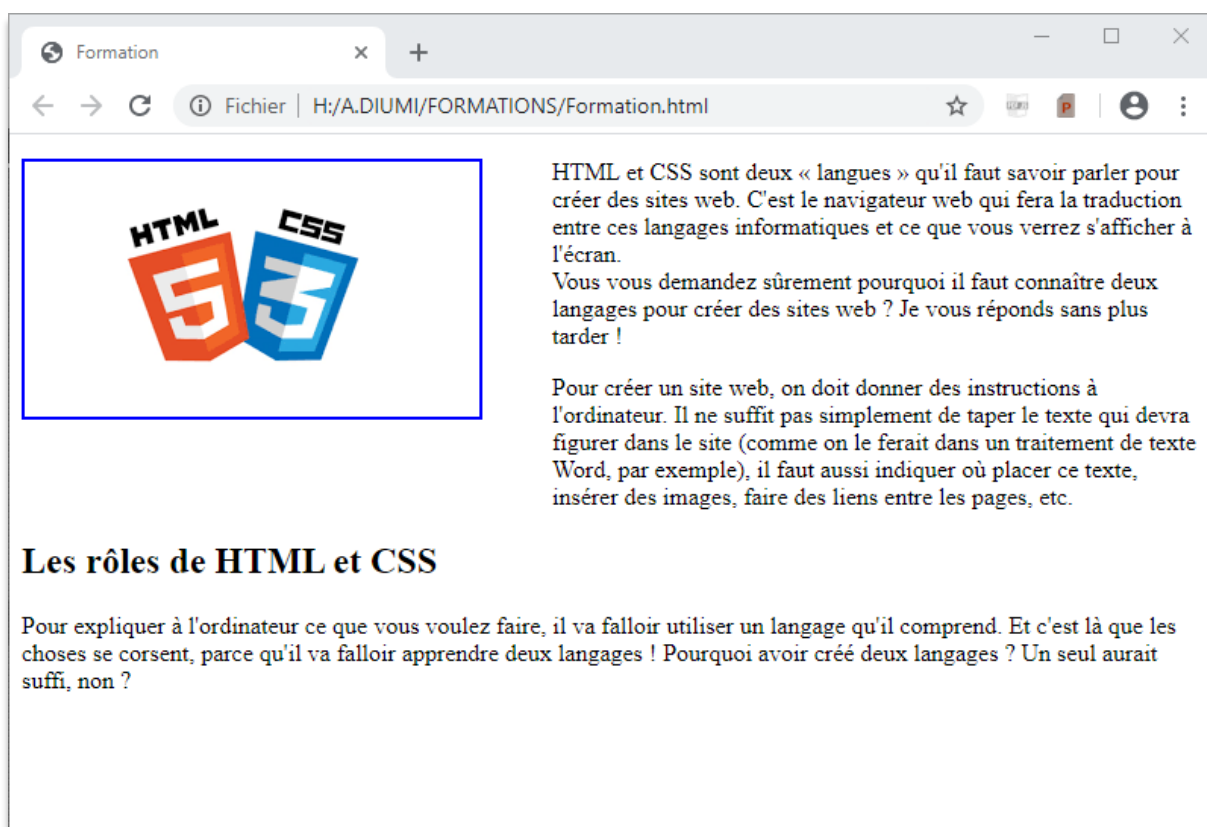
Résultat :



# III. POSITIONNEMENT ET TRANSFORMATION EN CSS

## III.1 LE POSITIONNEMENT FLOTTANT

Il est possible avec Ccss de flotter un élément autour du texte (faire un habillage). Voici à quoi ressemble un flottant :



Ici, j'ai flotté l'image autour du texte. La propriété qui nous permet de flotter un élément est **float** qui peut prendre une des deux valeurs :

- **left** : l'élément flottera à gauche.
- **right** : l'élément flottera à droite.

Il faut commencer par insérer le flottant (l'élément qui va flotter) et le reste de texte doit suivre cet élément dans html. Pour l'exemple précédent :

## Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
    
    <p >HTML et CSS sont deux « langues » qu'il faut savoir parler pour créer des
sites web. C'est le navigateur web qui fera la
traduction entre ces langages informatiques et ce que vous verrez s'afficher à
l'écran. <br/>
Vous vous demandez sûrement pourquoi il faut connaître deux langages pour crée
r des sites web ? Je vous réponds sans plus
tarder ! </p>

    <p >Pour créer un site web, on doit donner des instructions à l'ordinateur. Il
ne suffit pas simplement de taper le texte qui devra
figurer dans le site (comme on le ferait dans un traitement de texte Word, par
exemple), il faut aussi indiquer où placer ce texte,
insérer des images, faire des liens entre les pages, etc.</p>
    <h2>Les rôles de HTML et CSS </h2>
    <p >Pour expliquer à l'ordinateur ce que vous voulez faire, il va falloir util
iser un langage qu'il comprend. Et c'est là que les choses se
corsent, parce qu'il va falloir apprendre deux langages !
Pourquoi avoir créé deux langages ? Un seul aurait suffi, non ? <br/>

  </body>
</html>
```

J'ai besoin de flotter l'image à gauche, il suffit d'appliquer float : left au sélecteur de l'image dans Css comme ceci :

```
.image
{
  float: left;
}
```

Résultat :



Et si on flottait l'image à droite ?

Forme.css

```
.image
{
float: right;
}
```

Résultat :



Si l'on souhaite pour une raison quelconque que le deuxième paragraphe (Pour créer un site web, on doit...) se place totalement en dessous de l'image (le flottant), dans ce cas on doit stopper le flottant.

C'est la propriété **clear** qui permet de stopper un flottant, elle peut prendre une des valeurs suivantes :

- **left** : le texte se poursuit en-dessous après un float: left;
- **right** : le texte se poursuit en-dessous après un float: right;
- **both** : le texte se poursuit en-dessous, que ce soit après un float: left; ou après un float: right;. C'est la valeur que je vous recommande d'utiliser pour stopper un flottant.

Pour notre exemple, j'attribue un nom particulier au deuxième paragraphe avec l'attribut class :

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
    
    <p >HTML et CSS sont deux « langues » qu'il faut savoir parler pour créer des sites web. C'est le navigateur web qui fera la traduction entre ces langages informatiques et ce que vous verrez s'afficher à l'écran. <br/>
```

```
Vous vous demandez sûrement pourquoi il faut connaître deux langages pour créer des sites web ? Je vous réponds sans plus tarder ! </p>
```

```
<p class="para">Pour créer un site web, on doit donner des instructions à l'ordinateur. Il ne suffit pas simplement de taper le texte qui devra figurer dans le site (comme on le ferait dans un traitement de texte Word, par exemple), il faut aussi indiquer où placer ce texte, insérer des images, faire des liens entre les pages, etc.</p>
```

```
<h2>Les rôles de HTML et CSS </h2>
```

```
<p >Pour expliquer à l'ordinateur ce que vous voulez faire, il va falloir utiliser un langage qu'il comprend. Et c'est là que les choses se corsent, parce qu'il va falloir apprendre deux langages ! Pourquoi avoir créé deux langages ? Un seul aurait suffi, non ? <br/>
```

```
</body>
```

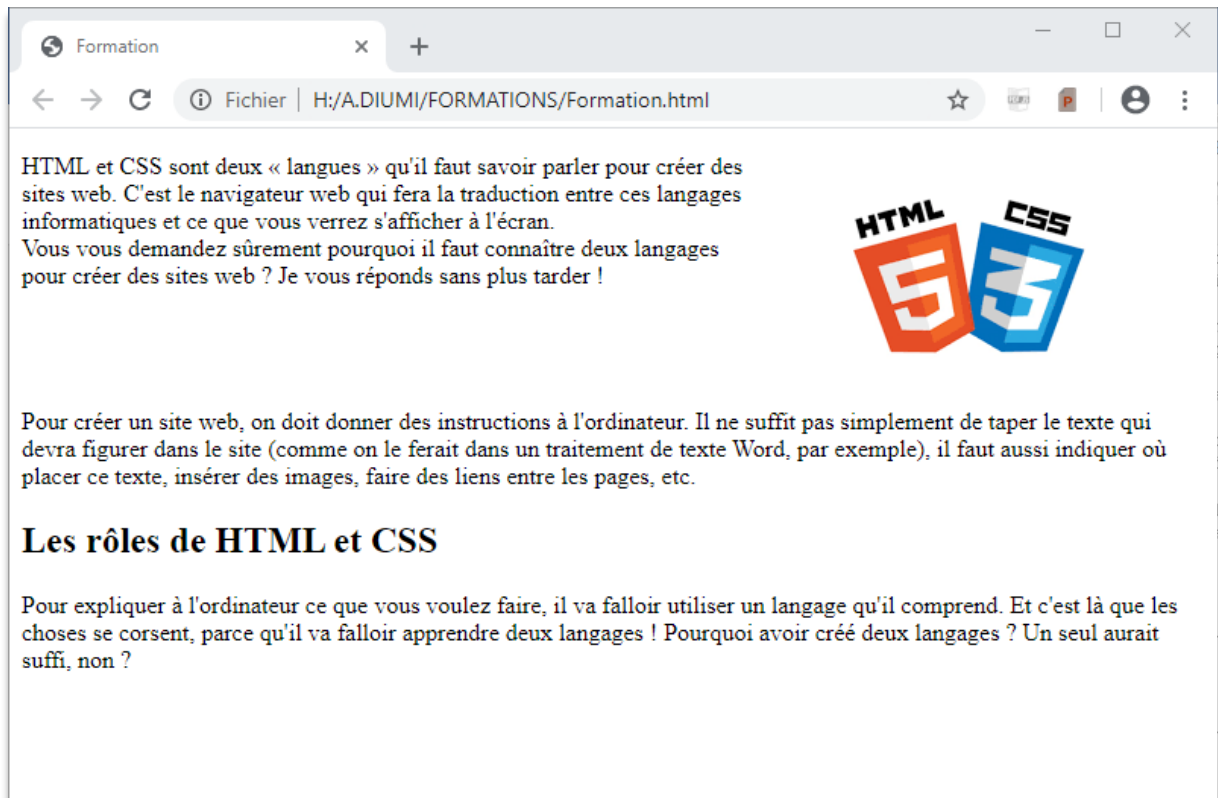
```
</html>
```

Forme.css

```
.image
{
  float: right;
}

.para
{
  clear: both;
}
```

Résultat :



Le deuxième paragraphe est entièrement en dessous du flottant.

Sachez qu'il est possible de mettre deux flottants, l'un à gauche, l'autre à droite et le texte au milieu comme le montre l'exemple suivant :

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>

  
  
  <p >HTML et CSS sont deux « langues » qu'il faut savoir parler pour créer des
sites web. C'est le navigateur web qui fera la
traduction entre ces langages informatiques et ce que vous verrez s'afficher à
l'écran. <br/>
Vous vous demandez sûrement pourquoi il faut connaître deux langages pour crée
r des sites web ? Je vous réponds sans plus tarder ! </p>
```



```
<p >Pour créer un site web, on doit donner des instructions à l'ordinateur. Il ne suffit pas simplement de taper le texte qui devra figurer dans le site (comme on le ferait dans un traitement de texte Word, par exemple), il faut aussi indiquer où placer ce texte, insérer des images, faire des liens entre les pages, etc.</p>
<h2>Les rôles de HTML et CSS </h2>
<p>Pour expliquer à l'ordinateur ce que vous voulez faire, il va falloir utiliser un langage qu'il comprend. Et c'est là que les choses se corsent, parce qu'il va falloir apprendre deux langages ! Pourquoi avoir créé deux langages ? Un seul aurait suffi, non ? </p>
</body>
</html>
```

### Forme.css

```
.image1
{
    float: left;
}

.image2
{
    float: right;
}
```

Résultat :



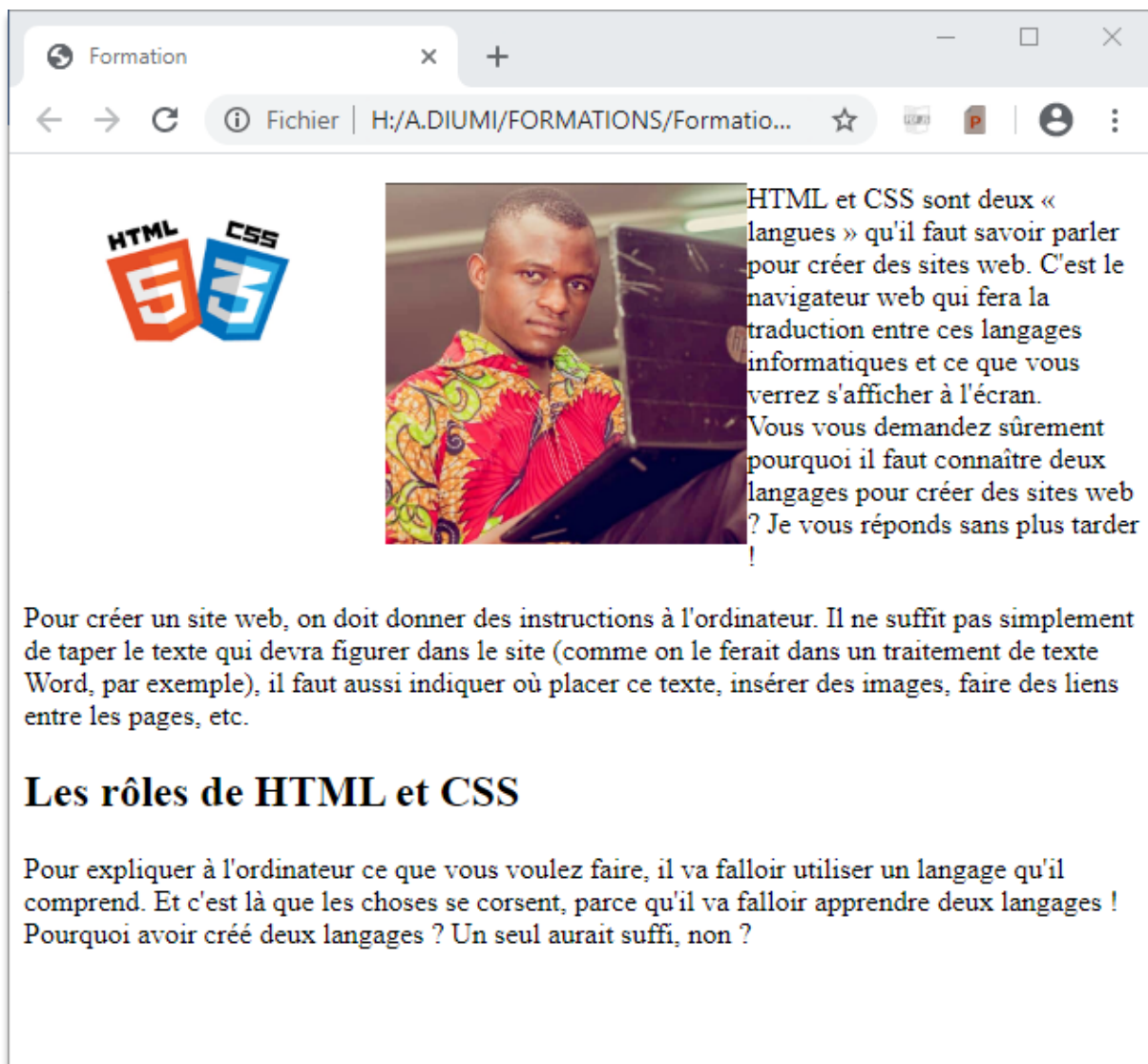
On peut aussi mettre plusieurs flottants les uns à la suite des autres. Reprenons le même exemple html, changeons seulement le fichier Ccss :

Forme.css

```
.image1
{
  float: left;
}

.image2
{
  float: left;
}
```

Résultat :



Il suffit d'ajouter une marge extérieure au bloc contenant le texte pour qu'il ne soit pas collé au deuxième flottant.

Certains développeurs utilisent cette technique pour la mise en page d'un site web. Prenons l'exemple suivant :

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
    <section class="bloc1">
      <h1> LES BASES DU LANGAGE ET MODE CONSOLE </h1>
      <p>Wikipédia présente Visual basic comme un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, créé par Microsoft pour son modèle COM. <br/>Visual basic est directement
```

dérivé du BASIC (acronyme pour Beginner's All purpose Symbolic Instruction Code) et permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôle ou objets Active X. </p>

<p>Pour coder dans un langage de programmation, il faut des outils adaptés : un éditeur de texte, un compilateur ainsi qu'un débogueur.<br/> Il y a des programmes qui combinent les trois outils et ces programmes sont appelés Environnement de développement intégré (EDI) ou IDE en anglais, pour Integrated development Environment. </p>

</section>

<section class="bloc2">

<h2>WISSEN CORPORATION </h2>

<p>C'est un groupe formé essentiellement de meilleurs étudiants de nos groupes d'étude et d'encadrement qui se sont fixés plusieurs objectifs dont le premier est celui du partage de la connaissance en Informatique et en mathématique. </p>

<p>Il y a une naissance en toute connaissance, Thucydide a dit : « Avoir des connaissances sans les partager, c'est se mettre au niveau de celui qui n'a pas d'idées ». Wissen Corporation est là pour assurer votre formation, votre encadrement suivant vos désirs. </p>

</section>

</body>

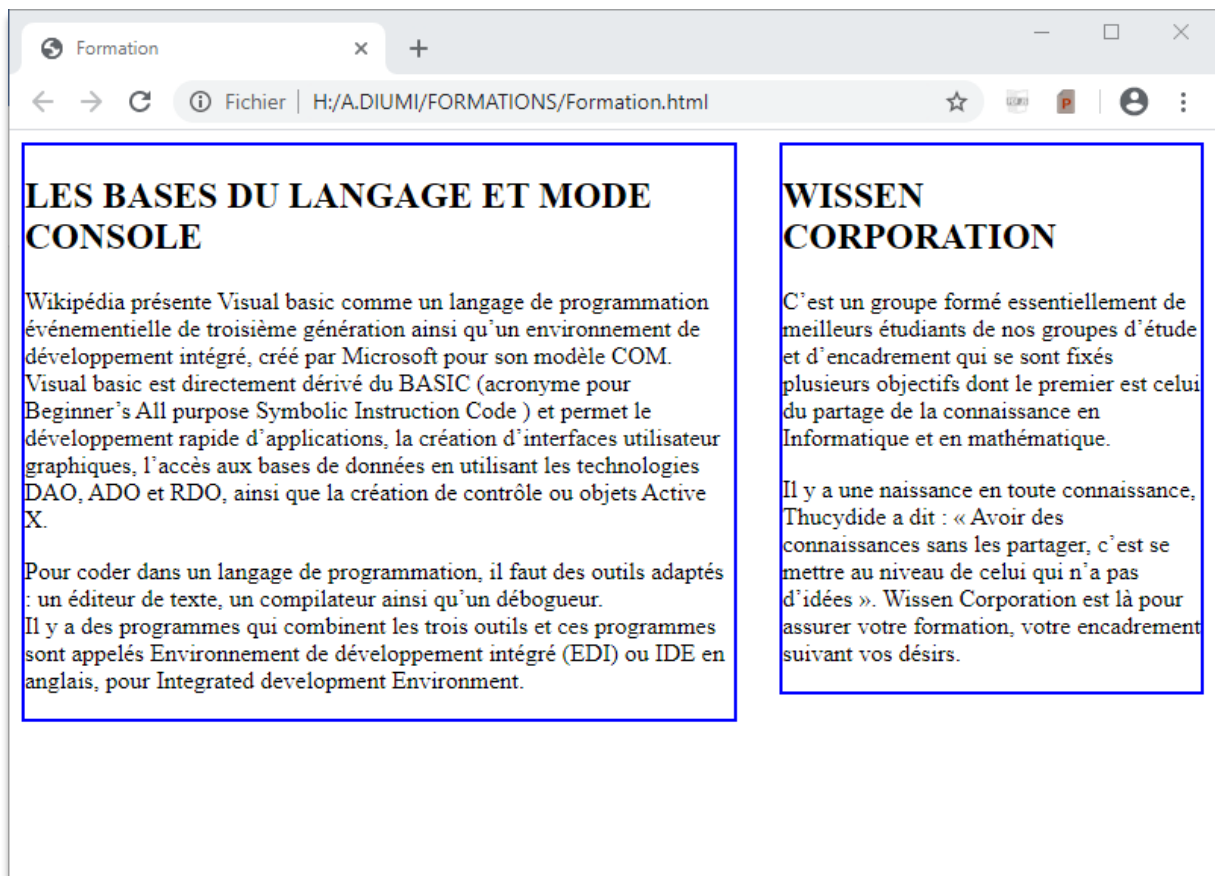
</html>

J'aimerais que les deux sections (bloc1 et bloc2) soient côte à côte, il suffit de flotter bloc1 comme ceci :

```
.bloc1
{
  float: left;
  border: 2px blue solid;
  width: 60%;
}
.bloc2
{
  border: 2px blue solid;
  margin-left:500px;
}
```

J'ai ajouté des bordures aux deux sections et une marge extérieure à gauche de la deuxième section.

Résultat :



Et voilà : les deux sections sont côte à côte.

### III.2 POSITIONNEMENT AVEC LA PROPRIETE *Display*

Un peu plus haut, nous avons vu que la plupart des balises en html sont souvent dans l'une des catégories suivantes :

- inline : une balise de type inline se trouve à l'intérieur d'une balise block. Une balise inline ne crée pas de retour à la ligne, le texte qui se trouve à l'intérieur s'écrit donc à la suite du texte précédent, sur la même ligne :
  - La largeur et la hauteur dépendent du contenu, c'est-à-dire qu'une telle balise n'occupe que l'espace nécessaire ;
  - La hauteur, la largeur et les marges ne sont pas modifiables ;
- block : une balise de type block sur votre page web crée automatiquement un retour à la ligne avant et après :
  - Elle occupe toute la largeur disponible ;
  - La hauteur dépend du contenu ;
  - La largeur, hauteur et les marges sont modifiables.

Votre page web sera en fait constituée d'une série de blocs les uns à la suite des autres. Une balise de type block occupe toute la largeur disponible.

Avec la propriété **Display**, il est possible de transformer n'importe quel élément de votre page d'un type vers un autre.

Cette propriété peut prendre une des valeurs suivantes :

- **Inline** : les éléments d'une ligne. Les éléments de type inline se placent les uns à côté des autres.
- **Block** : Eléments en forme de blocs. Se placent les uns en-dessous des autres et peuvent être redimensionnés.
- **Inline-block** : Eléments positionnés les uns à côté des autres (comme les inlines) mais qui peuvent être redimensionnés (comme les blocs).
- **None** : Eléments non affichés. Par exemple head

Essayons de voir chaque valeur avec un exemple concret

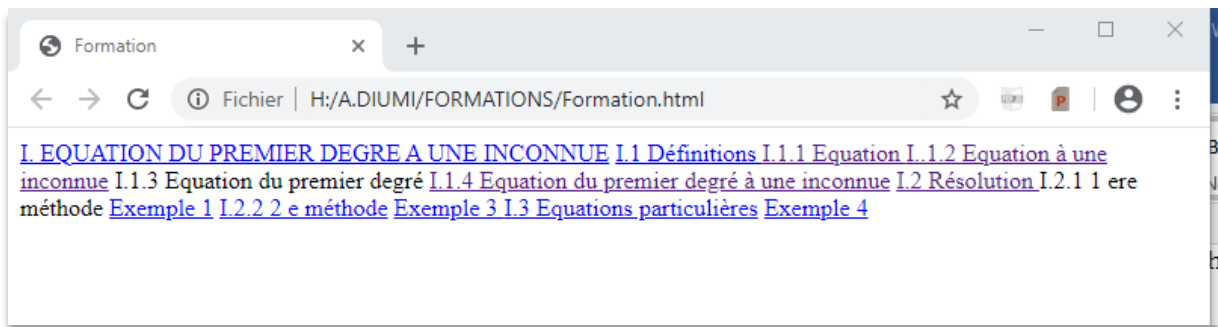
### Exemple1

Les liens vers les principaux points d'un cours de mathématiques

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
    <section class="bloc1">
<a href="#LI" class="titre1">I. EQUATION DU PREMIER DEGRE A UNE INCONNUE</a>
<a href="#LI1" >I.1 Définitions </a>
<a href href="#LI11">I.1.1 Equation </a>
<a href href="#LI112">I..1.2 Equation à une inconnue</a>
<a hrefhref="#LI13" >I.1.3 Equation du premier degré</a>
<a href href="#LI14">I.1.4 Equation du premier degré à une inconnue</a>
<a href href="#LI2">I.2 Résolution </a>
<a hrefhref="#LI21" >I.2.1 1ere méthode</a>
<a href="#Ex1" >Exemple 1</a>
<a href="#LI22" >I.2.2 2e méthode</a>
<a href="#Ex3" >Exemple 3 </a>
<a href="#LI3" >I.3 Equations particulières</a>
<a href="#Ex4" >Exemple 4 </a>
</body>
</html>
```

Les liens s'affichent les uns à côté des autres :

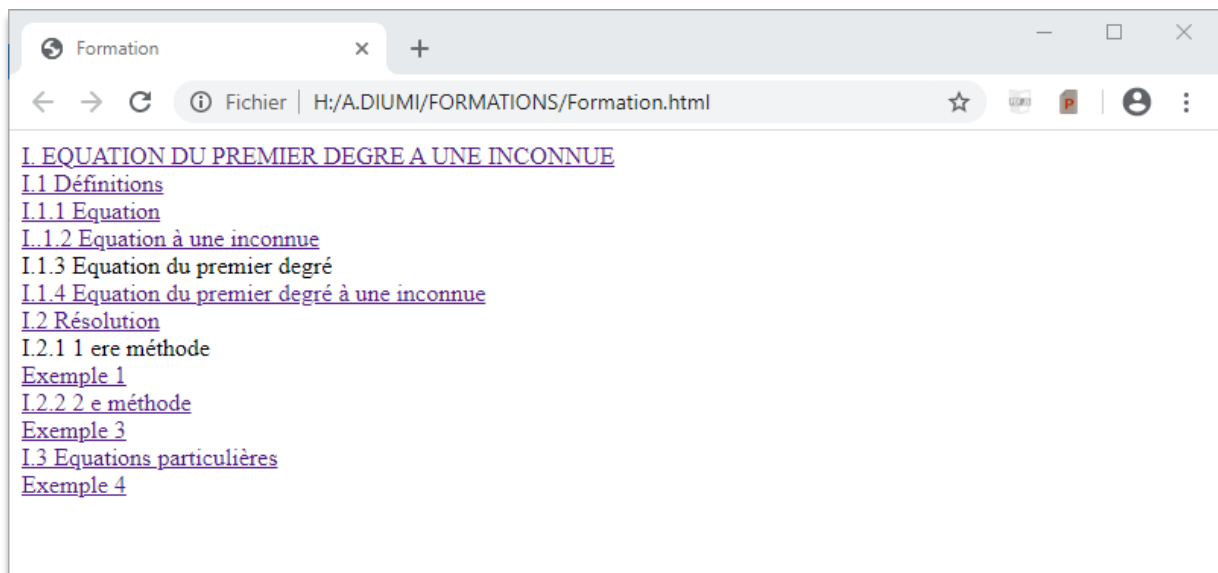


Pour faire en sorte qu'ils (les liens) se placent les uns en dessous des autres, on peut utiliser `<br/>`, mais avec cette balise, les liens seront toujours de type inline et il n'y aura pas moyen de les redimensionner. La meilleure solution est de les transformer en type block : avec cette transformation, ils vont se placer les uns en dessous des autres et nous auront la possibilité de les redimensionnées.

Forme.css

```
a
{
  display:block;
}
```

Résultat :



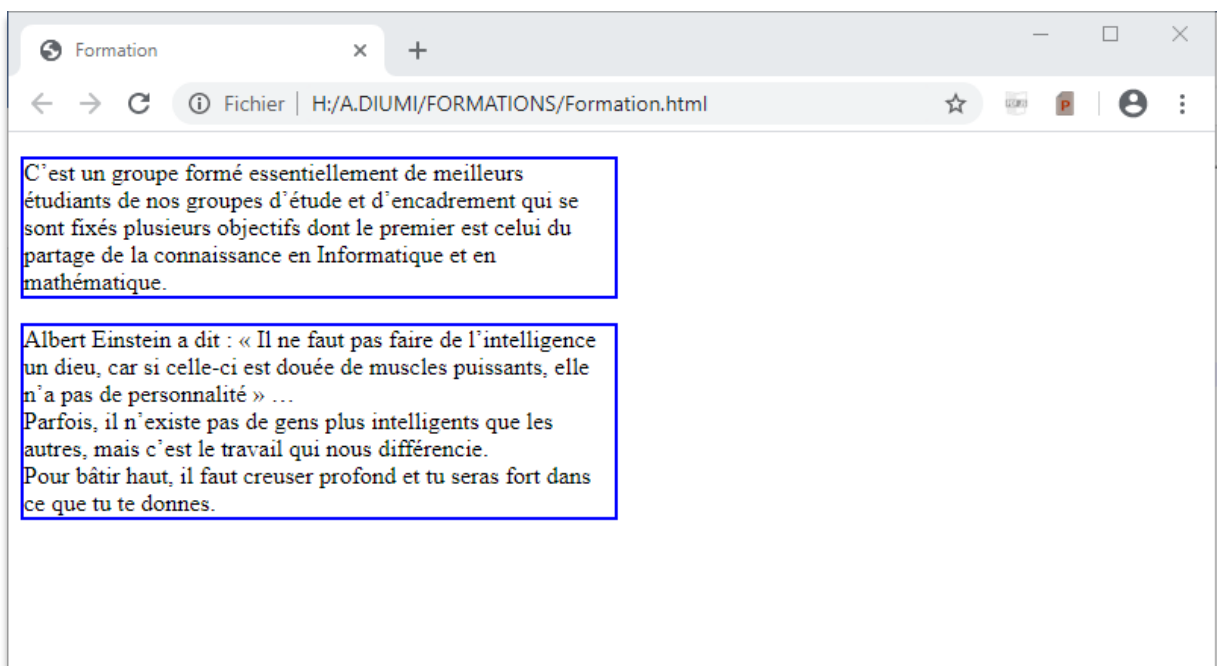
Essayez d'ajouter des bordures, vous verrez qu'elles occuperont toute la largeur car les liens sont maintenant de type block.

## Exemple 2

### Deux paragraphes

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
<p class="para1">C'est un groupe formé essentiellement de meilleurs étudiants
de nos groupes d'étude et d'encadrement qui se sont fixés plusieurs objectifs
dont le premier est celui du partage de la connaissance en Informatique et en
mathématique. </p>
  <p class="para2">
Albert Einstein a dit : « Il ne faut pas faire de l'intelligence un dieu,
car si celle-ci est douée de muscles puissants, elle n'a pas de personnalité »
...<br/>
Parfois, il n'existe pas de gens plus intelligents que les autres, mais c'est
le travail qui nous différencie. <br/>
Pour bâtir haut, il faut creuser profond et tu seras fort dans ce que tu te do
nnes. </p>
</body>
</html>
```

Comme vous le savez, les deux paragraphes vont se placer l'un en dessous de l'autre même si on modifie leurs largeurs :





Pour qu'ils se placent l'un à la suite de l'autre, il suffit de transformer leur type en inline.

### Exemple 3

Table de matières sous forme des liens

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>

<a href="#LI" >I. EQUATION DU PREMIER DEGRE A UNE INCONNUE</a> <br/>

<section class="sous">
<a href="#LI1" >I.1 Définitions </a><br/>
<a href="#LI11">I.1.1 Equation </a><br/>
<a href="#LI112">I..1.2 Equation à une inconnue</a><br/>
<a href="#LI13" >I.1.3 Equation du premier degré</a> <br/>
<a href="#LI14">I.1.4 Equation du premier degré à une inconnue</a> <br/>
<a href="#LI2">I.2 Résolution </a><br/>
<a href="#LI21" >I.2.1 1 ere méthode</a> <br/>
<a href="#Ex1" >Exemple 1</a><br/>
<a href="#LI22" >I.2.2 2 e méthode</a> <br/>
<a href="#Ex3" >Exemple 3 </a> <br/>
<a href="#LI3" >I.3 Equations particulières</a> <br/>
<a href="#Ex4" >Exemple 4 </a><br/>
</section>

<a href = "#LII">II. EQUATIONS REDUCTIBLES AU PREMIER DEGRE </a><br/>
<section class="sous">
<a href="#LII1" >II.1 EQUATIONS PRODUITS  $A . B . C \dots = 0$ </a> <br/>
<a href="#Ex5" >Exemple 5 </a> <br/>
<a href="#Ex6" >Exemple 6 </a> <br/>
<a href="#Ex7" >Exemple 7 </a><br/>
<a href = "#LII2">II.2. EQUATIONS FRACTIONNAIRES</a> <br/>
<a href="#Ex8" >Exemple 8 </a><br/>
<a href="#Ex9" >Exemple 9 </a><br/>
<a href="#LII3" >II.3 EQUATIONS CONTENANT DES VALEURS ABSOLUES </a><br/>
<a href="#Ex10" >Exemple 10 </a><br/>
<a href="#Ex11" >Exemple 11</a> <br/>
</section>
<a href="#LIII" >III. PROBLEMES DONT LA RESOLUTION CONDUIT A UNE EQUATION DU P
REMIER DEGRE A UNE INCONNUE</a><br/>
```

```

<section class="sous">
<a href="#Ex12" >Exemple 12 </a> <br/>
<a href="#Ex13" >Exemple 13</a> <br/>
</section>
<a href="#LIV">IV. EXERCICES RESOLUS</a> <br/>
</body>
</html>

```

J'aimerais afficher au départ seulement les quatre principaux points et cacher les sous points :

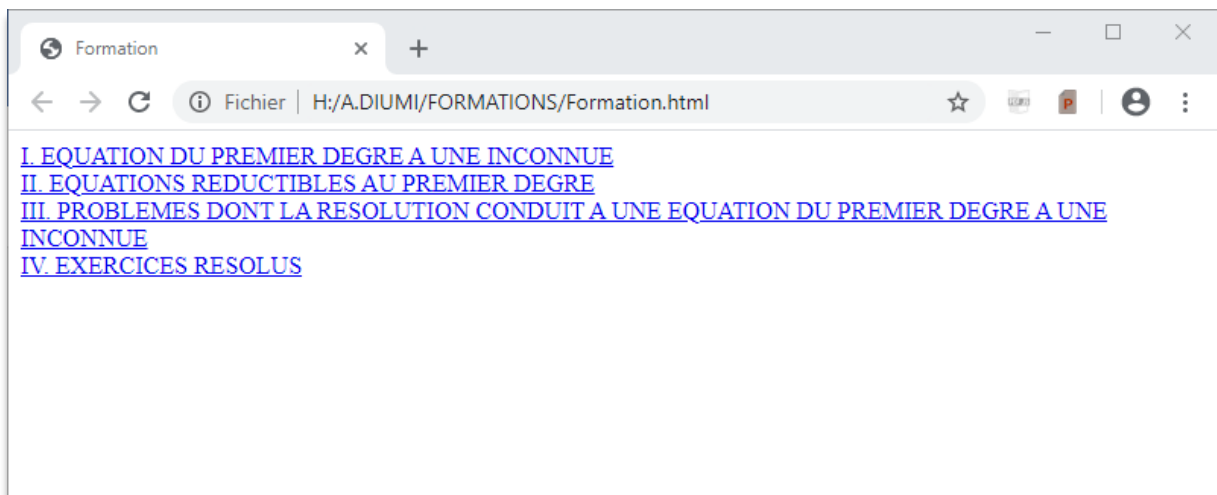
Forme.css

```

.sous
{
    display: none;
}

```

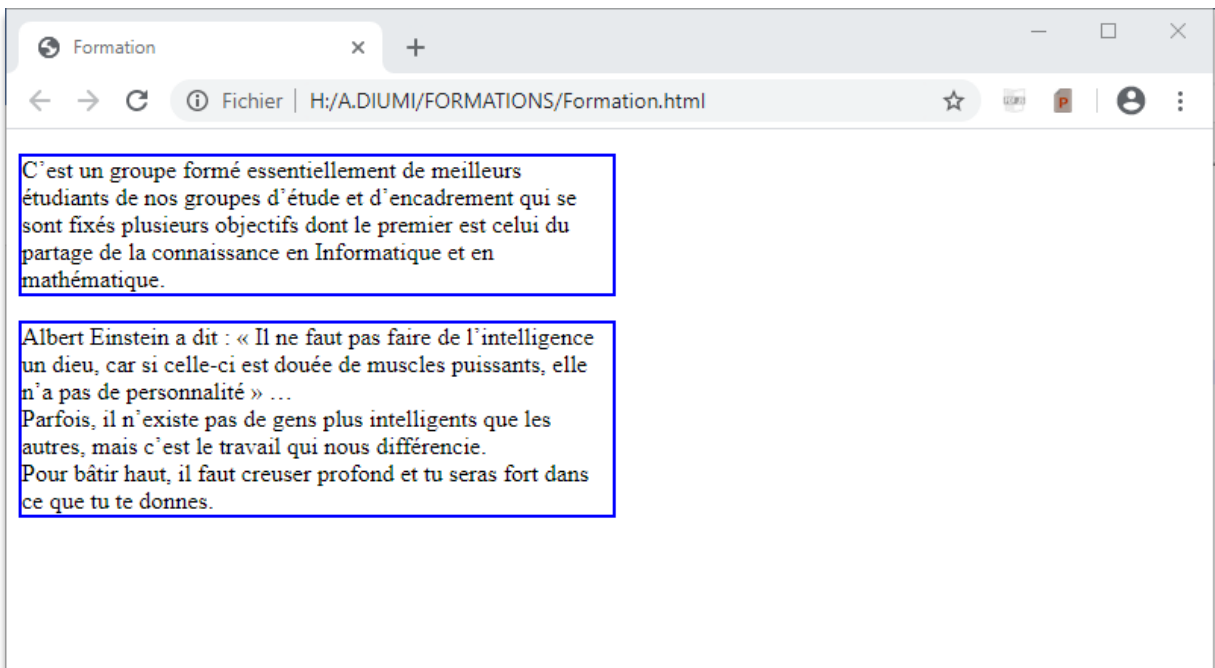
Résultat :



Pour faire apparaître ces éléments (les sous points pour notre exemple) par la suite, on devra faire appel à JavaScript.

## Exemple 4

Même fichier html de l'exemple 2, on avait obtenu le résultat suivant :

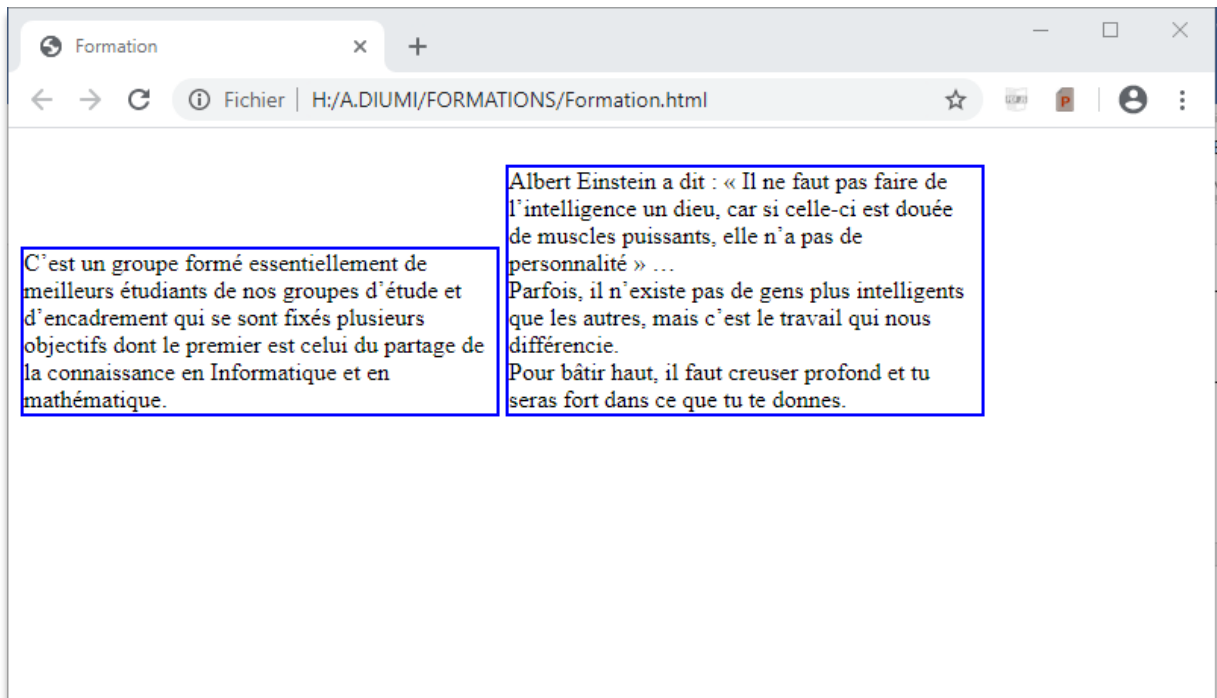


Pour que les deux blocs se positionnent l'un à côté de l'autre et qu'on ait la possibilité de les redimensionner (leur donner des tailles précises), on doit les transformer en type inline-block.

Forme.css

```
.para1
{
  display: inline-block;
  border: 2px blue solid;
  width: 40%;
}
.para2
{
  display: inline-block;
  border: 2px blue solid;
  width: 40%;
}
```

Résultat :



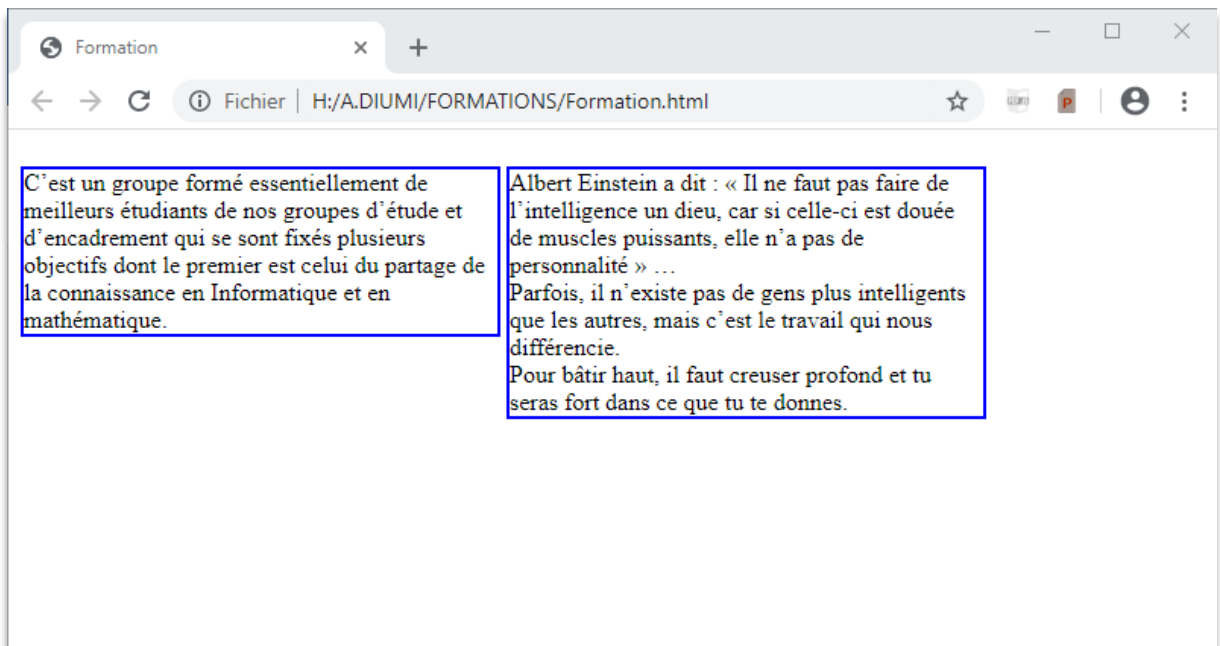
Vous avez sans doute remarqué que les deux éléments se sont alignés sur une même ligne de base (appelée *baseline*). Lorsque les éléments sont transformés en `inline-block`, nous avons la possibilité de modifier leur alignement vertical grâce à la propriété ***vertical-align***. Voici quelques-unes des valeurs possibles pour cette propriété :

- ***baseline*** : aligne de la base de l'élément avec celle de l'élément parent (par défaut). Voir exemple ci-dessus.
- ***top*** : aligne en haut ;

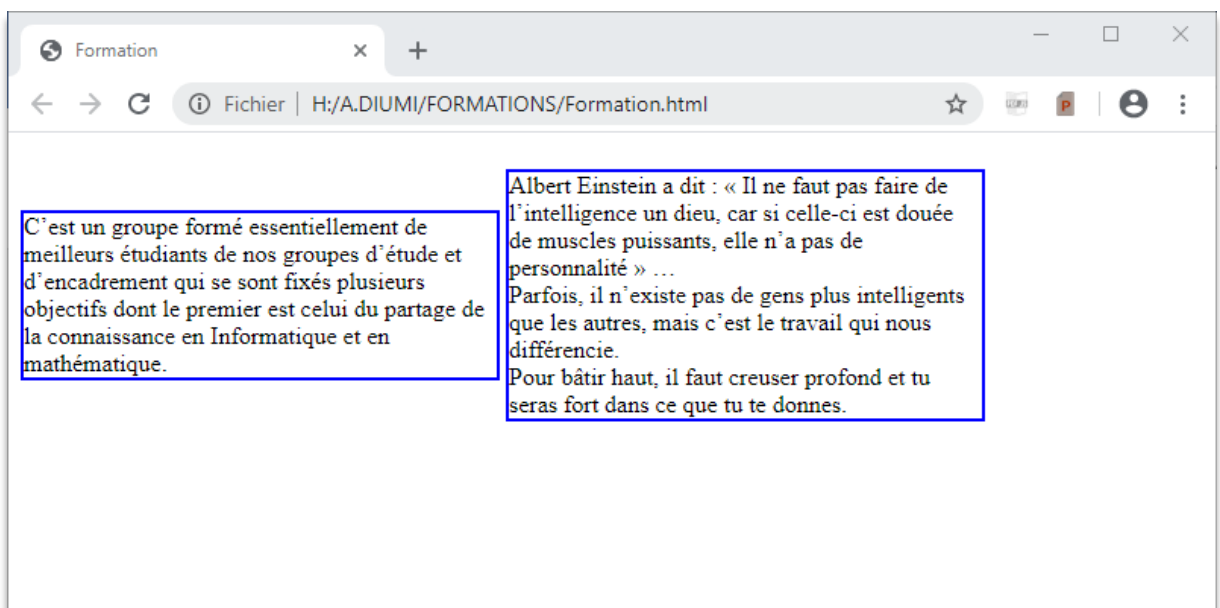
forme.css

```
.para1
{
  display: inline-block;
  vertical-align: top;
  border: 2px blue solid;
  width: 40%;
}
.para2
{
  display: inline-block;
  vertical-align: top;
  border: 2px blue solid;
  width: 40%;
}
```

Résultat :



- **middle** : centre verticalement ;



- **bottom** : aligne en bas ;
- **(valeur en px ou %)** : aligne à une certaine distance de la ligne de base (baseline).

## III.3 POSITIONNEMENT AVEC FLEXBOX

### III.3.1 Généralités

J'ai préféré vous parler de cette valeur possible de la puissante propriété display à part car elle est un peu particulière selon moi. Par défaut, quand un bloc contient de sous blocs (une section qui contient de sous sections, ul qui contient de li), les sous-blocs se positionnent les uns en dessous des autres comme le montre l'exemple suivant :

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel="stylesheet" href="forme.css" />
  </head>
  <body>

<section class="bloc">

<section class="sousBloc1">
  <h1> VISUAL BASIC </h1>
<h3> LES BASES DU LANGAGE ET MODE CONSOLE </h3>
<p>Wikipédia présente Visual basic comme un langage de
programmation événementielle de troisième génération ainsi qu'un
environnement de développement intégré, créé par Microsoft pour son
modèle COM. <br/>Visual basic est directement dérivé du BASIC (acronyme
pour Beginner's All purpose Symbolic Instruction Code ) et permet le
développement rapide d'applications, la création d'interfaces utilisateur
graphiques, l'accès aux bases de données en utilisant les technologies DAO,
ADO et RDO, ainsi que la création de contrôle ou objets Active X. </p>

<p>Pour coder dans un langage de programmation, il faut des outils
adaptés : un éditeur de texte, un compilateur ainsi qu'un débogueur.<br/> Il y
a
des programmes qui combinent les trois outils et ces programmes sont
appelés Environnement de développement intégré (EDI) ou IDE en anglais,
pour Integrated development Environment. </p>

<p>L'éditeur de texte nous permet d'écrire le code source du programme.
Le compilateur (Interpréteur) pour transformer le code source en binaire qui
est le seul langage compréhensible par la machine.<br/>
Le débogueur permet de traquer les erreurs dans le programme.
```

```

Pour Visual basic, nous pouvons utiliser : Visual studio, Visual studio
Express, SharpDevelop, MonoDevelop etc. </p>
</section>

<section class="sousBloc2">
<h2>WISSEN CORPORATION </h2>

<p>C'est un groupe formé essentiellement de meilleurs étudiants de nos
groupes d'étude et d'encadrement qui se sont fixés plusieurs objectifs dont
le premier est celui du partage de la connaissance en Informatique et en
mathématique. </p>
<p>
Albert Einstein a dit : « Il ne faut pas faire de l'intelligence un dieu,
car si celle-
ci est douée de muscles puissants, elle n'a pas de personnalité » ...<br/>
Parfois, il n'existe pas de gens plus intelligents que les autres, mais c'est
le travail qui nous différencie.<br/>
Pour bâtir haut, il faut creuser profond et tu seras fort dans ce que tu te do
nnes. </p>

<p>Il y a une naissance en toute connaissance, Thucydide a dit : « Avoir
des connaissances sans les partager, c'est se mettre au niveau de celui qui
n'a pas d'idées ». Wissen Corporation est là pour assurer votre formation,
votre encadrement suivant vos désirs. </p>
</section>
</section>
</body>
</html>

```

Ajoutons une bordure à chaque sous-section et à la section principale :

Forme.css

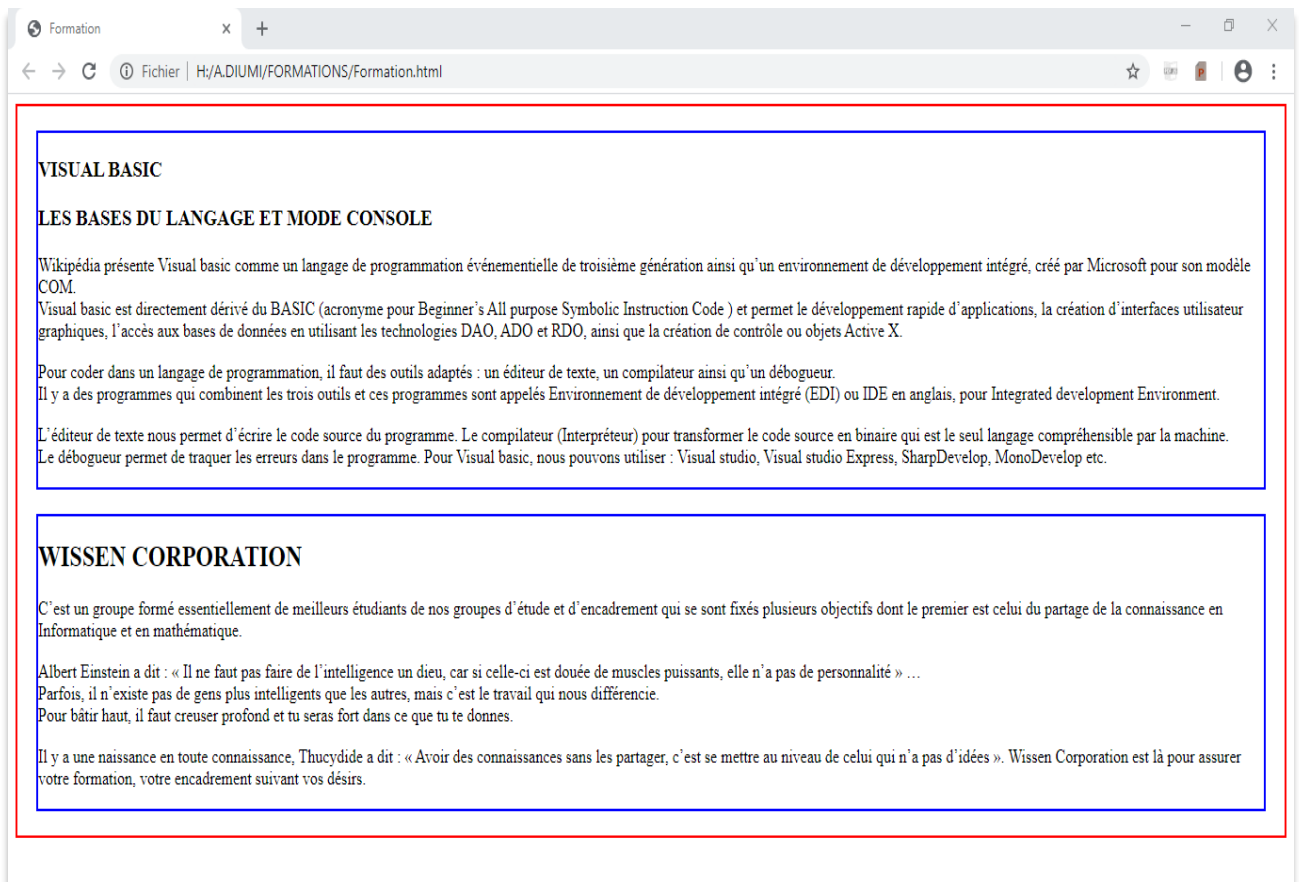
```

.bloc
{
border: 2px red solid;
}

.sousBloc1, .sousBloc2
{
border: 2px blue solid;
margin: 20px;
}

```

## Résultat :



Si nous appliquons **display : flex** au bloc parent, les sous-bloc (les enfants) seront repartis en colonnes. Comme dans notre exemple, les deux sous-blocs seront repartis en colonne, c'est-à-dire seront positionnés l'un à côté de l'autre.

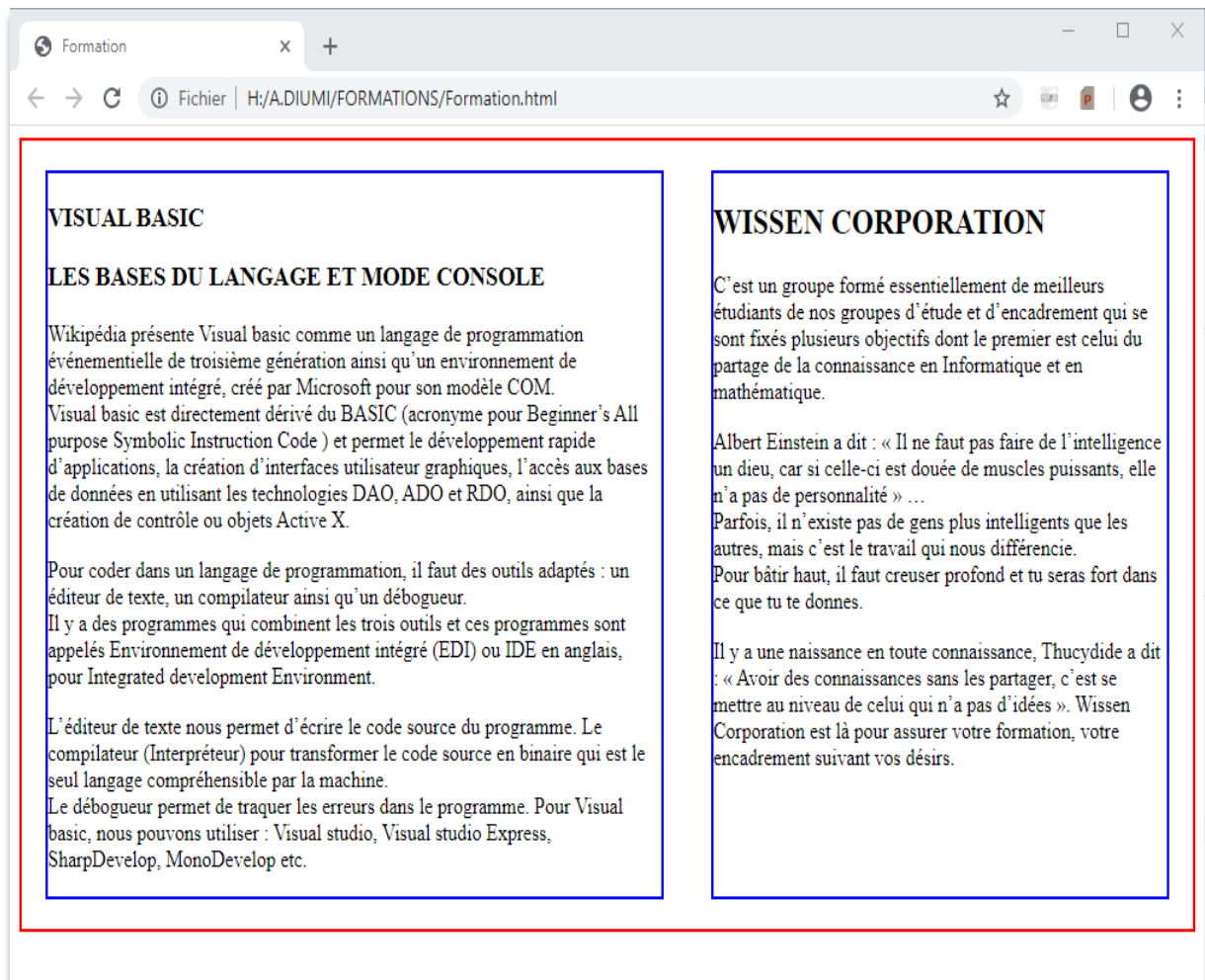
Forme.css

```
.bloc
{
display: flex;
border: 2px red solid;
}

.sousBloc1, .sousBloc2
{
border: 2px blue solid;
margin: 20px;
}
```



Résultat :



Notez que les blocs enfants n'occupent que la hauteur et la largeur nécessaire au remplissage : les colonnes ne remplissent pas forcément toute la largeur.

Pour chaque sous-bloc, la largeur peut être gérée par la propriété **flex** :

```
.sousBloc
{
    flex: nombre;
}
```

Le nombre donne le rapport de largeur (ou de hauteur) entre les sous-blocs. En mettant 1 à tous les sous-blocs, ils seront tous de la même taille.

Par exemple, si je mets **flex : 4** pour le premier sous-bloc et **flex : 1** pour second :

## Forme.css

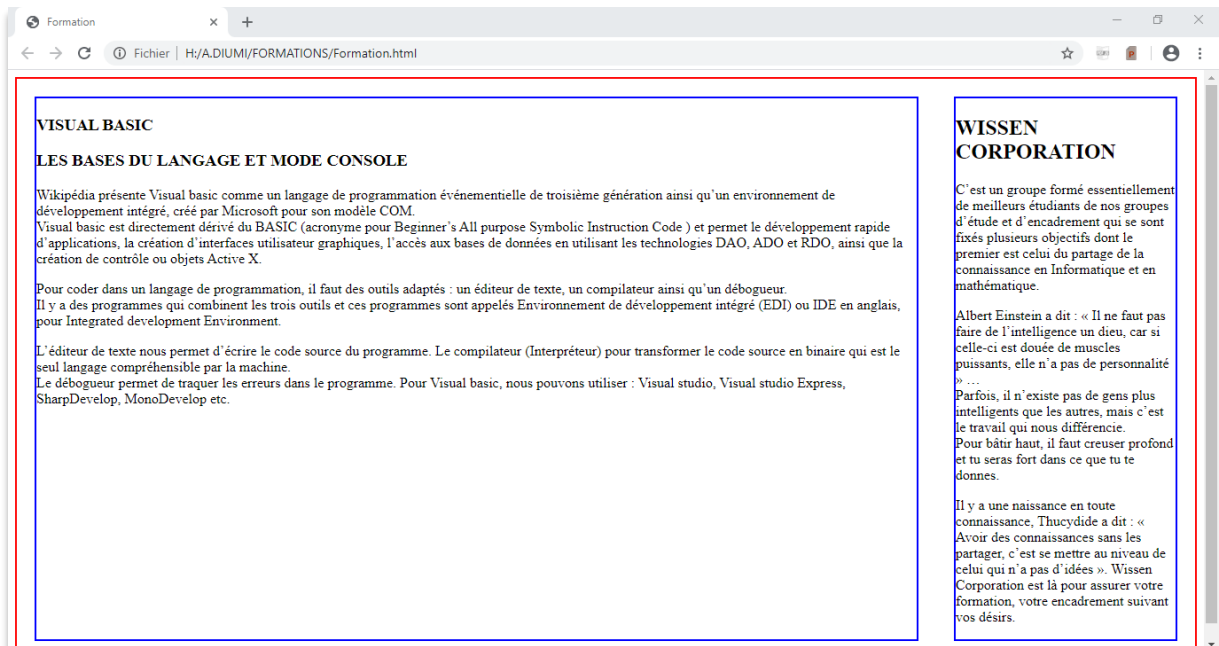
```
.bloc
{
display: flex;
border: 2px red solid;
}

.sousBloc1, .sousBloc2
{
border: 2px blue solid;
margin: 20px;
}

.sousBloc1
{
flex: 4;
}

.sousBloc2
{
flex: 1;
}
```

## Résultat :



En appliquant **flex :4** et **flex :1** respectivement au premier et deuxième sous-bloc, c'est comme si nous avons divisé la largeur disponible en 5 parties (4+1) et avons attribué les  $\frac{4}{5}$  au premier sous-bloc et le reste ( $\frac{1}{5}$ ) au deuxième sous-bloc.

Sachez qu'il est également possible de contrôler la largeur et la hauteur de chaque sous-bloc avec les propriétés **width** et **height**.

### III.3.2 Propriétés liées a display : flex

Dans cette partie, nous allons voir quelques propriétés liées à display :flex.

Nous allons utiliser l'exemple suivant :

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>

<section class="bloc">
  <div class="titre">
    <h1> VISUAL BASIC </h1>
    <h3> LES BASES DU LANGAGE ET MODE CONSOLE </h3>
  </div>
  <p class="para1">Wikipédia présente Visual basic comme un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, créé par Microsoft pour son modèle COM. <br/>Visual basic est directement dérivé du BASIC (acronyme pour Beginner's All purpose Symbolic Instruction Code ) et permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôle ou objets Active X. </p>

  <p class="para2">Pour coder dans un langage de programmation, il faut des outils adaptés : un éditeur de texte, un compilateur ainsi qu'un débogueur.<br/> Il y a des programmes qui combinent les trois outils et ces programmes sont appelés Environnement de développement intégré (EDI) ou IDE en anglais, pour Integrated development Environment. </p>
```

```
<p class="para3">L'éditeur de texte nous permet d'écrire le code source du programme.  
Le compilateur (Interpréteur) pour transformer le code source en binaire qui est le seul langage compréhensible par la machine.<br/>  
Le débogueur permet de traquer les erreurs dans le programme.  
Pour Visual basic, nous pouvons utiliser : Visual studio, Visual studio Express, SharpDevelop, MonoDevelop etc. </p>  
</section>  
</body>  
</html>
```

### a) La propriété flex-direction

Elle permet de gérer l'alignement des sous-blocs, les valeurs possibles sont :

- **row** (valeur par défaut) :aligne les sous blocs verticalement de la gauche vers la droite.

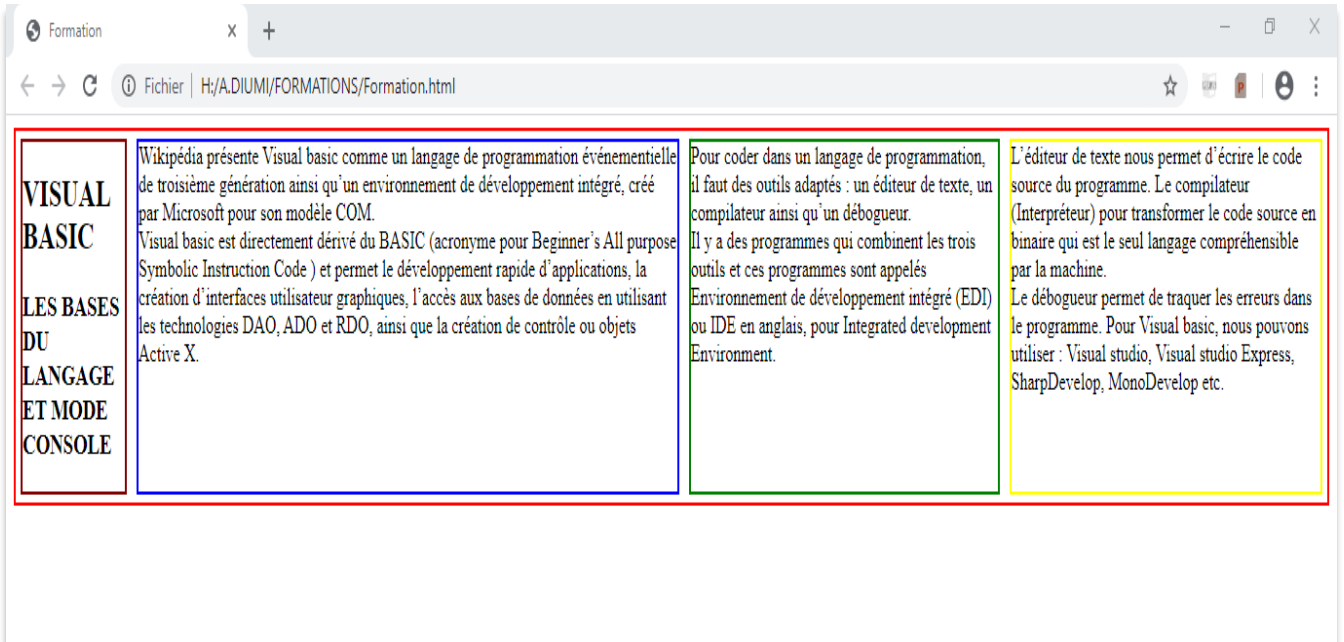
Forme.css

J'ai ajouté une bordure différente à chaque sous bloc afin de bien l'identifier.

```
.bloc  
{  
display: flex;  
flex-direction: row;  
border: 2px red solid;  
}  
  
.para1  
{  
border: 2px blue solid;  
margin: 5px;  
}  
  
.para2  
{  
border: 2px green solid;  
margin: 5px;  
}  
  
.para3  
{  
border: 2px yellow solid;  
margin: 5px;  
}  
  
.titre  
{
```

```
border: 2px maroon solid;
margin: 5px;
size: 1.5em;
}
```

Résultat :



Le premier sous bloc est placé à gauche ainsi de suite...

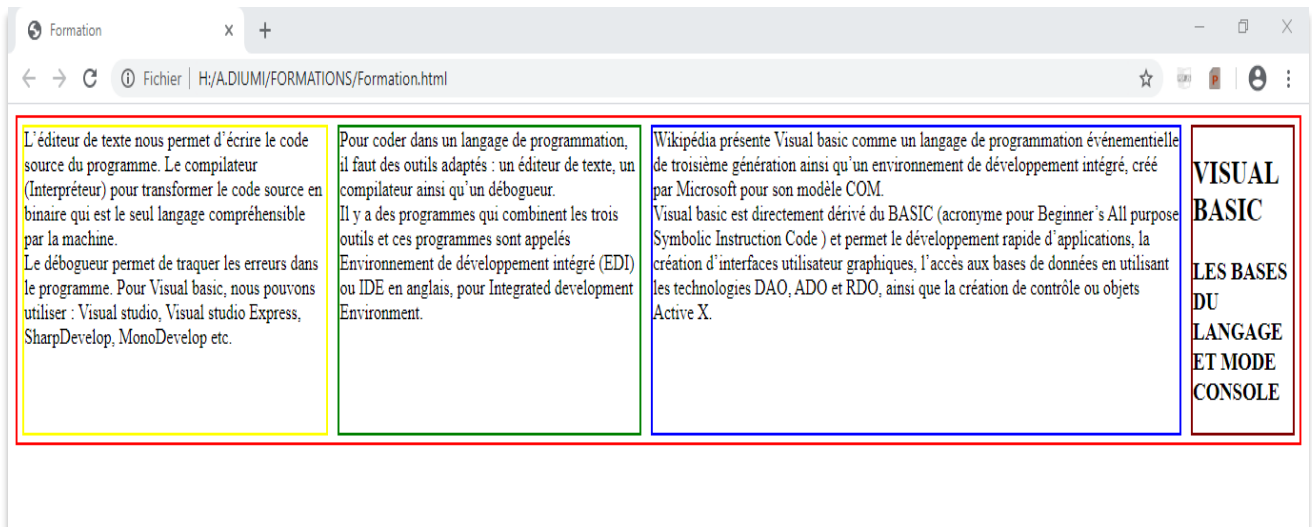
- **row-reverse** : alignement les sous blocs verticalement (sur une ligne) en commençant de la droite vers la gauche.

Forme.css

J'ai mis seulement le code relatif au bloc principal, le reste ne change pas

```
.bloc
{
display: flex;
flex-direction: row-reverse;
border: 2px red solid;
}
```

## Résultat :



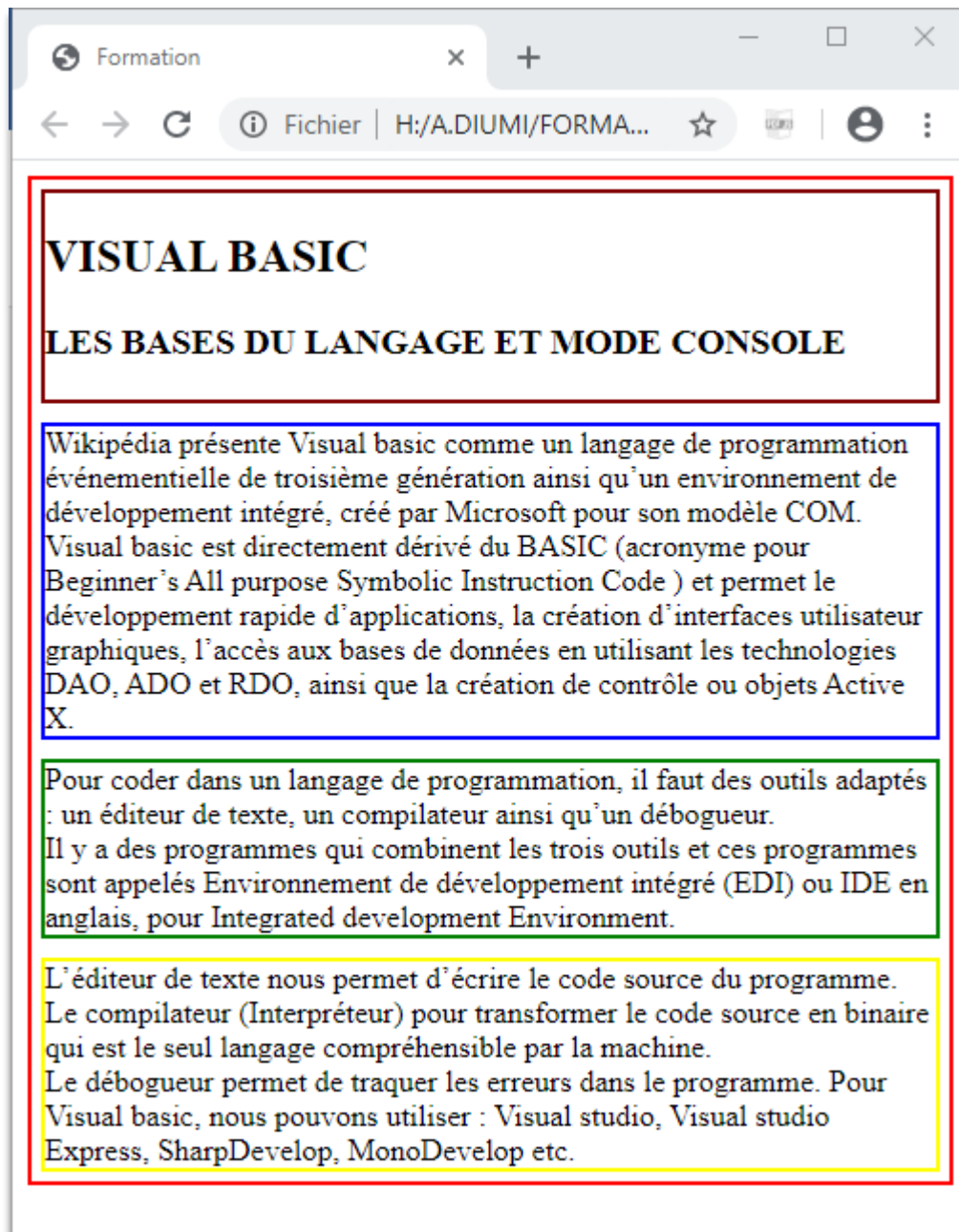
Vous l'avez sans doute remarqué : le premier sous-bloc est placé à droite.

- **Column** : aligne les sous blocs horizontalement (sur une colonne) en commençant du haut vers le bas.

Forme.css

```
.bloc
{
display: flex;
flex-direction: column;
border: 2px red solid;
}
```

Résultat :

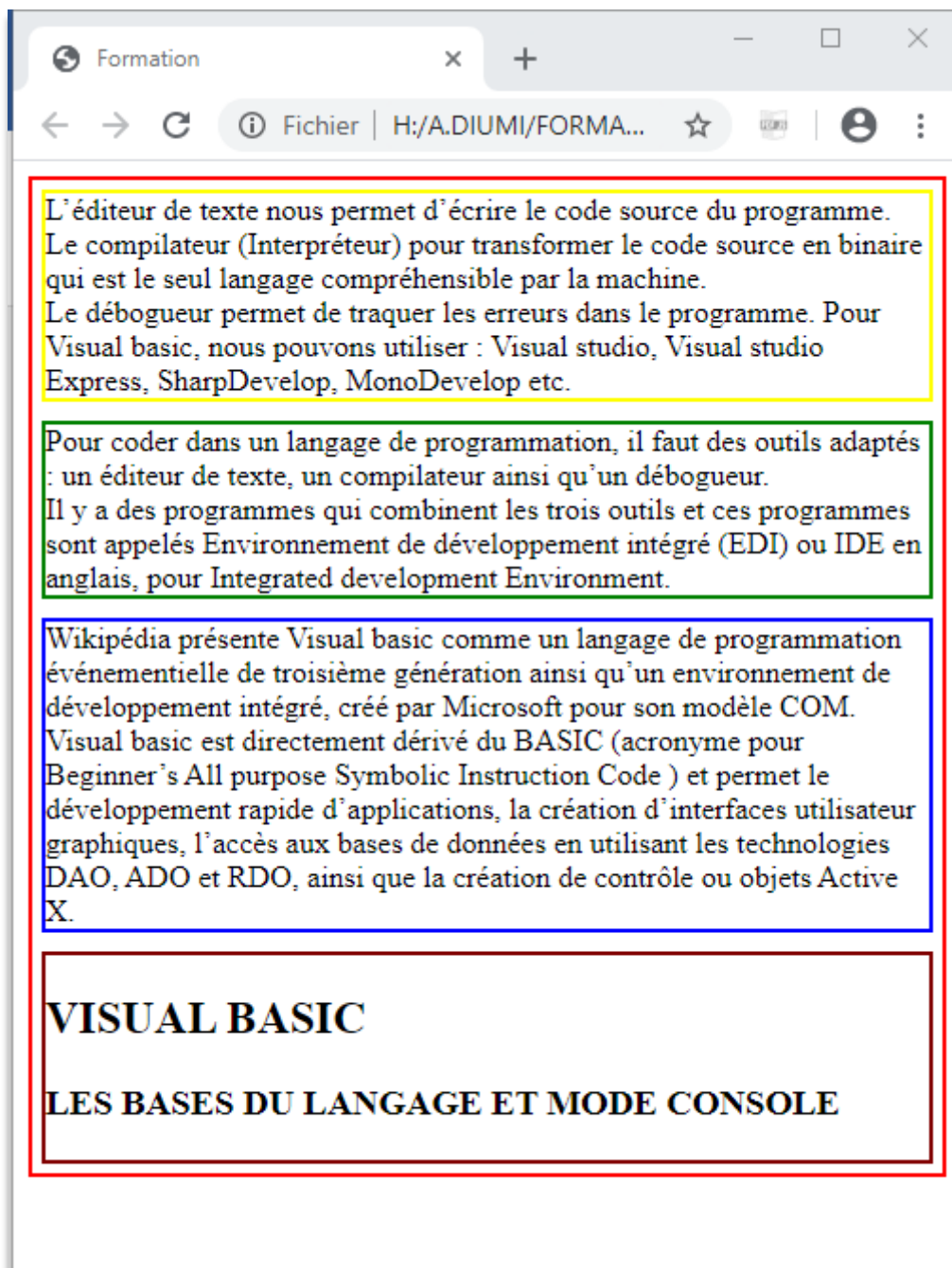


- **column-reverse** : positionne les sous blocs sur une colonne en partant du bas vers le haut.

Forme.css

```
.bloc
{
display: flex;
flex-direction: column-reverse;
border: 2px red solid;
}
```

Résultat :



### *b) La propriété flex-wrap*

Elle permet de gérer le passage des sous blocs à la ligne, elle peut prendre l'une de deux valeurs suivantes :

- **nowrap**: les sous blocs se placent sur une même ligne, il n'y a pas de passage à la ligne. La largeur de chaque sous bloc s'adapte automatiquement pour que tous les sous blocs soient sur la même ligne. C'est le comportement par défaut.



- **wrap** : certains sous blocs vont à la ligne suivante s'il n' y a plus de place.

Comme exemple, utilisons le même fichier html, ajoutons à chaque sous bloc une largeur.

Avec **nowrap** :

Forme.css

```
.bloc
{
display: flex;
flex-wrap: nowrap;
border: 2px red solid;
}

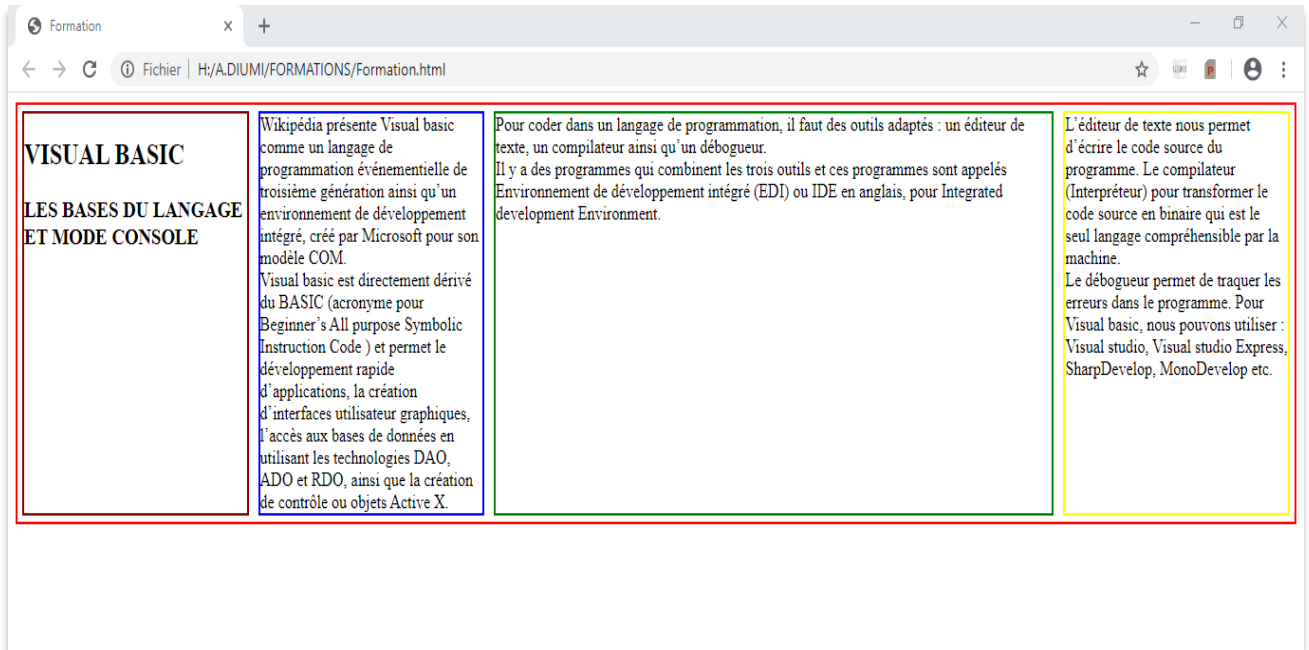
.para1
{
border: 2px blue solid;
margin: 5px;
width: 20%;
}

.para2
{
border: 2px green solid;
margin: 5px;
width: 50%;
}

.para3
{
border: 2px yellow solid;
margin: 5px;
width: 20%;
}

.titre
{
border: 2px maroon solid;
margin: 5px;
size: 1.5em;
width: 20%;
}
```

Résultat :



Le navigateur a ajusté la largeur de chaque sous-bloc pour qu'il se positionnent sur la même ligne.

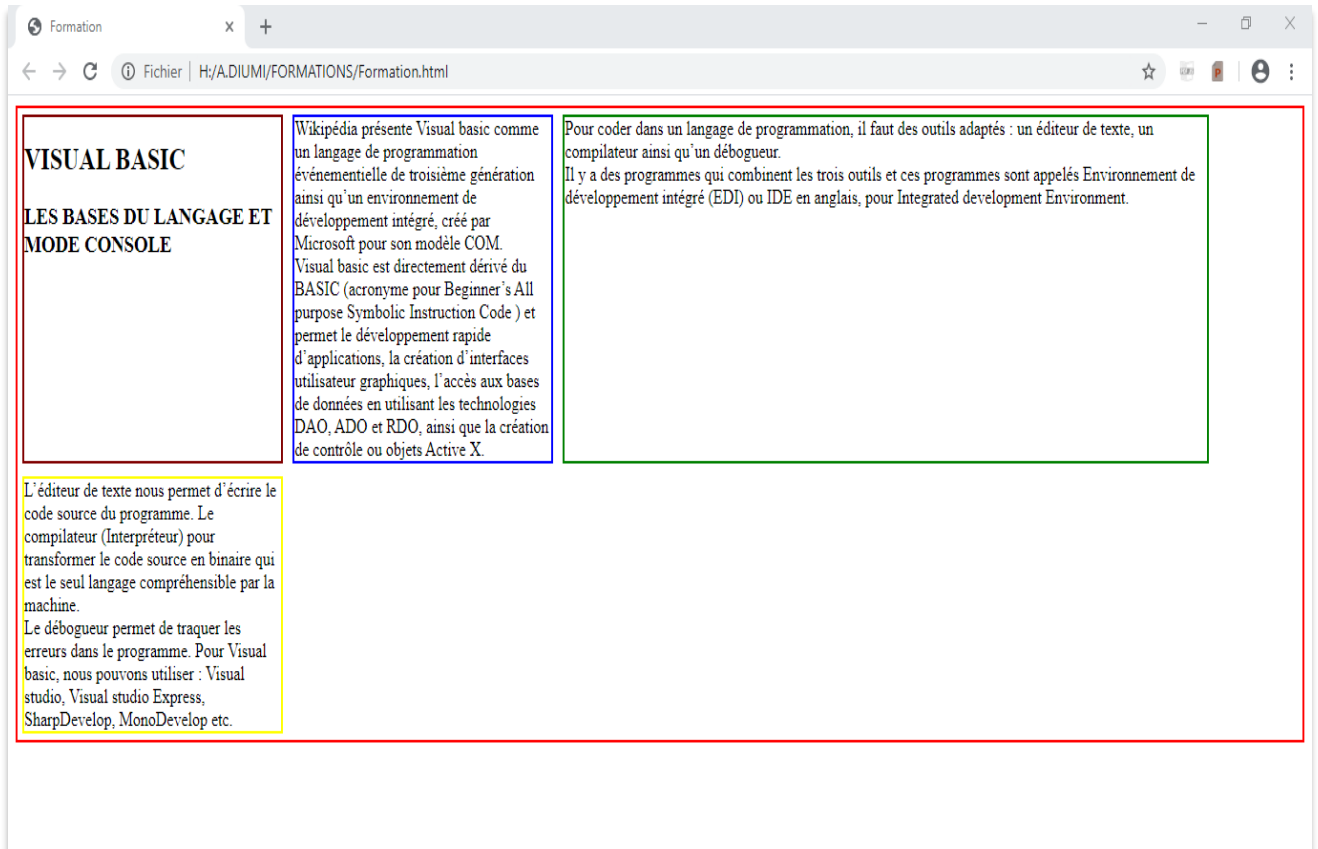
Avec wrap :

Forme.css

J'ai mis seulement le code relatif au bloc principal, le reste de code ne change pas

```
.bloc
{
display: flex;
flex-wrap: wrap;
border: 2px red solid;
}
```

Résultat :



Le dernier sous-bloc passe à la ligne suivante car il n'a plus d'espace sur la première ligne.

### c) La propriété align-items

Permet de gérer l'alignement vertical des sous blocs, elle permet de déplacer l'axe des sous blocs. Les valeurs possibles sont : **flex-end** en bas (ou à droite) ; **flex-start** : en haut (ou à gauche, par défaut) ; **center** : centré ; **baseline** : ligne de base. Pour les exemples suivants, ajoutons une hauteur pour chaque sous-bloc

#### Exemple1

Forme.css

```
.bloc
{
display: flex;
align-items: center;
border: 2px red solid;
}

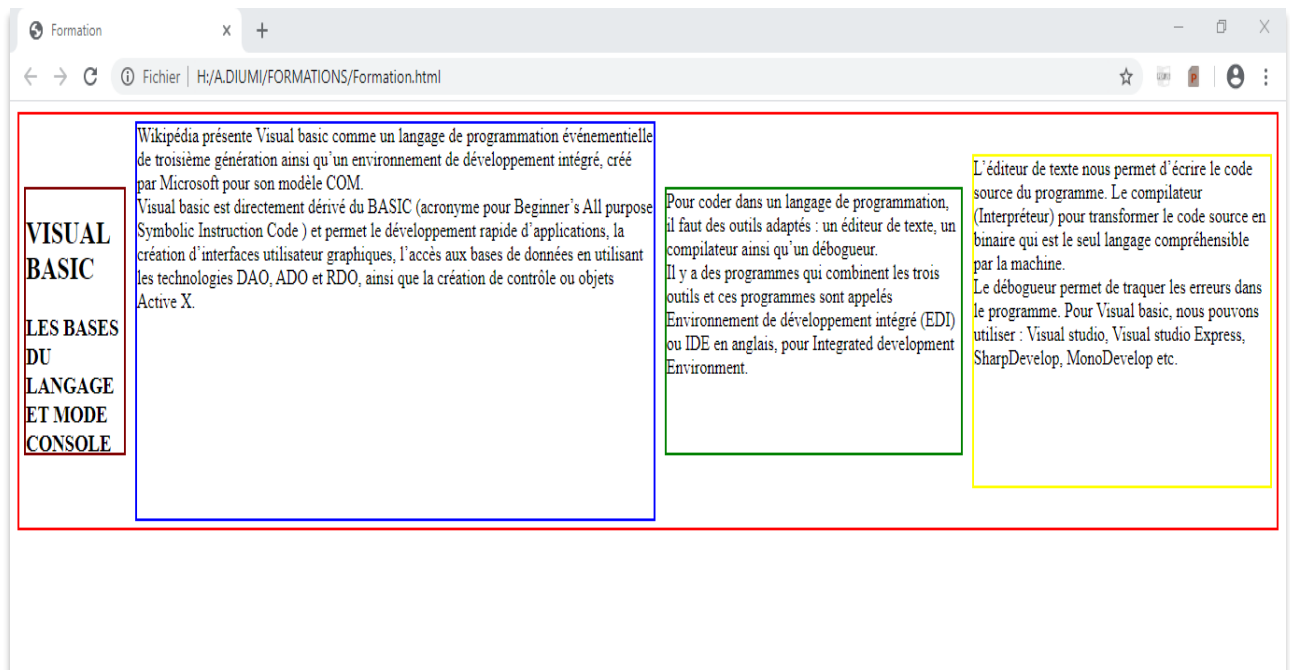
.para1
{
border: 2px blue solid;
margin: 5px;
```

```

height : 300px;
}
.para2
{
    border: 2px green solid;
    margin: 5px;
    height : 200px;
}
.para3
{
    border: 2px yellow solid;
    margin: 5px;
    height : 250px;
}
.titre
{
    border: 2px maroon solid;
    margin: 5px;
    size: 1.5em;
    height :200px;
}
}

```

## Résultat :



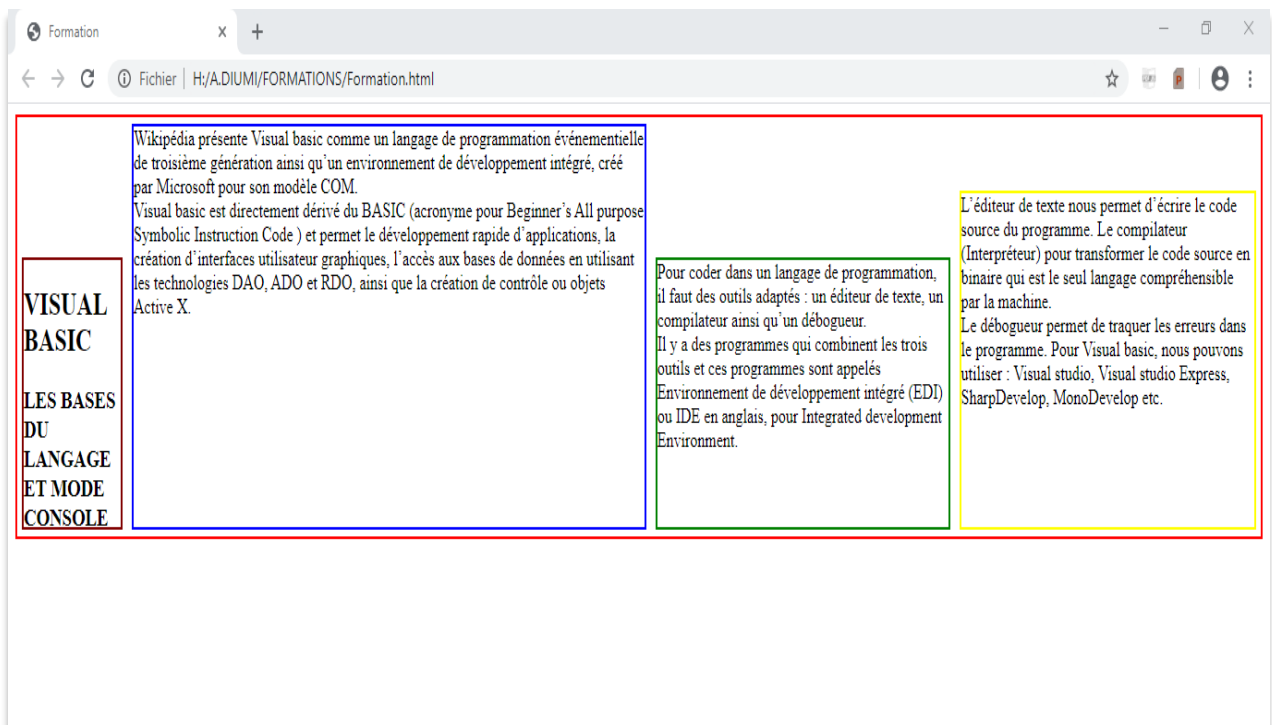
## Exemple 2

J'ai modifié seulement le code du bloc principal, le reste ne change pas

Forme.css

```
.bloc
{
display: flex;
align-items: flex-end;
border: 2px red solid;
}
```

Résultat :

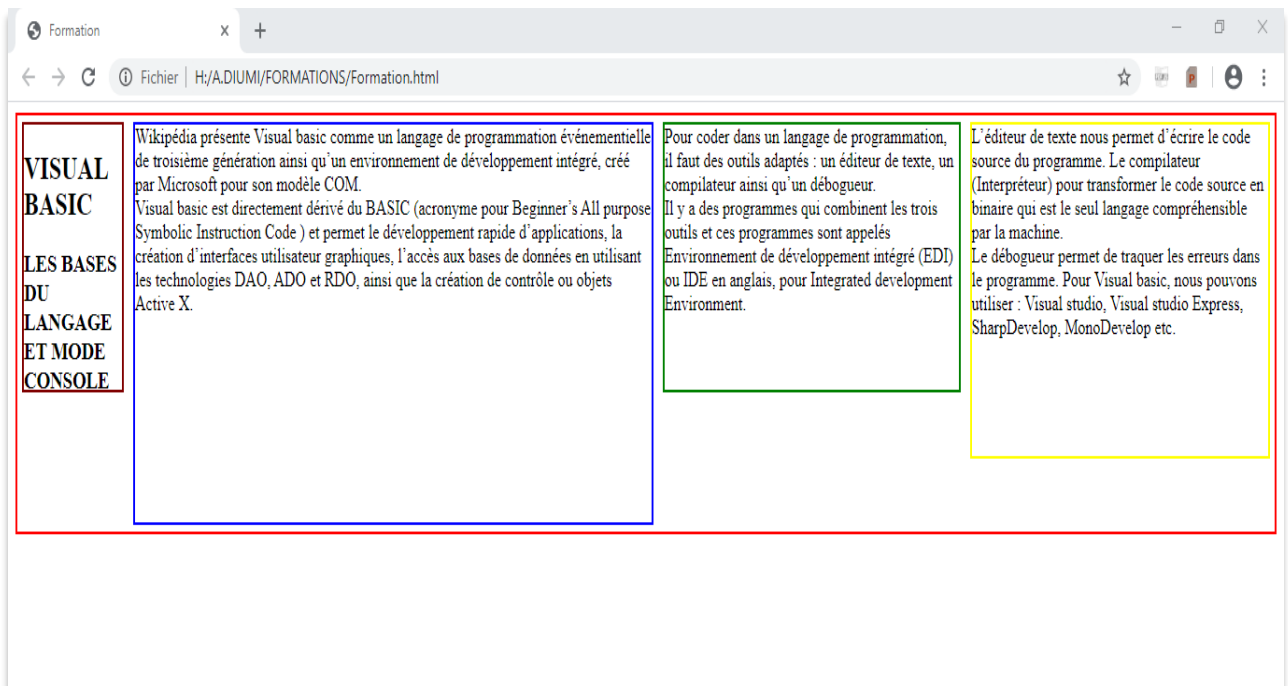


## Exemple3

Forme.css

```
.bloc
{
display: flex;
align-items: flex-start;
border: 2px red solid;
}
```

## Résultat :



### d) La propriété order

Cette propriété permet de choisir l'ordre d'affichage de sous blocs quel que soit l'ordre dans html. Dans notre exemple, le premier sous bloc est le titre, donc c'est lui qui s'affiche le premier par défaut, changeons l'ordre d'affichage :

Forme.css

```
.bloc
{
display: flex;

border: 2px red solid;
}

.para1
{
border: 2px blue solid;
margin: 5px;
height : 300px;
order: 2;
}

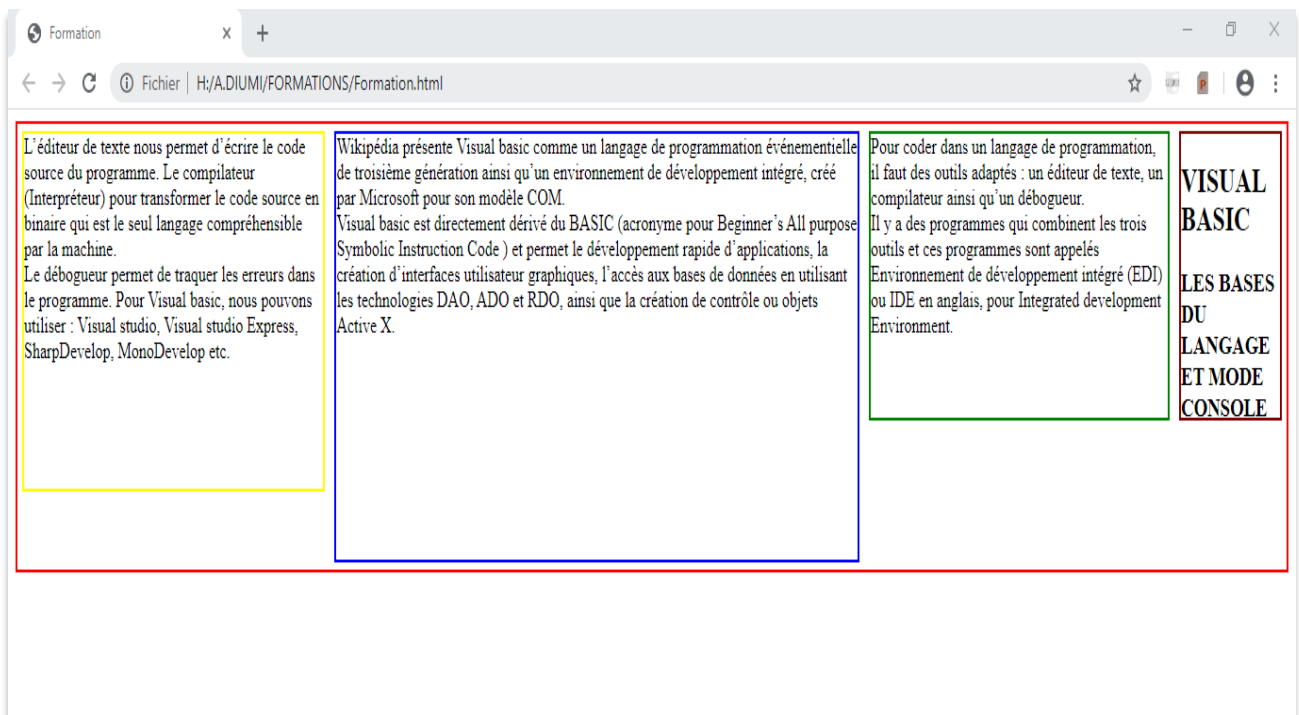
.para2
{
border: 2px green solid;
margin: 5px;
height : 200px;
```

```

    order: 3;
}
.para3
{
    border: 2px yellow solid;
    margin: 5px;
    height : 250px;
    order: 1;
}
.titre
{
    border: 2px maroon solid;
    margin: 5px;
    size: 1.5em;
    height :200px;
    order: 4;
}
}

```

Résultat :



Et voilà, l'ordre a totalement changé par rapport à la disposition en html.

## III.4 LE POSITIONNEMENT ABSOLU

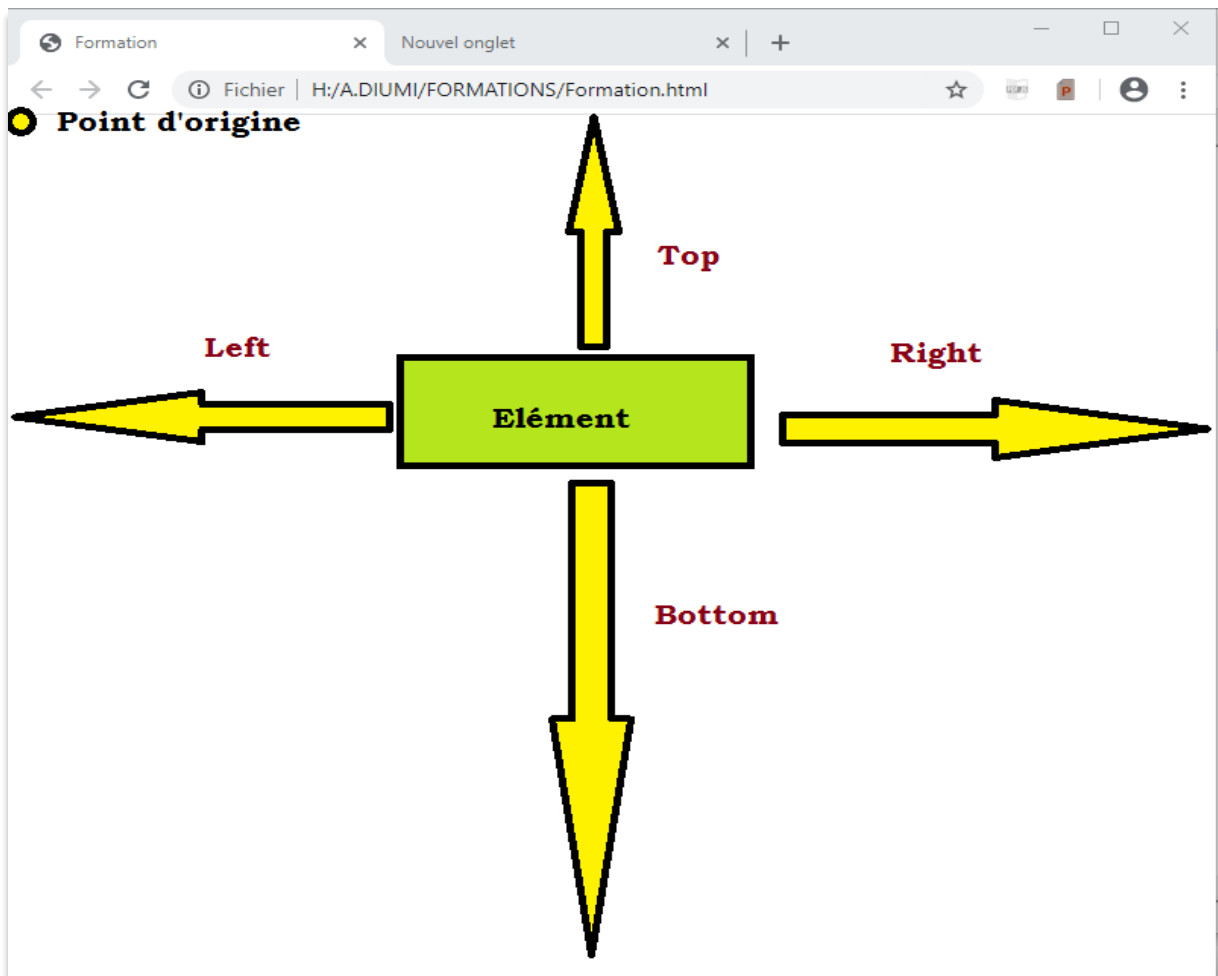
Le positionnement absolu nous permet de placer un élément n'importe où dans la page web. On utilise la propriété position à laquelle on attribue la valeur absolue comme ceci :

```
element
{
    position: absolute;
}
```

Ensuite, il faut indiquer avec précision où l'on souhaite placer l'élément. Pour ce faire, on va utiliser quatre propriétés CSS :

- ❖ **left** : position par rapport à la gauche de la page ;
- ❖ **right** : position par rapport à la droite de la page ;
- ❖ **top** : position par rapport au haut de la page ;
- ❖ **bottom** : position par rapport au bas de la page.

On peut leur donner une valeur en pixels, ou bien une valeur en pourcentage.





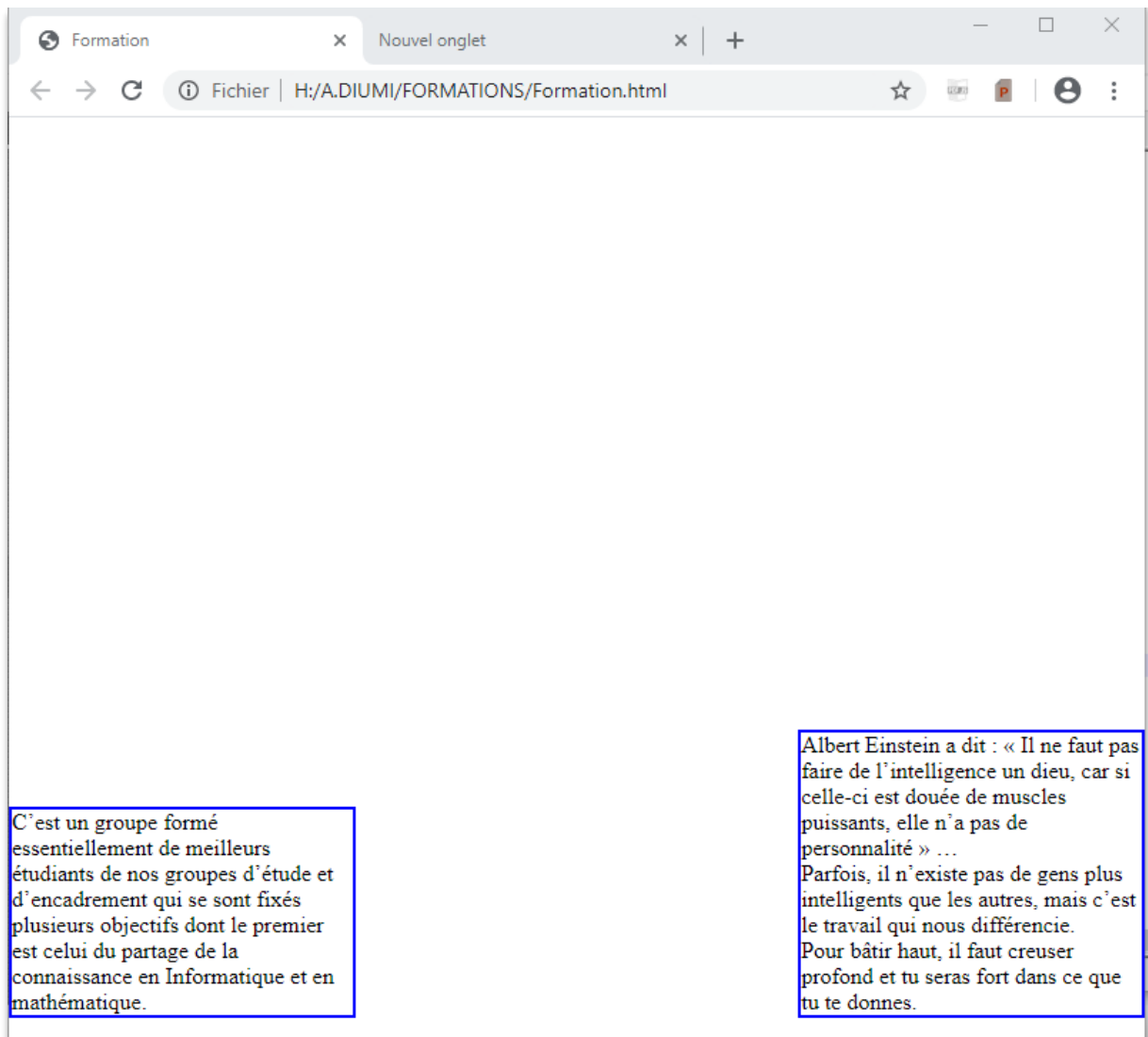
Maintenant, vous libre de positionner votre élément partout où vous voulez en donnant au moins une valeur à ces quatre propriétés.

Reprenons l'exemple de deux paragraphes, si j'ai besoin que le premier se place au coin inférieur gauche et le deuxième au coin inférieur droit, je dois écrire :

Forme.css

```
.para1
{
  position: absolute;
  bottom: 0px;      /* 0% ferait la meme chose */
  left: 0px;
  border: 2px blue solid;
  width: 30%;
}
.para2
{
  position: absolute;
  bottom: 0px;
  right: 0px;
  border: 2px blue solid;
  width: 30%;
}
```

Résultat :



Les éléments positionnés en absolu sont placés par-dessus le reste des éléments de la page. En d'autres termes, un élément positionné en absolu est positionné de façon indépendante de tout texte qui vient avant ou après dans le document, Voyons cela avec un exemple

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
<p class="para1">C'est un groupe formé essentiellement de meilleurs étudiants
de nos groupes d'étude et d'encadrement qui se sont fixés plusieurs objectifs
dont le premier est celui du partage de la connaissance en Informatique et en
mathématique. </p>
```

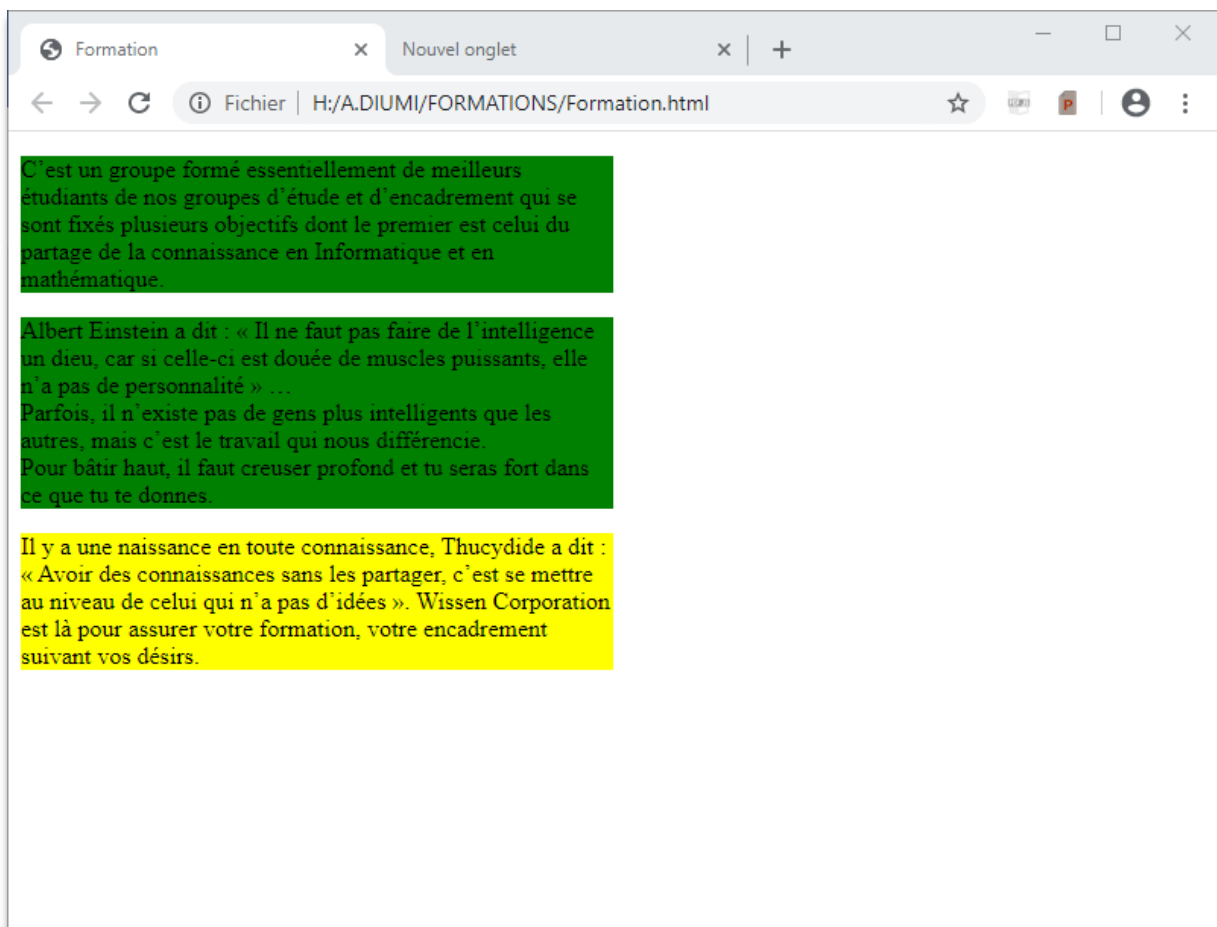
```

<p class="para2">
Albert Einstein a dit : « Il ne faut pas faire de l'intelligence un dieu, car
si celle-
ci est douée de muscles puissants, elle n'a pas de personnalité » ...<br/>
Parfois, il n'existe pas de gens plus intelligents que les autres, mais c'est
le travail qui nous différencie.<br/>
Pour bâtir haut, il faut creuser profond et tu seras fort dans ce que tu te do
nnes. </p>

<p class="para3">Il y a une naissance en toute connaissance, Thucydide a dit :
« Avoir des connaissances sans les partager, c'est se mettre au niveau de cel
ui qui n'a pas d'idées ». Wissen Corporation est là pour assurer votre formati
on, votre encadrement suivant vos désirs. </p>
</body>
</html>

```

Par défaut, le premier élément inséré se positionnera à la première position (coin supérieur gauche), le deuxième suivra le premier ainsi de suite comme le montre la figure suivante :

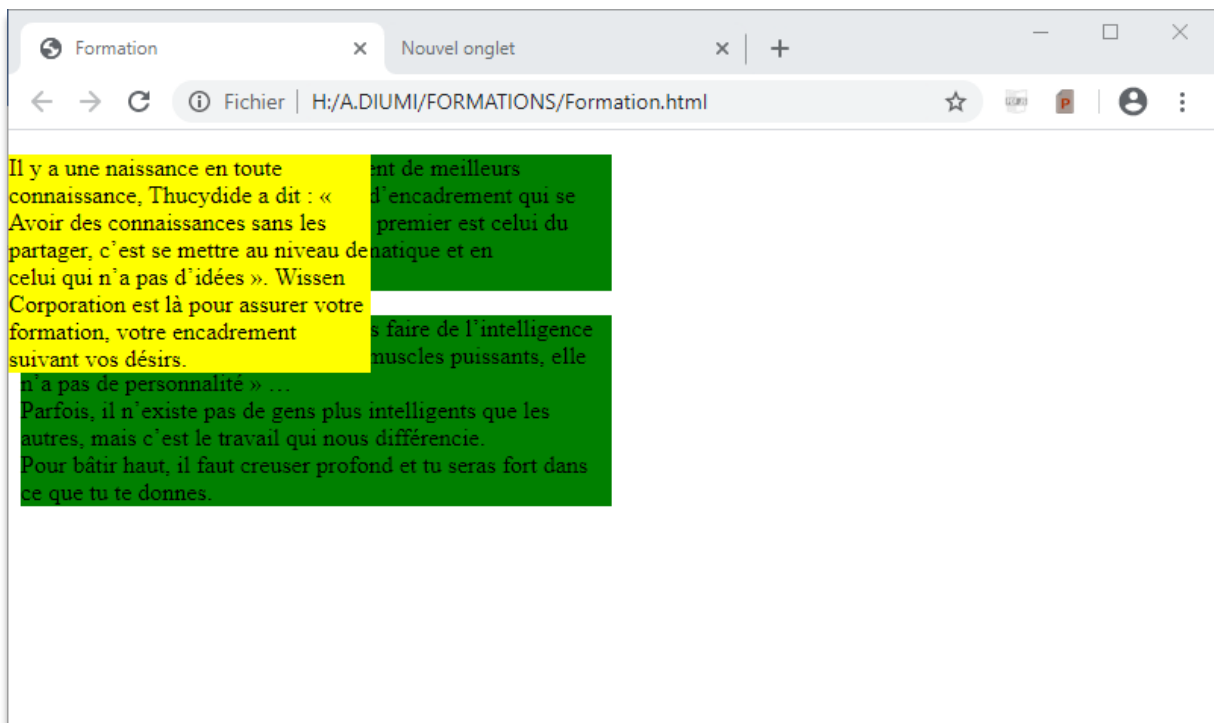


Positionnons le dernier paragraphe (ayant un fond jaune) en absolu en le plaçant au coin supérieur gauche qui revient par défaut au premier paragraphe :

Forme.css

```
.para1, .para2
{
    background-color: green;
    width: 50%;
}
.para3
{
    position: absolute;
    top: 0px;
    left: 0px;
    background-color: yellow;
    width: 30%;
}
```

Résultat :



Le troisième paragraphe positionné en absolu se place par-dessus les deux autres paragraphes.

Par ailleurs, si vous placez deux éléments en absolu vers le même endroit, ils risquent de se chevaucher. Dans ce cas, utilisez la propriété **z-index** pour indiquer quel élément doit apparaître au-dessus des autres. L'élément ayant la valeur de **z-index** la plus élevée sera placé par-dessus les autres, comme l'illustre l'exemple suivant.

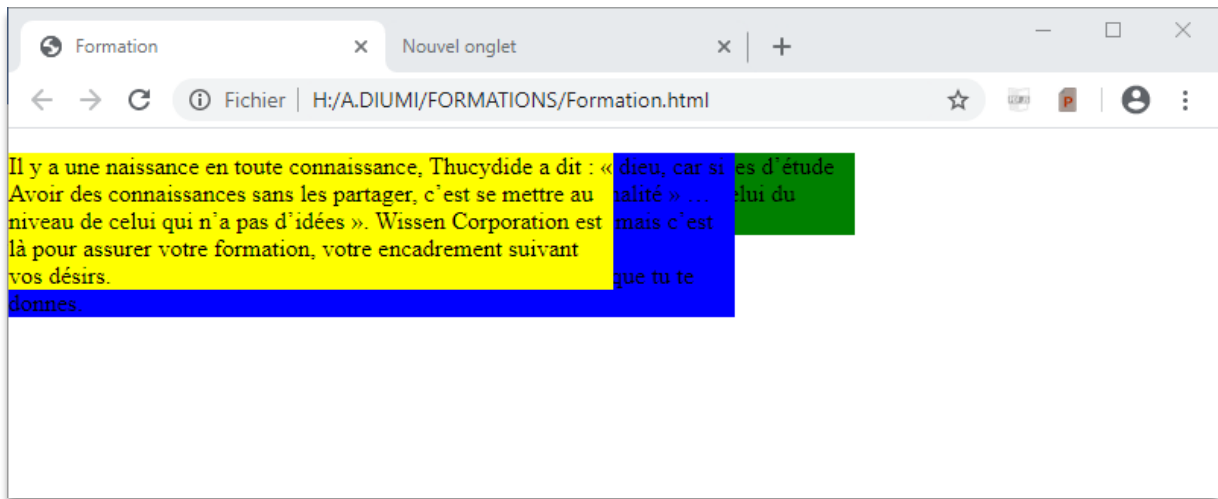
Forme.css

```
.para1
{
  position: absolute;
  top: 0px;
  left: 0px;
  z-index: 1;
  background-color: green;
  width: 70%;
}

.para2
{
  position: absolute;
  top: 0px;
  left: 0px;
  z-index: 2;
  background-color: blue;
  width: 60%;
}

.para3
{
  position: absolute;
  top: 0px;
  left: 0px;
  z-index: 3;
  background-color: yellow;
  width: 50%;
}
```

Résultat :



Le troisième paragraphe se place par-dessus les autres car il a la valeur de z-index la plus élevée.

### III.5 LE POSITIONNEMENT FIXE

Cette technique de positionnement est identique au positionnement absolu mais, cette fois, l'élément reste toujours visible, même si on descend plus bas dans la page. On attribue la valeur **fixed** à la propriété **position**.

Exemple

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel="stylesheet" href="forme.css" />
  </head>
  <body>
    <ul class="menu">
      <li > <a href="Accueil.html" >Accueil </a></li>
      <li> <a href="cours.html" >Cours </a></li>
      <li> <a href="tutos.html" >Tutos </a></li>
      <li> <a href="Forum.html" >Forum </a></li>
      <li> <a href="Apropos.html" >A propos </a></li>
    </ul>
    <p> Paragraphe1 </p>
    <p> Paragraphe2 </p>
    <p> Paragraphe3 </p>
    <p> Paragraphe4 </p>
    <p> Paragraphe5 </p>
```

```
<p> Paragraphe6 </p>
<p> Paragraphe7 </p>
<p> Paragraphe8 </p>
<p> Paragraphe9 </p>
<p> Paragraphe10 </p>
<p> Paragraphe11 </p>
<p> Paragraphe12 </p>
<p> Paragraphe13 </p>
<p> Paragraphe14 </p>
<p> Paragraphe15 </p>
<p> Paragraphe19 </p>
<p> Paragraphe 17 </p>
<p> Paragraphe 18 </p>
<p> Paragraphe 19 </p>
<p> Paragraphe 20 </p>
<p> Paragraphe 21 </p>
<p> Paragraphe 22 </p>
<p> Paragraphe 23 </p>
<p> Paragraphe 24 </p>
<p> Paragraphe 25 </p>
<p> Paragraphe 26 </p>
<p> Paragraphe 27 </p>
<p> Paragraphe 28 </p>
<p> Paragraphe 29 </p>
<p> Paragraphe 30 </p>
</body>
</html>
```

## Forme.css

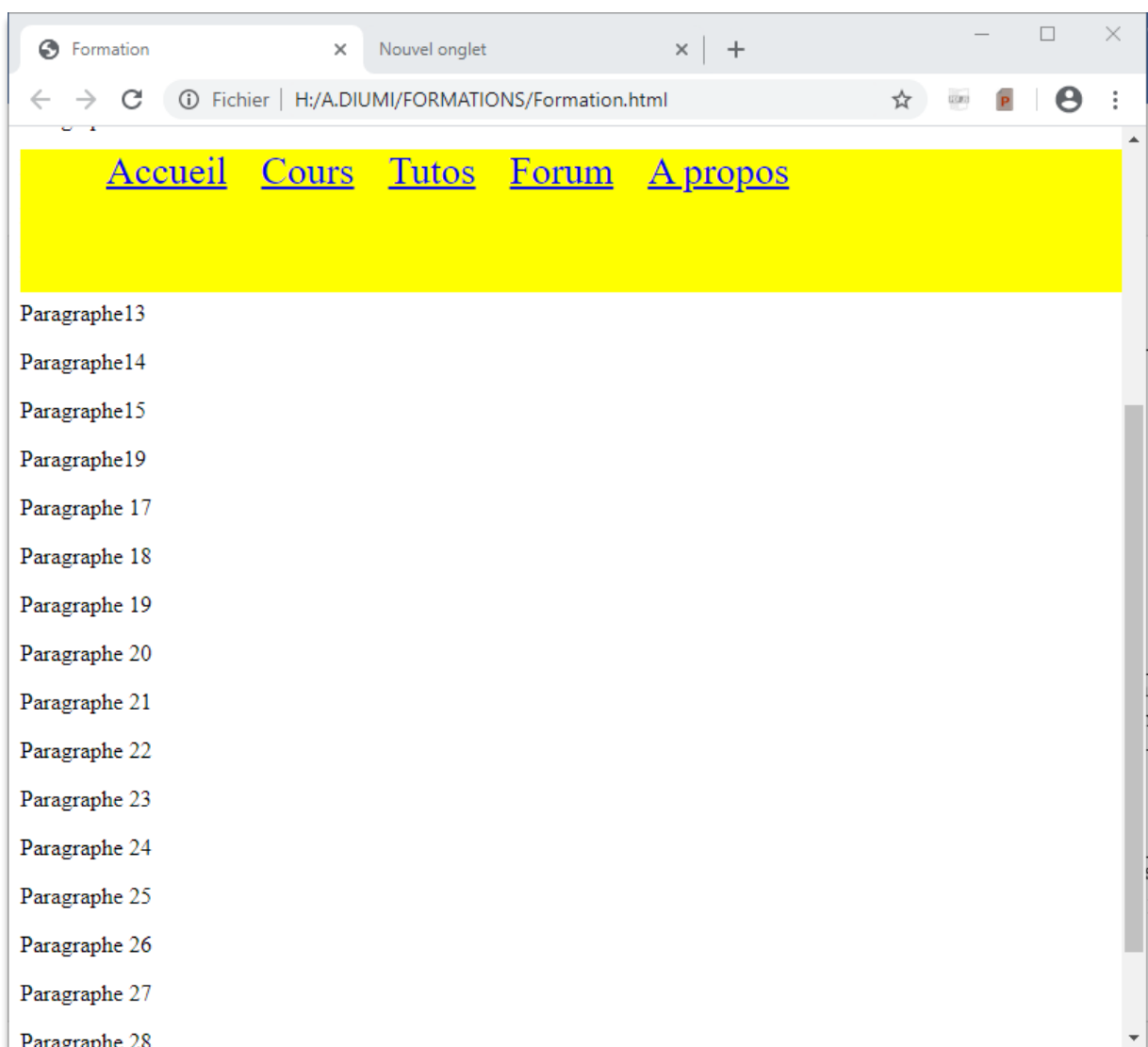
```
ul
{
  list-style-type: none;
  background-color:yellow;
  height: 100px;
}

li
{
  display: inline-block;
  margin-left: 20px;
  font-size: 1.7em;
}
```

J'aimerais que les principaux liens de navigation restent visibles même si on descend plus bas dans la page, je dois ajouter ceci dans CSS :

```
.menu
{
  position: fixed;
  top:0px;
}
```

Essayez d'observer le résultat, vous verrez que les liens restent toujours visibles en haut même si on descend plus bas dans la page (figure suivante).





## III.5 POSITIONNEMENT RELATIF

Ce positionnement permet d'effectuer des « ajustements » : l'élément est décalé par rapport à sa position initiale.

Par exemple, dans le paragraphe suivant :

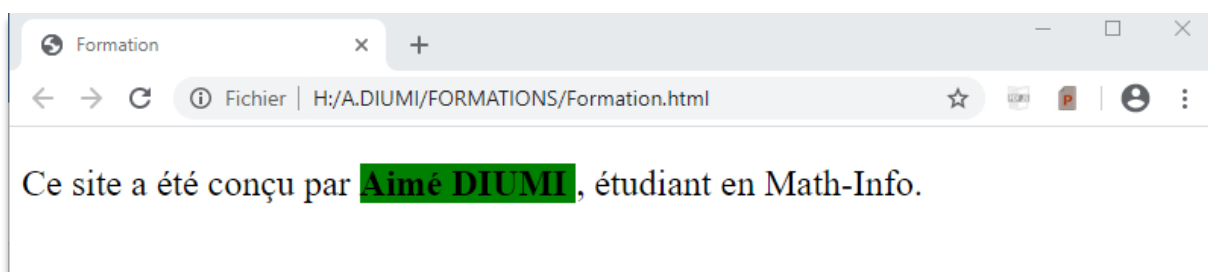
Formation.html

```
<p> Ce site a été conçu par <span class="nom"> Aimé DIUMI </span>, étudiant en Math-Info. </p>
```

J'ai ajouté une couleur de fond au nom Aimé DIUMI pour qu'il soit visible :

forme.css

```
.nom
{
  background-color: green;
  font-weight: bolder;
}
```



Si on applique un positionnement relatif au nom « Aimé DIUMI », l'origine a changé : le point de coordonnées (0, 0) ne se trouve plus en haut à gauche de la fenêtre cette fois l'origine se trouve en haut à gauche... de la position actuelle de votre élément comme le montre la figure suivante :

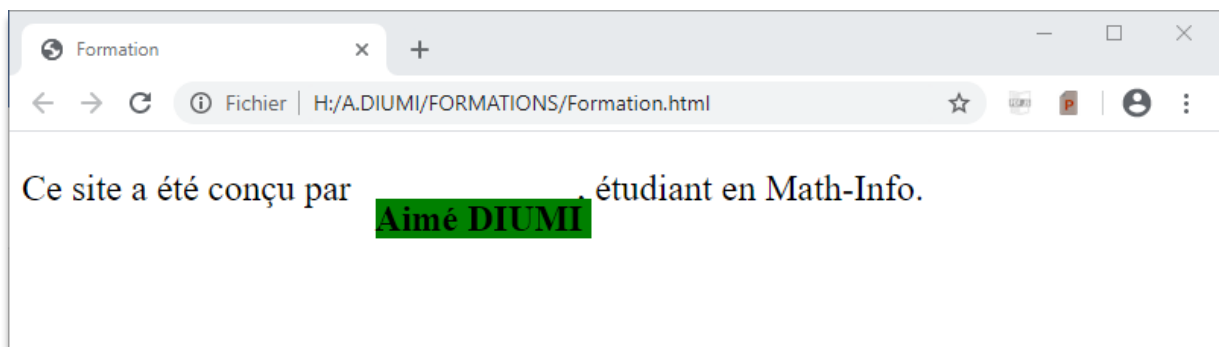


Si on applique les propriétés top, left, right et bottom, l'élément va se déplacer par rapport à cette position. Essayons de tester cela avec le décalage suivant :

Forme.css

```
.nom
{
  position:relative;
  top:20px;
  left: 10px;
  background-color: green;
  font-weight: bolder;
}
```

Résultat :



Notez que les autres éléments ne sont pas influencés par le décalage de l'élément positionné en relatif.

## III.6 POSITIONNEMENT MULTI COLONNE

Le module de positionnement multicolonne (ou Multi-column Layout

Module en anglais) permet de faire écouler du contenu sur plusieurs colonnes de largeurs égales, tel qu'on peut le voir dans le monde de l'imprimerie. En d'autres termes, ce mode de positionnement permet d'afficher ou repartir le contenu sur plusieurs colonnes de même largeur.

Pour cela, on utilise la propriété **columns** et ses propriétés dérivées :

Ces différentes propriétés peuvent être scindées en trois parties fonctionnelles :

- Le premier groupe de propriétés définit le nombre et la taille des colonnes :
  - **columns** (raccourci de column-count et column-width) : nombre de colonnes et éventuellement largeur de chaque colonne ;
  - column-min-width : largeur minimale de chaque colonne ;
  - column-width-policy : le mode d'affichage des colonnes (valeurs "flexible", "strict" ou "inherit").
- Le second groupe gère ce qu'il y a entre les colonnes :
  - column-gap : distance entre chaque colonne ;
  - column-rule (raccourci de column-rule-color, column-rule-style et column-rule-width) : couleur, style et largeur de la séparation entre colonnes.
- Enfin, une propriété column-span permet à un élément de s'étendre sur plusieurs colonnes.

Prenons l'exemple du fichier html suivant :

Formation.html

```
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>

<section class="bloc">
<h1>AVANT PROPOS </h1>
<h3>WISSEN CORPORATION </h3>

<p>C'est un groupe formé essentiellement de meilleurs étudiants de nos
groupes d'étude et d'encadrement qui se sont fixés plusieurs objectifs dont
le premier est celui du partage de la connaissance en Informatique et en
```

mathématique. </p>

<p>Il y a une naissance en toute connaissance, Thucydide a dit : « Avoir des connaissances sans les partager, c'est se mettre au niveau de celui qui n'a pas d'idées ». Wissen Corporation est là pour assurer votre formation, votre encadrement suivant vos désirs. </p>

<h1> VISUAL BASIC </h1>

<h3> LES BASES DU LANGAGE </h3>

<p>Wikipédia présente Visual basic comme un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, créé par Microsoft pour son modèle COM. <br/>Visual basic est directement dérivé du BASIC (acronyme pour Beginner's All purpose Symbolic Instruction Code ) et permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôle ou objets Active X. </p>

<p>Pour coder dans un langage de programmation, il faut des outils adaptés : un éditeur de texte, un compilateur ainsi qu'un débogueur.<br/> Il y a des programmes qui combinent les trois outils et ces programmes sont appelés Environnement de développement intégré (EDI) ou IDE en anglais, pour Integrated development Environment. </p>

<p>L'éditeur de texte nous permet d'écrire le code source du programme. Le compilateur (Interpréteur) pour transformer le code source en binaire qui est le seul langage compréhensible par la machine.<br/> Le débogueur permet de traquer les erreurs dans le programme. Pour Visual basic, nous pouvons utiliser : Visual studio, Visual studio Express, SharpDevelop, MonoDevelop etc. </p>

<h4>I. LES VARIABLES </h4>

<p>Visual Basic, comme la plupart des langages de programmation, utilise des variables pour stocker des valeurs.<br/>

Elles servent à conserver momentanément des données en mémoire. </p></section>

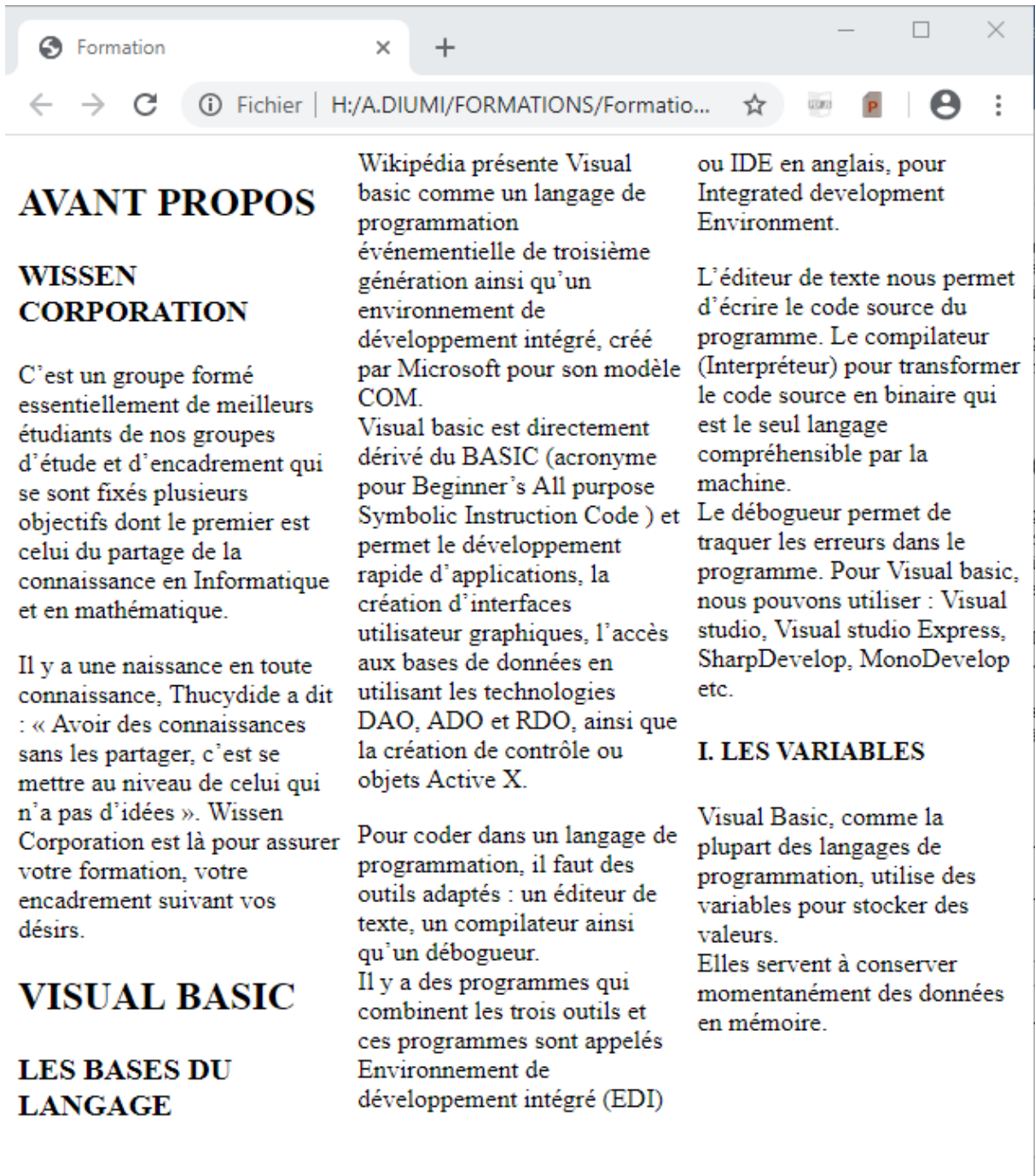
</body>

</html>

Forme.css

```
.bloc
{
  column-count:3;
  column-gap: 10px;
}
```

Résultat :



## III.7 LA TRANSFORMATION EN CSS

La propriété **transform** permet d'appliquer des transformations en deux dimensions sur un élément : rotation, décalage, zoom, déformation et perspective.

### III.7.1 La rotation

Pour effectuer une rotation dans le sens des aiguilles d'une montre, on utilise la fonction rotate dont les valeurs sont exprimées en degrés (deg) ou en radians (rad).

Exemple

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel="stylesheet" href="forme.css" />
  </head>
  <body>

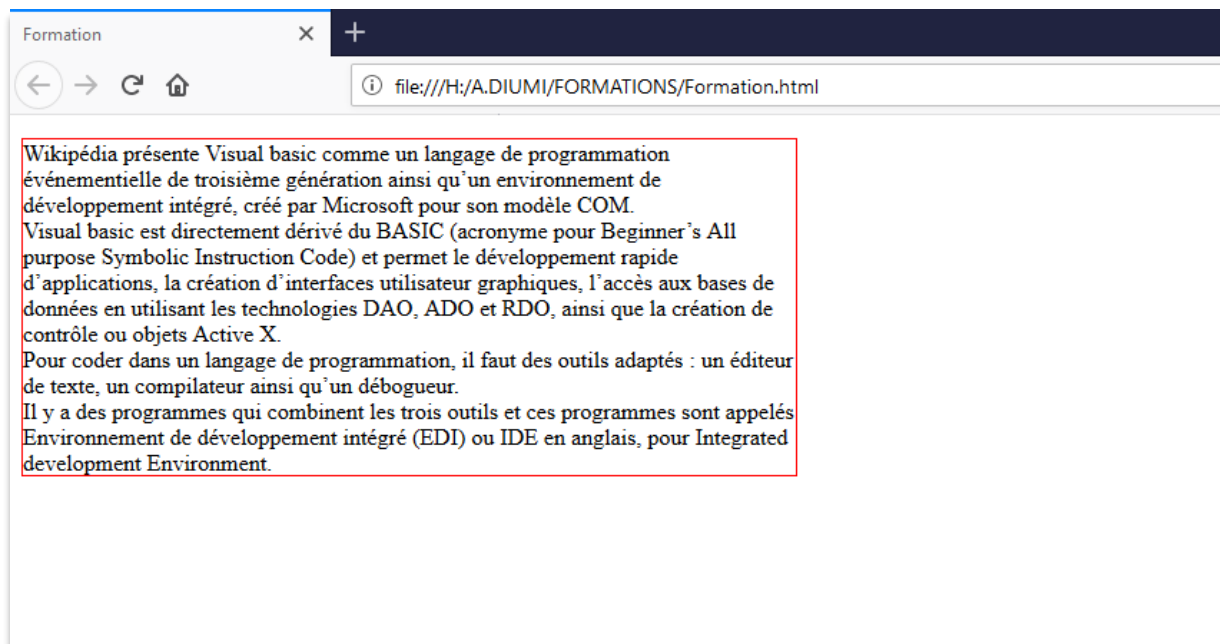
<p>Wikipédia présente Visual basic comme
un langage de programmation événementielle de troisième génération ainsi
qu'un environnement de développement intégré, créé par Microsoft pour son mod
èle COM.<br/>
Visual basic est directement dérivé du BASIC (acronyme pour Beginner's All pur
pose Symbolic Instruction Code)
et permet le développement rapide d'applications, la création
d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisan
t les technologies DAO,
ADO et RDO, ainsi que la création de contrôle ou objets Active X. <br/>
Pour coder dans un langage de programmation, il faut des outils adaptés :
un éditeur de texte, un compilateur ainsi qu'un débogueur. <br/>
Il y a des programmes qui combinent les trois outils et ces programmes sont a
ppelés
Environnement de développement intégré (EDI) ou IDE en anglais, pour Integrat
ed development Environment. </p>
</body>
</html>
```

Ajoutons une bordure et fixons la largeur de 50% au paragraphe

Forme.css

```
p
{
width: 40%;
border:1px red solid;
}
```

Résultat :

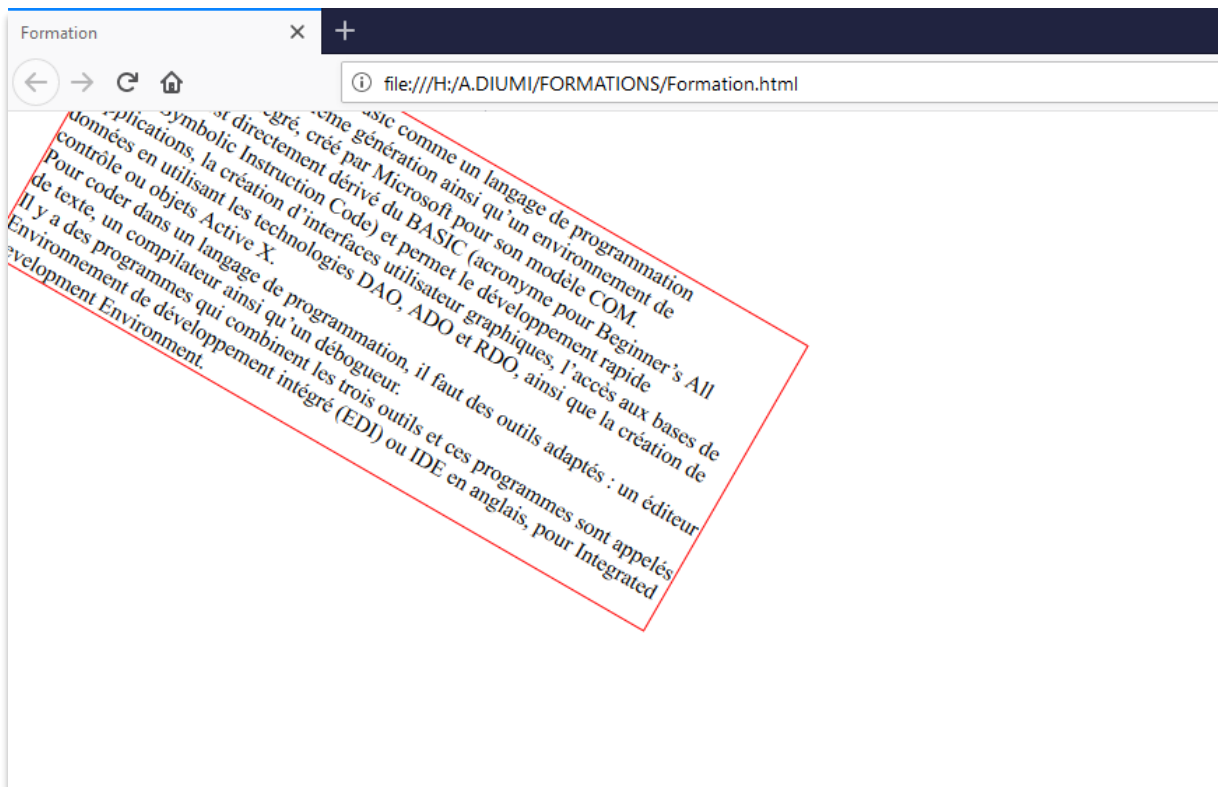


Tournons ce paragraphe à 30 degrés dans le sens des aiguilles d'une montre :

Forme.css

```
p
{
width: 40%;
border:1px red solid;
transform:rotate(30deg);
}
```

Résultat :



### III.7.2 La déformation oblique

Pour obtenir une déformation oblique d'un élément, on utilise toujours la propriété **transform** mais avec le paramètre **shew(angle)**

Où l'angle peut être défini en gradients (grad), radians (rad) ou degrés (deg). Elle peut accepter une valeur positive (déplacement dans le sens des aiguilles d'une montre) ou négative (sens contraire des aiguilles d'une montre).

Exemple

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel="stylesheet" href="forme.css" />
  </head>
  <body>

  <p>Wikipédia présente Visual basic comme un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, créé par Microsoft pour son modèle COM. <br/>Visual basic est directement
```



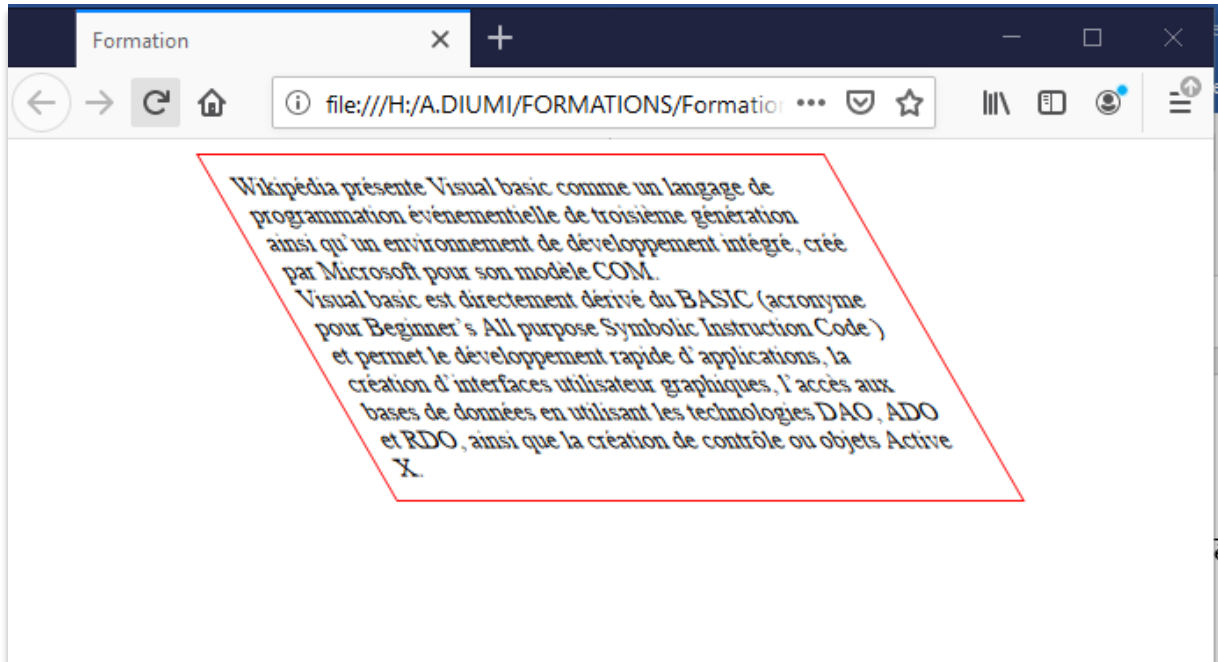
```
dérivé du BASIC (acronyme pour Beginner's All purpose Symbolic Instruction Code) et permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôle ou objets Active X. </p>
</body>
</html>
```

Forme.css

J'ai ajouté une bordure et des marges au paragraphe

```
p
{
border:red solid 1px;
transform:skew(30deg);
width:50%;
margin:auto;
padding:10px;
}
```

Résultat :

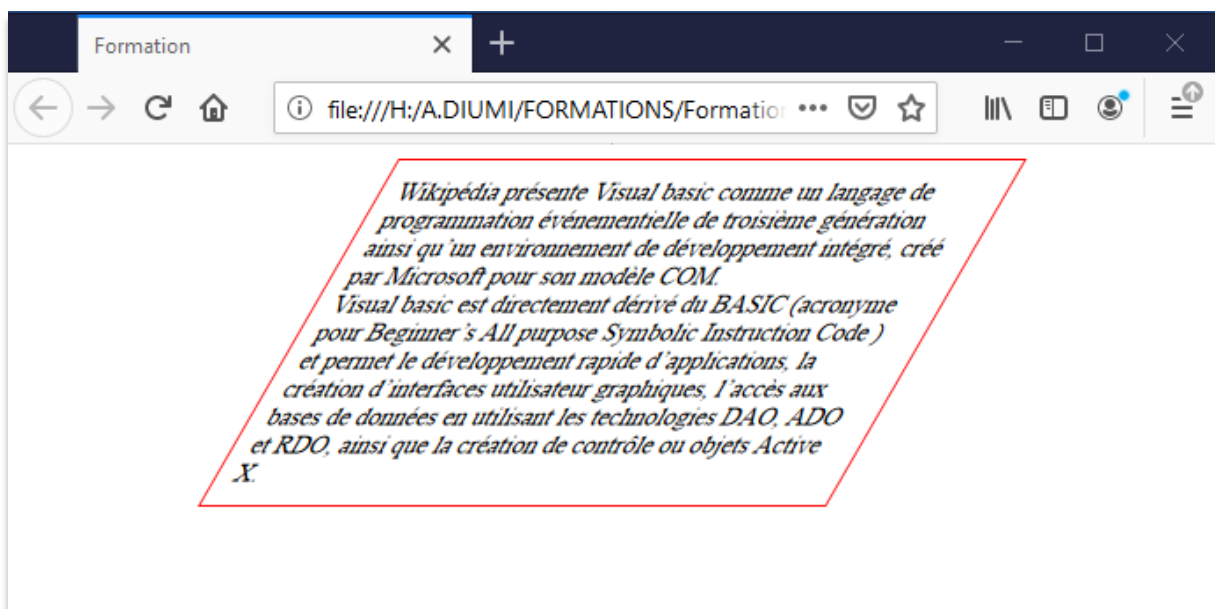


Essayons maintenant avec une valeur négative

Forme.css

```
p
{
border:red solid 1px;
transform:skew(-30deg);
width:50%;
margin:auto;
padding:10px;
}
```

Résultat :



### III.7.3 Le déplacement

Pour effectuer le déplacement d'un élément, on utilise le paramètre **translate** qui s'utilise comme suit `transform: translate (X, Y) ;`

Où

X et Y représentent respectivement la longueur du déplacement sur l'axe des abscisses (axes des X) et sur l'axe des ordonnées (axes des Y). Elles acceptent des valeurs positives ou négatives. Pour X, une valeur positive déplacera l'élément vers la droite et une valeur négative vers la gauche. Pour Y, une valeur positive déplacera l'élément vers le haut et une valeur négative vers le bas.

Notez qu'il est tout en fait possible d'indiquer qu'une seule valeur qui s'appliquera dans ce cas pour les deux axes.

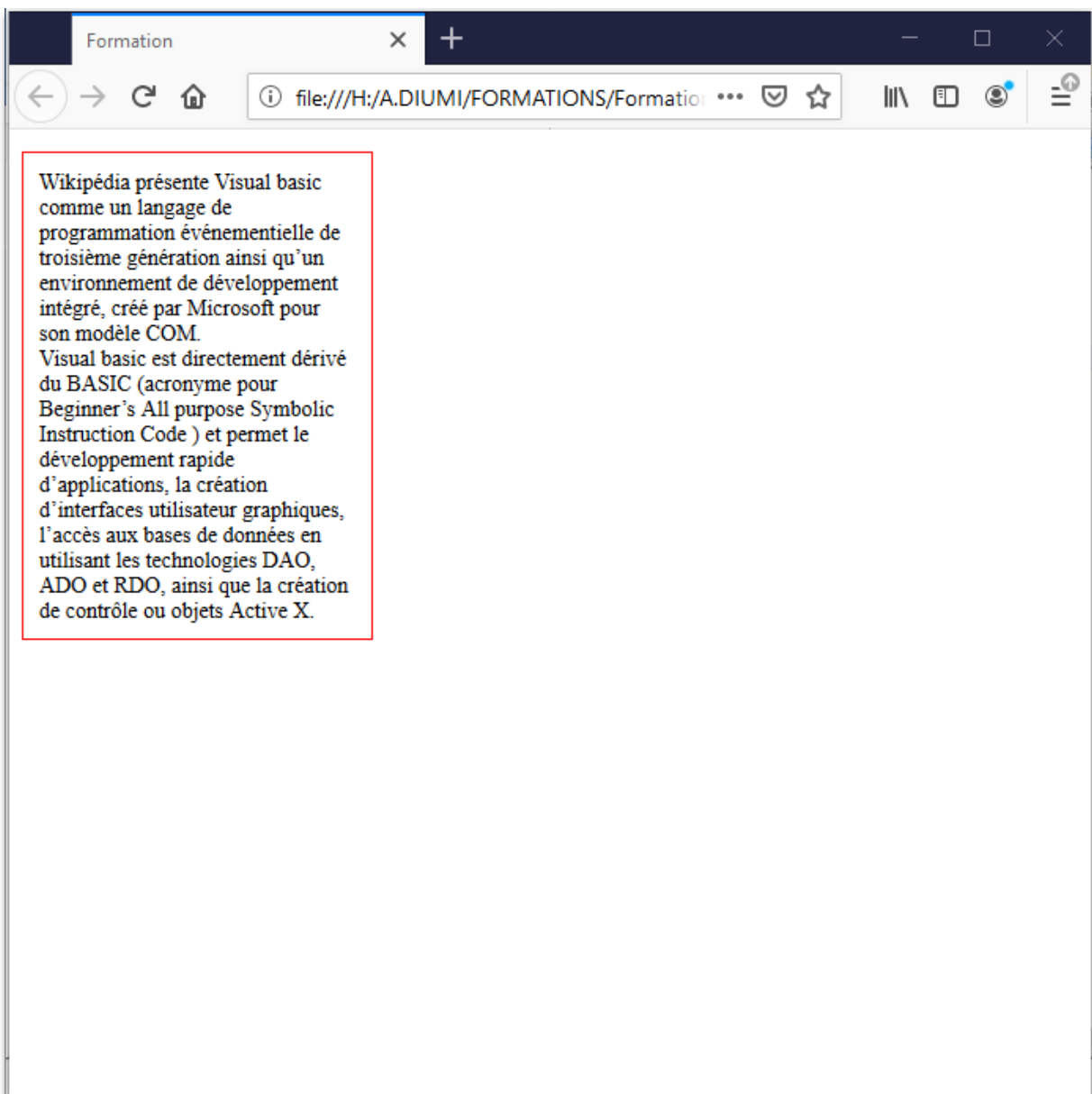
### Exemple

Considérons le même fichier html ci-haut. Appliquons-lui le style suivant :

Forme.css

```
p
{
border:red solid 1px;
width:30%;
padding:10px;
}
```

Résultat :



Si je veux déplacer ce paragraphe de 200px vers la droite et de 100px vers le bas, j'écrirai :

Forme.css















```
p
{
  transform:translate(200px,100px);
border:red solid 1px;
width:30%;
padding:10px;
}
```

Résultat :



### III.7.4 Le curseur de la souris

Il est possible de changer l'apparence du curseur lorsqu'il pointe sur un élément quelconque. On utilise la propriété **cursor** qui peut prendre une des valeurs représentées par la figure suivante :

pointer	
move	
e-resize	
ne-resize	
nw-resize	
n-resize	
se-resize	
sw-resize	
s-resize	
w-resize	
text	
wait	
help	
crosshair	

Exemple

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel="stylesheet" href="forme.css" />
  </head>
  <body>
    <p> C'est un groupe formé essentiellement de meilleurs étudiants de nos
```

```
groupes d'étude et d'encadrement qui se sont fixés plusieurs objectifs dont le premier est celui du partage de la connaissance en Informatique et en mathématique. </p>
```

```
</body>  
</html>
```

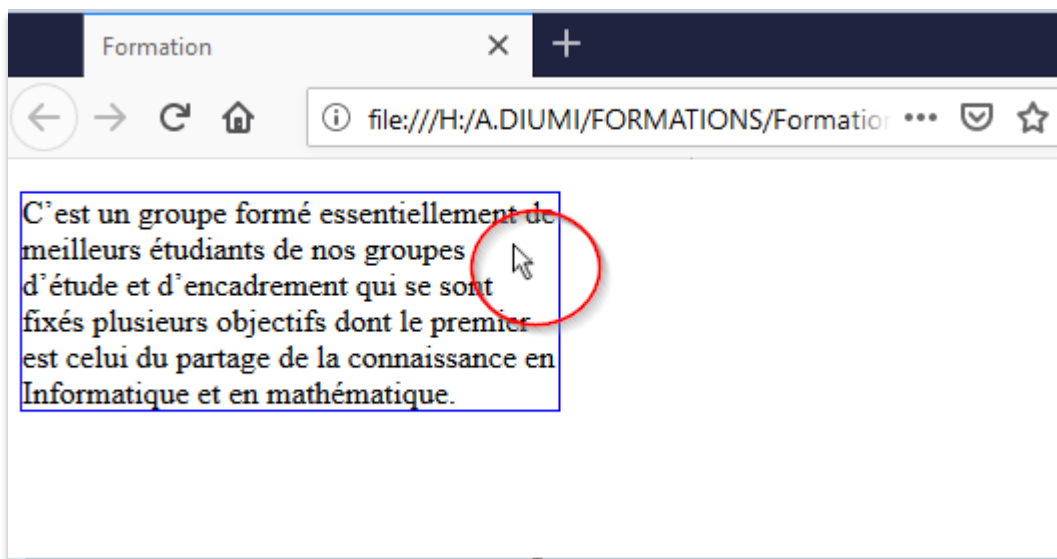
Ajoutons une bordure et une largeur de 40% au paragraphe

Forme.css

```
p  
{  
  border:1px blue solid;  
  width: 40%;  
}
```

Par défaut (pour Mozilla Firefox), voici la forme du curseur lors du survol de la souris sur l'élément :

Sur une zone vide :



Changeons cette apparence :

## Forme.css

```
p
{
  cursor:move;
  border:1px blue solid;
  width: 40%;
}
```

Testez, vous allez voir le résultat.

Notez au moins que l'apparence du curseur peut varier selon le navigateur et le système d'exploitation.

## IV. LES TABLEAUX

### IV.1 Création d'un tableau

Pour créer un tableau en html, on utilise la balise `<table>` `</table>`.

```
<table>      Début du tableau  
  
Contenu du tableau  
  
</table>    Fin du tableau
```

Pour insérer le contenu du tableau, sachez qu'en HTML, un tableau se construit ligne par ligne. Dans chaque ligne, on indique le contenu des différentes cellules.

Chaque ligne du tableau est délimitée par la balise `<tr>` `</tr>`.

Dans chaque ligne, on doit indiquer le contenu des différentes cellules et chaque contenu de la cellule est délimité par la balise `<td>` `</td>`.

Exemple

Je veux construire un tableau de quatre lignes de 3 colonnes chacune, j'écrirai :

Formation.html

```
<!doctype html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <title> Formation</title>  
    <link rel="stylesheet" href="forme.css" />  
  </head>  
  <body>  
  
    <table>  
    <tr>  
    <td> Aimé           </td>  
    <td> DIUMI          </td>  
    <td>  Mont Ngafula   </td>  
    </tr>  
    <tr>  
    <td> Kleine         </td>  
    <td> MBO            </td>  
    <td> Lemba          </td>
```



```

    </tr>
<tr>
<td> Julie          </td>
<td> DIKOLO         </td>
<td> Masina         </td>
    </tr>

    <tr>
<td> Claudia        </td>
<td> FALASH         </td>
<td> Lemba          </td>
    </tr>
</table>

</body>
</html>

```

Résultat :

Aimé	DIUMI	Mont Ngafula
Kleine	MBO	Lemba
Julie	DIKOLO	Masina
Claudia	FALASH	Lemba

Notre tableau est créé mais il manque des bordures, pour ajouter des bordures on le fait grâce à la propriété **border** de CSS :

Forme.css

```

td
{
border: 1px black solid;
}

```

Comme sélecteur, j'ai utilisé td car les bordures vont s'appliquer aux cellules.

Résultat :

Aimé	DIUMI	Mont Ngafula
Kleine	MBO	Lemba
Julie	DIKOLO	Masina
Claudia	FALASH	Lemba

Vous l'avez sans doute remarqué, il y a deux bordures entre les cellules, si vous désirez qu'il y ait une seule bordure entre deux cellules, il vous faut utiliser la propriété CSS spécifique aux tableaux, ***border-collapse***, qui signifie « coller les bordures entre elles ».

Cette propriété peut prendre deux valeurs :

- ***collapse*** : les bordures seront collées entre elles;
- ***separate*** : les bordures seront dissociées. C'est la valeur par défaut.

Pour notre exemple, on aimerait qu'il n'y ait qu'une seule bordure entre deux cellules, on doit écrire :

Forme.css

```
table
{
  border-collapse: collapse;
}

td
{
border: 1px black solid;
}
```

Résultat :

Aimé	DIUMI	Mont Ngafula
Kleine	MBO	Lemba
Julie	DIKOLO	Masina
Claudia	FALASH	Lemba

Notez que si les bordures de cellules sont séparées (`border-collapse : separate`), vous avez alors la possibilité de préciser l'espacement entre les cellules du tableau comme le montre l'exemple suivant :

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
```

```

<table>
<tr>
<td> Elombo      </td>
<td> Nestor      </td>
<td> Lemba       </td>
  </tr>
<tr>
<td> Kalombo     </td>
<td> Andy        </td>
<td> Matete      </td>
  </tr>
<tr>
<td> Mboliasa    </td>
<td> Idriss      </td>
<td> Masina      </td>
  </tr>
  <tr>
<td> Tshiela     </td>
<td> Lagrace     </td>
<td> Masina      </td>
  </tr>
</table>

</body>
</html>

```

Ajoutons une bordure :

Forme.css

```

td
{
  border: 1px black solid;
}

```

Résultat :

Elombo	Nestor	Lemba
Kalombo	Andy	Matete
Mboliasa	Idriss	Masina
Tshiela	Lagrace	Masina

Rajoutons de l'espace entre les cellules :

Forme.css

```
td
{
    border: 1px black solid;
}
table
{
    border-spacing : 10px;
}
```

Résultat :

Elombo	Nestor	Lemba
Kalombo	Andy	Matete
Mboliasa	Idriss	Masina
Tshiela	Lagrace	Masina

On peut indiquer deux valeurs différentes pour l'espacement vertical et horizontal :

Forme.css

```
td
{
    border: 1px black solid;
}
table
{
    border-spacing : 5px 20px;
}
```

Résultat :

Elombo	Nestor	Lemba
Kalombo	Andy	Matete
Mboliasa	Idriss	Masina
Tshiela	Lagrace	Masina

## IV.2 Tableaux comportant des cellules vides

Sachez qu'un tableau peut contenir des cellules vides, il suffit de ne rien mettre entre les balises `<td>` `</td>` comme le montre l'exemple suivant :

Formation.html

```
<table>
  <tr>
    <td> 12</td>
    <td> 52</td>
    <td> </td>
    <td> 7 </td>
  </tr>
  <tr>
    <td> </td>
    <td> 30 </td>
    <td> </td>
    <td> 5</td>
  </tr>
  <tr>
    <td> 12 </td>
    <td> </td>
    <td> 7 </td>
    <td> 5</td>
  </tr>
  <tr>
    <td> </td>
    <td> 4 </td>
    <td> </td>
    <td> </td>
  </tr>
</table>
```

Ajoutons les bordures, définissons la largeur et la hauteur de chaque cellule

Forme.css

```
td
{
  border: 1px black solid;
  width: 75px;
  height: 20px;
  text-align:center;
}
```

Résultat :

12	52		7
	30		5
12		7	5
	4		

Il y a une propriété CSS (`empty-cells`) qui permet d'indiquer au navigateur ce qu'il doit faire quand il rencontre une cellule vide. Elle peut prendre une des deux valeurs suivantes :

- ❖ **show** : les cellules vides sont affichées.
- ❖ **hide** : les cellules ne sont pas affichées comme le montre l'exemple suivant :

forme.css

```
td
{
  border: 1px black solid;
  width: 75px;
  height: 20px;
  text-align:center;
  empty-cells: hide;
}
```

Résultat :

12	52		7
	30		5
12		7	5
	4		

## IV.3 La ligne de l'en-tête

La ligne d'en-tête se crée avec un `<tr>` comme pour une ligne normale, mais les cellules qu'elle contient sont, cette fois, encadrées par des balises `<th>` et non pas `<td>` !

Par exemple si je veux ajouter les en-têtes Prénom, Nom et commune, j'écrirai :

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
<table>
<tr>
<th> PRENOM </th>
<th> NOM </th>
<th> COMMUNE </th>
<tr>
<td> Aimé </td>
<td> DIUMI </td>
<td> Mont Ngafula </td>
</tr>
<tr>
<td> Kleine </td>
<td> MBO </td>
<td> Lemba </td>
</tr>
<tr>
<td> Julie </td>
<td> DIKOLO </td>
<td> Masina </td>
</tr>
<tr>
<td> Claudia </td>
<td> FALASH </td>
<td> Lemba </td>
</tr>
</table>
</body>
</html>
```

Forme.css

```
table
{
  border-collapse: collapse;
}

td, th
{
  border: 1px black solid;
}
```

Résultat :

PRENOM	NOM	COMMUNE
Aimé	DIUMI	Mont Ngafula
Kleine	MBO	Lemba
Julie	DIKOLO	Masina
Claudia	FALASH	Lemba

## IV.4 Le titre du tableau

On ajoute un titre au tableau grâce à la balise **<caption>**. Cette balise se place tout au début du tableau, juste avant l'en-tête.

Pour notre exemple, si je veux ajouter le titre suivant : Membres de Wissen :

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
<table>
<caption> MEMBRES DE WISSEN </caption>
<tr>
<th> PRENOM </th>
<th> NOM </th>
<th> COMMUNE </th>
```



```

<tr>
<td> Aimé          </td>
<td> DIUMI         </td>
<td>  Mont Ngafula </td>
  </tr>
<tr>
<td> Kleine        </td>
<td> MBO           </td>
<td>  Lemba         </td>
  </tr>
<tr>
<td> Julie         </td>
<td> DIKOLO        </td>
<td>  Masina        </td>
  </tr>

  <tr>
<td>  Claudia      </td>
<td>  FALASH        </td>
<td>  Lemba         </td>
  </tr>
</table>
</body>
</html>

```

Résultat :

MEMBRES DE WISSEN		
PRENOM	NOM	COMMUNE
Aimé	DIUMI	Mont Ngafula
Kleine	MBO	Lemba
Julie	DIKOLO	Masina
Claudia	FALASH	Lemba

Sachez que vous pouvez changer la position du titre avec la propriété CSS ***caption-side*** qui peut prendre deux valeurs :

- ***top*** : le titre sera placé au-dessus du tableau, c'est la valeur par défaut ;
- ***bottom*** : le titre sera placé en dessous du tableau.

Forme.css

```
table
{
  border-collapse: collapse;
}

td, th
{
border: 1px black solid;
}

caption
{
  caption-side: bottom;
}
```

Résultat :

PRENOM	NOM	COMMUNE
Aimé	DIUMI	Mont Ngafula
Kleine	MBO	Lemba
Julie	DIKOLO	Masina
Claudia	FALASH	Lemba

MEMBRES DE WISSEN

## IV.5 Diviser un gros tableau

Si votre tableau est assez gros, vous aurez tout intérêt à le découper en plusieurs parties. Pour cela, il existe des balises HTML qui permettent de définir les trois « zones » du tableau :

- **<thead></thead>** définit l'en-tête du tableau (en haut) ;
- **<tbody></tbody>** définit le corps du tableau (au centre);
- **<tfoot></tfoot>** définit le pied du tableau (en bas).

Généralement, si on a un long tableau, dans le pied du tableau, on recopie les cellules d'en-tête. Cela permet de voir, même en bas du tableau, à quoi se rapporte chacune des colonnes.

MEMBRES DE WISSEN			<i>En-tête</i>
PRENOM	NOM	COMMUNE	
Aimé	DIUMI	Mont Ngafula	<i>Corps du tableau</i>
Kleine	MBO	Lemba	
Julie	DIKOLO	Masina	
Claudia	FALASH	Lemba	
Nestor	ELOMBO	Lemba	
Andy	KALOMBO	Mont-Ngafula	
Fanny	LOSO	Masina	
Hermans	IMBALEVA	N'Djili	
Moise	DIKOLO	Masina	
PRENOM	NOM	COMMUNE	<i>Pied du tableau</i>

Il est conseillé d'écrire les balises dans l'ordre suivant :

1. <thead>
2. <tfoot>
3. <tbody>

Dans le code, on renseigne donc d'abord la partie du haut, ensuite la partie du bas, et enfin la partie principale (<tbody>). Le navigateur se chargera d'afficher chaque élément au bon endroit.

Exemple

Pour le tableau, ci-haut, voici le code html

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>

<table>
<caption> MEMBRES DE WISSEN </caption>
<thead>
<tr>
<th> PRENOM </th>
<th> NOM </th>
<th> COMMUNE </th>
</tr>
</thead>
```

```

<tfoot>
<tr>
<th> PRENOM </th>
<th> NOM </th>
<th> COMMUNE </th>
</tr>
</tfoot>

<tbody>
<tr>
<td> Aimé </td>
<td> DIUMI </td>
<td> Mont Ngafula </td>
</tr>
<tr>
<td> Kleine </td>
<td> MBO </td>
<td> Lemba </td>
</tr>
<tr>
<td> Julie </td>
<td> DIKOLO </td>
<td> Masina </td>
</tr>
<tr>
<td> Claudia </td>
<td> FALASH </td>
<td> Lemba </td>
</tr>
<tr>
<td> Nestor </td>
<td> ELOMBO </td>
<td> Lemba </td>
</tr>
<tr>
<td> Andy </td>
<td> KALOMBO </td>
<td> Mont-Ngafula </td>
</tr>
<tr>
<td> Fanny </td>
<td> LOSO </td>
<td> Masina </td>
</tr>

```

```

<tr>
<td> Hermans                </td>
<td>  IMBALEVA              </td>
<td>  N'Djili                </td>
</tr>

<tr>
<td> Moise                  </td>
<td>  DIKOLO                 </td>
<td>  Masina                 </td>
</tr>
</tbody>
</table>

</body>
</html>

```

## IV.6 Fusion des cellules d'un tableau

Il arrive de fois qu'on ait besoin de fusionner certaines cellules du tableau comme le tableau suivant :

TITRE DU LIVRE	AUTEUR	DOMAINE	
		MATHS	INFORMATIQUE
Logique mathématique	Aimé DIUMI	Oui	non
Equations et inéquations	Kleine MBO	oui	non
Création d'un site web avec Html et Css	Aimé DIUMI	non	oui
Système de numération		Les deux	

Certaines cellules sont fusionnées notamment celles de l'auteur du livre Création d'un site web avec Html et Css ainsi que celui de Système de numération.

On peut effectuer deux types de fusion :

- *La fusion de colonnes* : La fusion s'effectue horizontalement. Le cas des cellules du domaine du livre Système de numération. Pour ce type de fusion, on utilise l'attribut **colspan**.
- *La fusion de lignes* : On groupe les lignes, la fusion s'effectuera verticalement. Le cas des cellules contenant l'auteur des livres Création d'un site web et Système de numération. On utilisera l'attribut **rowspan**.

On ajoute l'attribut à la balise **td** ou **th**. Comme valeur à donner à l'attribut, on indique le nombre de cellules à fusionner entre elles.

## IV.6.1 Fusion de colonnes

Expliquons ce mode de fusion avec un exemple simple

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
<table>
<caption> LIVRES PUBLIES </caption>
<tr>
<th > TITRE DU LIVRE </th>
<th > AUTEUR </th>
<th> MATHS </th>
<th> INFORMATIQUE </th>
</tr>
<tr>
<td> Logique mathématique </td>
<td> Aimé DIUMI </td>
<td> Oui </td>
<td > non </td>
  </tr>
<tr>
<td> Equations et inéquations </td>
<td> Kleine MBO </td>
<td> oui </td>
<td > non </td>
  </tr>
<tr>
<td> Création d'un site web avec Html et Css </td>
<td > Aimé DIUMI </td>
<td> non </td>
<td > oui </td>
  </tr>
  <tr>
<td> Système de numération </td>
<td> Aimé DIUMI </td>
<td > oui </td>
<td> oui </td>
  </tr>
</table>
</body>
</html>
```

Résultat :

TITRE DU LIVRE	AUTEUR	MATHS	INFORMATIQUE
Logique mathématique	Aimé DIUMI	Oui	non
Equations et inéquations	Kleine MBO	oui	non
Création d'un site web avec Html et Css	Aimé DIUMI	non	oui
Système de numération	Aimé DIUMI	oui	oui

Comme le livre système de numération est utilisé pour les mathématiques et l'informatique, on peut fusionner les deux cellules contenant « oui » pour écrire « Les deux » par exemple.

Comme un tableau en html se construit ligne par ligne, ce qui nous intéresse ici c'est la dernière ligne et plus précisément les deux dernières cellules de la dernière ligne : on ajoute l'attribut colspan à la balise td contenant « oui » et comme valeur, on mettra 2 car nous voulons fusionner deux cellules. Après l'ajout de cet attribut, la dernière ligne n'aura plus quatre cellules (td) mais trois car deux cellules sont fusionnées.

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>
<table>
<caption> LIVRES PUBLIES </caption>
<tr>
<th > TITRE DU LIVRE </th>
<th > AUTEUR </th>
<th> MATHS </th>
<th> INFORMATIQUE </th>
</tr>
<tr>
<td> Logique mathématique </td>
<td> Aimé DIUMI </td>
<td> Oui </td>
<td > non </td>
</tr>
<tr>
<td> Equations et inéquations </td>
<td> Kleine MBO </td>
```

```

<td> oui          </td>
<td > non        </td>
  </tr>
<tr>
<td> Création d'un site web avec Html et Css          </td>
<td > Aimé DIUMI          </td>
<td> non          </td>
<td > oui          </td>
  </tr>
  <tr>
<td> Système de numération          </td>
<td> Aimé DIUMI </td>
<td colspan="2"> Les deux </td>
  </tr>
</table>
</body>
</html>

```

Le fait d'ajouter **colspan="2"** signifie que cette cellule prend la place de deux cellules.

Résultat :

LIVRES PUBLIES			
TITRE DU LIVRE	AUTEUR	MATHS	INFORMATIQUE
Logique mathématique	Aimé DIUMI	Oui	non
Equations et inéquations	Kleine MBO	oui	non
Création d'un site web avec Html et Css	Aimé DIUMI	non	oui
Système de numération	Aimé DIUMI	Les deux	

Jouons un peu avec la fusion : fusionnons les colonnes « Auteur », « Maths » et « Informatique » du livre Logique mathématique en « fusion de 3 cellules ». Pour la ligne du livre « Equations et inéquations », fusionnons les 4 cellules en « fusion de 4 cellules » :

Formation.html

```

<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>

```



```

<table>
<caption> LIVRES PUBLIES </caption>

<tr>
<th > TITRE DU LIVRE </th>
<th > AUTEUR </th>
<th> MATHS </th>
<th> INFORMATIQUE </th>
</tr>

<tr>
<td> Logique mathématique </td>
<td colspan="3"> Fusion de 3 cellules </td>
</tr>

<tr>
<td colspan="4"> Fusion de 4 cellules </td>
</tr>

<tr>
<td> Création d'un site web avec Html et Css </td>
<td > Aimé DIUMI </td>
<td> non </td>
<td > oui </td>
</tr>

<tr>
<td> Système de numération </td>
<td> Aimé DIUMI </td>
<td colspan="2"> Les deux </td>
</tr>

</table>

</body>
</html>

```

Résultat :

LIVRES PUBLIES

TITRE DU LIVRE	AUTEUR	MATHS	INFORMATIQUE
Logique mathématique	Fusion de 3 cellules		
Fusion de 4 cellules			
Création d'un site web avec Html et Css	Aimé DIUMI	non	oui
Système de numération	Aimé DIUMI	Les deux	

## IV.6.2 Fusion de lignes

Reprenons le tableau initial avant la fusion de colonnes

LIVRES PUBLIES

TITRE DU LIVRE	AUTEUR	MATHS	INFORMATIQUE
Logique mathématique	Aimé DIUMI	Oui	non
Equations et inéquations	Kleine MBO	oui	non
Création d'un site web avec Html et Css	Aimé DIUMI	non	oui
Système de numération	Aimé DIUMI	oui	oui

C'est le même auteur qui a écrit les livres « Création d'un site web et Système de numération », alors pourquoi ne pas fusionner les deux cellules contenant le nom de l'auteur de ces deux livres ? Il s'agit là d'une fusion de lignes, on utilise l'attribut **rowspan**.

Ce sont les deux dernières lignes qui nous concernent. L'auteur du livre « Création d'un site web... » est la deuxième colonne de l'avant dernière ligne, c'est là qu'on va ajouter l'attribut **rowspan="2"**. Pour la dernière ligne, la deuxième colonne (auteur) disparaîtra car elle est fusionnée avec la deuxième colonne de la ligne précédente.

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>

<table>
<caption> LIVRES PUBLIES </caption>

<tr>
<th > TITRE DU LIVRE </th>
<th > AUTEUR </th>
<th> MATHS </th>
```

```

<th> INFORMATIQUE </th>
</tr>

<tr>
<td> Logique mathématique </td>
<td> Aimé DIUMI </td>
<td> Oui </td>
<td > non </td>
</tr>

<tr>
<td> Equations et inéquations </td>
<td> Kleine MBO </td>
<td> oui </td>
<td > non </td>
</tr>

<tr>
<td> Création d'un site web avec Html et Css </td>
<td rowspan="2"> Aimé DIUMI </td>
<td> non </td>
<td > oui </td>
</tr>

<tr>
<td> Système de numération </td>
<td > oui </td>
<td> oui </td>
</tr>
</table>
</body>
</html>

```

Résultat :

LIVRES PUBLIES			
TITRE DU LIVRE	AUTEUR	MATHS	INFORMATIQUE
Logique mathématique	Aimé DIUMI	Oui	non
Equations et inéquations	Kleine MBO	oui	non
Création d'un site web avec Html et Css	Aimé DIUMI	non	oui
Système de numération		oui	oui

Fusionnons à présent toute la colonne de l'Informatique : On ajoute `rowspan="4"` à la colonne « Informatique » de la première ligne, ensuite pour les autres lignes, on supprime la colonne « Informatique » :

## Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel=stylesheet href="forme.css" />
  </head>
  <body>

<table>
<caption> LIVRES PUBLIES </caption>

<tr>
<th > TITRE DU LIVRE </th>
<th > AUTEUR </th>
<th> MATHS </th>
<th> INFORMATIQUE </th>
</tr>

<tr>
<td> Logique mathématique </td>
<td> Aimé DIUMI </td>
<td> Oui </td>
<td rowspan="4"> non </td>
</tr>
<tr>
<td> Equations et inéquations </td>
<td> Kleine MBO </td>
<td> oui </td>

</tr>
<tr>
<td> Création d'un site web avec Html et Css </td>
<td rowspan="2"> Aimé DIUMI </td>
<td> non </td>

</tr>

<tr>
<td> Système de numération </td>

<td > oui </td>

</tr>
```

```
</table>
```

```
</body>
```

```
</html>
```

Résultat :

LIVRES PUBLIES			
TITRE DU LIVRE	AUTEUR	MATHS	INFORMATIQUE
Logique mathématique	Aimé DIUMI	Oui	non
Equations et inéquations	Kleine MBO	oui	
Création d'un site web avec Html et Css	Aimé DIUMI	non	
Système de numération		oui	

# V. LA NUMEROTATION AUTOMATIQUE AVEC CSS

Il est possible avec CSS de générer une notation automatique pour les sections et sous sections. Pour cela, nous allons utiliser deux propriétés :

**Counter-reset**, elle crée ou réinitialise un compteur donné. Elle s'utilise comme ceci :

```
counter-reset: nom_compteur;
```

**counter-increment** : incrémente le compteur désigné, elle s'utilise comme suit :

```
counter-increment: nom_compteur valeur_incrémementation ;
```

Le pas d'incrémementation est par défaut de 1. Mais une autre valeur peut être spécifiée. Prenons l'exemple suivant :

Supposons que notre document est composé des titres de niveau 1 (h1) pour les chapitres et les titres de niveau 2 (h2) pour les sections : un chapitre peut avoir plusieurs sections. Nous aimerions que chaque fois que nous insérerions un titre de niveau 1 (chapitre) que CSS puisse le numéroter automatiquement (chapitre 3 par exemple) selon la place qu'il occupe dans le document.

Le premier titre h1 aura pour numéro 1, le deuxième aura pour le numéro 2 et ainsi de suite sauf si on réinitialise le compteur pour recommencer à compter à partir de zéro.

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel="stylesheet" href="forme.css" />
  </head>
  <body>
    <section >
      <h1>LES VARIABLES </h1>
      <h2>Déclaration explicite d'une variable </h2>
      <p> Paragraphe </p>
      <p> Paragraphe </p>
      <h2>Déclaration implicite d'une variable </h2>
      <h2>Portée des variables </h2>
      <p> Paragraphe </p>
      <h1>TYPES DE DONNEES </h1>
      <h2>Types de données numériques </h2>
      <h2>Types de données caractères </h2>
```

```

<h2>Types de données divers </h2>

<h1>LES CONSTANTES </h1>
<h1>CONVERSION DE TYPES </h1>
  <p> Paragraphe </p>
  <p> Paragraphe </p>
<h1>OPERATEURS ET EXPRESSIONS </h1>
  <h2> Opérateurs </h2>
  <h2> Expressions </h2>
<h1>LES INSTRUCTIONS DE BASE EN MODE CONSOLE </h1>
  <h2>Ecriture sur l'écran </h2>
  <h2>Lecture de données tapées au clavier </h2>
</section >
</body>
</html>

```

Nous devons créer deux compteurs : un pour les chapitres (h1) et l'autre pour les sections (h2). Vu que les sections (h2) sont les sous éléments des chapitres, la création ou réinitialisation du compteur des sections doit se faire dans h1 et celui des chapitres dans la balise `<section >` comme ci :

Forme.css

```

section
{
counter-reset: chap;
}

h1
{
  counter-reset: section;
}

```

« chap » est le nom que j'ai donné au compteur des chapitres et « section », celui du compteur des sections (h2).

Après cela, il faut incrémenter les compteurs pour chaque h1, de même pour h2, le code s'ajoute :

Forme.css

```

section
{
counter-reset: chap;
}

h1

```

```

{
  counter-reset: section;
  counter-increment: chap;
}
h2
{
  counter-increment: section;
}

```

Enfin, nous voudrions que chaque chapitre (h1) s'affiche comme ceci :

CHAPITRE 1 : LES VARIABLES par exemple. Pour y arriver, on ajoute ceci :

Forme.css

```

section
{
  counter-reset: chap;
}

h1
{
  counter-reset: section;
  counter-increment: chap;
}

h2
{
  counter-increment: section;
}

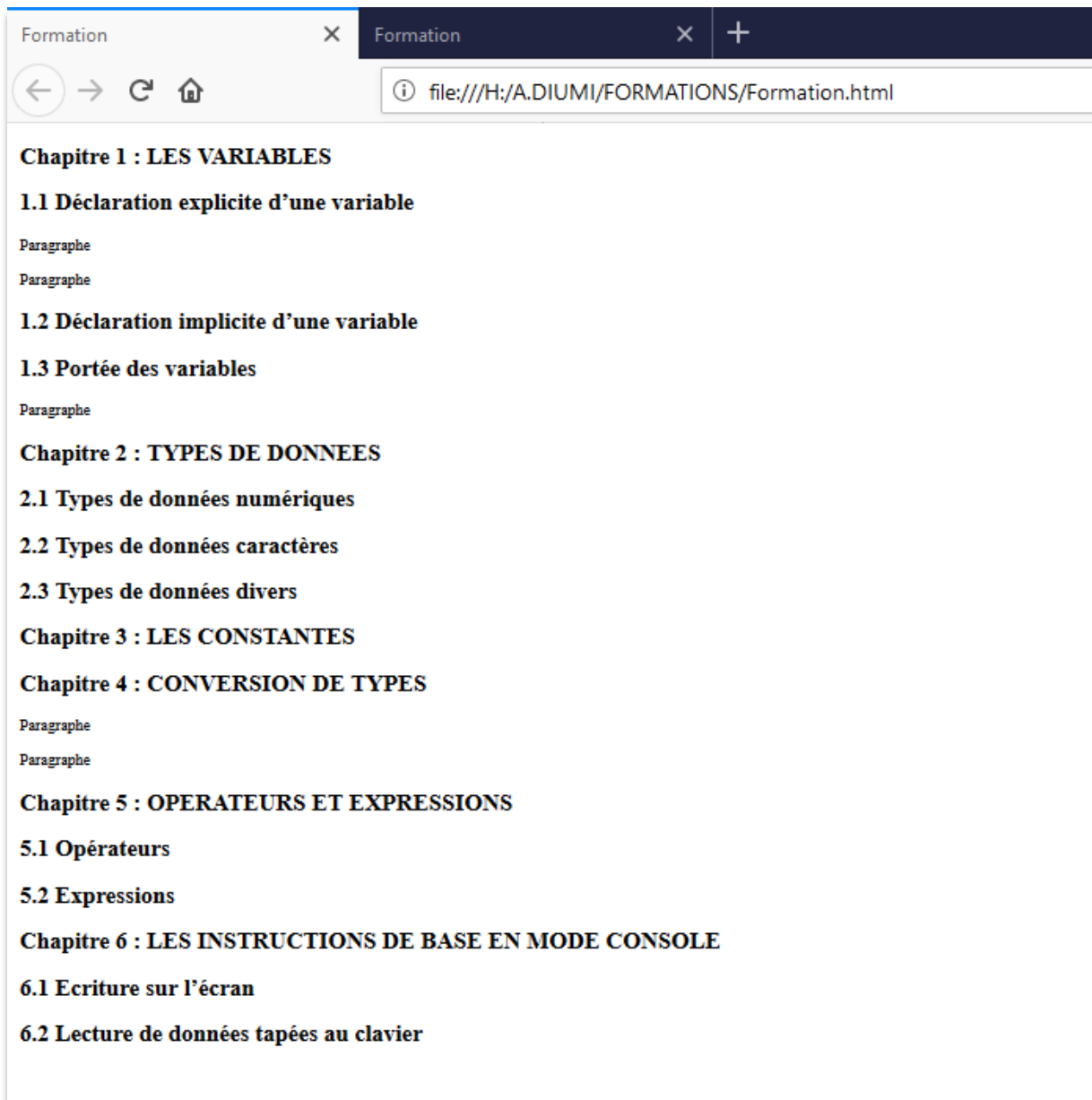
h1:before
{
  content: "CHAPITRE " counter(chap) " : ";
}

h2:before
{
  content: counter(chap) "." counter(section) " ";
}

```



Résultat :



The image shows a web browser window with two tabs labeled 'Formation'. The address bar contains the file path: file:///H:/A.DIUMI/FORMATIONS/Formation.html. The main content area displays a table of contents with the following structure:

- Chapitre 1 : LES VARIABLES**
  - 1.1 Déclaration explicite d'une variable
    - Paragraphe
    - Paragraphe
  - 1.2 Déclaration implicite d'une variable
  - 1.3 Portée des variables
    - Paragraphe
- Chapitre 2 : TYPES DE DONNEES**
  - 2.1 Types de données numériques
  - 2.2 Types de données caractères
  - 2.3 Types de données divers
- Chapitre 3 : LES CONSTANTES**
- Chapitre 4 : CONVERSION DE TYPES**
  - Paragraphe
  - Paragraphe
- Chapitre 5 : OPERATEURS ET EXPRESSIONS**
  - 5.1 Opérateurs
  - 5.2 Expressions
- Chapitre 6 : LES INSTRUCTIONS DE BASE EN MODE CONSOLE**
  - 6.1 Ecriture sur l'écran
  - 6.2 Lecture de données tapées au clavier

## VI. COMPLEMENTS

### VI.1 LES CARACTÈRES SPECIAUX EN HTML

Pour insérer des caractères spéciaux en HTML, il faut utiliser un codage particulier. Ce codage consiste à saisir une combinaison précédée du caractère & et terminée par un point-virgule (;).

Voici le code HTML des quelques caractères spéciaux :

Caractère	Code HTML
"	&quot ;
&	&amp ;
€	&euro ;
<	&lt ;
>	&gt ;
œ	&oelig ;
Space	&nbsp ;
¬	&not ;
–	&masr ;
±	&plusmn ;
´	&acute ;
μ	&micro ;
·	&middot ;
<sup>1</sup>	&sup1 ;
»	&raquo ;
½	&frac12 ;
¿	&iquest ;
Á	&Aacute ;
Ã	&Atilde ;
Å	&Aring ;
Ç	&Ccedil ;
É	&Eacute ;
Ë	&Euml ;
Í	&Iacute ;
Ï	&Iuml ;
Ñ	&Ntilde ;
Ó	&Oacute ;
Õ	&tilde ;
×	&times ;
Û	&Ugrave ;
Û	&Ucirc ;
Ý	&Yacute ;
ß	&szlig ;
á	&aacute ;
ã	&atilde ;

Caractère	Code HTML
£	&pound ;
¥	&yen ;
¤	&curren ;
§	&sect ;
¨	&uml ;
©	&copy ;
«	&laquo ;
®	&reg ;
°	&deg ;
<sup>2</sup>	&sup2 ;
<sup>3</sup>	&sup3 ;
¶	&para ;
¸	&cedil ;
º	&ordm ;
¼	&frac14 ;
¾	&frac34 ;
À	&Agrave ;
Â	&Acirc ;
Ä	&Auml ;
Æ	&Aelig ;
È	&Egrave ;
Ê	&Ecirc ;
Ì	&Igrave ;
Î	&Icirc ;
Ð	&eth ;
Ò	&Ograve ;
Ô	&Ocir ;
Ö	&Ouml ;
Ø	&Oslash ;
Ú	&Uacute ;
Ü	&Uuml ;
Þ	&thorn ;
à	&agrave ;
â	&acirc ;
ä	&auml ;

å	&aring
ç	&ccedil ;
é	&eacute ;
ë	&euml ;
í	&iacute ;
ï	&iuml ;
ñ	&ntilde ;
ó	&oacute ;
õ	&otilde ;
÷	&divide ;
ù	&ugrave ;
û	&ucirc ;
ý	&yacute ;
ÿ	&yuml ;

æ	&aelig ;
è	&egrave ;
ê	&ecirc ;
ì	&igrave ;
î	&icirc ;
ð	&eth ;
ò	&ograve ;
ô	&ocirc ;
ö	&ouml ;
ø	&oslash ;
ú	&uacute ;
ü	&uuml ;
þ	&thorn ;

## VI.2 INSERTION DU CODE SOURCE DANS UNE PAGE WEB

### VI.2.1 Insertion du code source

Il peut arriver qu'on ait besoin d'afficher le code source d'un langage quelconque (Visual basic, Java, Html, Ccssn dans la page web. Dans ce cas, on écrit le code source à insérer dans la page entre les balises `<code>` `</code>` et elles même comprises entre les balises `<pre>` `</pre>` comme l'illustre la figure suivante :

```
<pre>
    <code>

<!-- on insère le code source ici-->

    </code>
</pre>
```

Exemple :

On affiche dans notre page web les codes sources d'un petit programme en Visual basic

Formation.html

```
<p> Voici les codes sources du programme en Visual basic qui détermine si
un nombre est premier ou pas
</p>
<pre>
```

```

<code>
Module Module1

Sub Main()
    'ce programme permet de déterminer si un nombre est premier ou
pas
    Dim n, i As Integer
    Dim test As Boolean = True
    Console.WriteLine("Entrez un nombre entier positif")
5:   n = CInt(Console.ReadLine())
    If n < 0 Then
z")   Console.WriteLine("Le nombre doit etre positif, recommence

        GoTo 5
    End If
    For i = 2 To n - 1 Step 1
        If n Mod i = 0 Then
            test = False
        End If
    Next

    If test = True Then
        Console.WriteLine("Le nombre est premier")
    Else
        Console.WriteLine("Le nombre n'est pas premier")
    End If

    Console.ReadLine()
End Sub

End Module
</code>
</pre>

```

Résultat :



Voici les codes sources du programme en Visual basic qui détermine si un nombre est premier ou pas

```
Module Module1

Sub Main()
    'ce programme permet de déterminer si un nombre est premier ou pas
    Dim n, i As Integer
    Dim test As Boolean = True
    Console.WriteLine("Entrez un nombre entier positif")
5:   n = CInt(Console.ReadLine())
    If n < 0 Then
        Console.WriteLine("Le nombre doit etre positif, recommencez")
        GoTo 5
    End If
    For i = 2 To n - 1 Step 1
        If n Mod i = 0 Then
            test = False
        End If
    Next
    If test = True Then
        Console.WriteLine("Le nombre est premier")
    Else
        Console.WriteLine("Le nombre n'est pas premier")
    End If
    Console.ReadLine()
End Sub

End Module
```

Et voilà le code source a été inséré, mais il nous reste à régler une chose : la coloration syntaxique du code source.

## VI.2.2 Coloration syntaxique du code source

Pour effectuer la coloration syntaxique du code source dans une page web, nous avons plusieurs solutions, on peut citer : [GeSHi](#), [SyntaxHighlighter](#), [Pygments](#), [Prism](#), etc. Dans notre cas, nous allons utiliser Prism.

Prism est composé de deux fichiers prism.css et prism.js. Ces deux fichiers sont à télécharger dans [Page de téléchargement de Prism](#) :

Select your compression level, as well as the languages and plugins you need.

1

Compression level:

- Development version
- Minified version

### Core

- Core 34.69KB

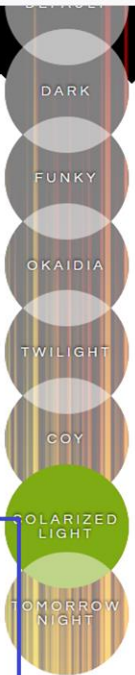
### Themes

- 2
- Default 2.28KB
  - Dark 2.02KB
  - Funky 2.44KB
  - Okaidia ocodia 1.77KB
  - Twilight remybach 4.02KB
  - Coy tshedor 3.93KB
  - Solarized Light hectormatos2011 2.54KB
  - Tomorrow Night Rosey 1.72KB

### Languages

- 3
- Select/unselect all
  - Markup + HTML + XML + SVG + MathML + SSMML + Atom + RSS 3.32KB
  - CSS 2.02KB
  - Brainfuck Golmote 0.29KB
  - BrightScript RunDevelopment 1.27KB
  - Bro wayward710 1.32KB
  - BSL (1C:Enterprise) +
  - DataWeave machaval 1.36KB
  - DAX peterbud 3.56KB
  - Dhall RunDevelopment 2.27KB
  - Diff uranusjr 1.33KB
  - Django/Jinja2 romanvm 1.69KB
  - GameMaker Language LiarOnce 8.04KB
  - G-code RunDevelopment 0.27KB
  - Git Igraudel 1.61KB
  - GLSL Golmote 1.08KB
  - Go arnehormann 0.76KB
  - GraphQL Golmote 1.2KB
  - Groovy robfletcher 2.27KB
  - JS stack trace sbri 0.95KB
  - JS Templates RunDevelopment 9.53KB
  - Julia cdagnino 1.76KB
  - Keyman mcdurdin 1.16KB
  - Kotlin Kotlin Script
  - Mizar Golmote 1.07KB
  - MongoDB airs0urce 4.01KB
  - Monkey Golmote 0.99KB
  - MoonScript RunDevelopment 2.33KB
  - N1QL TMWilds 3.31KB
  - N4JS bsmith-n4 0.78KB
  - PHP milesj 8.2KB
  - PHPDoc RunDevelopment 0.8KB
  - PHP Extras milesj 0.33KB
  - PL/SQL Golmote 1.94KB
  - PowerQuery
  - PromQL arendjr 2.4KB
  - .properties Golmote 0.3KB
  - Protocol Buffers justboris 1.19KB
  - Pug Golmote 4.45KB
  - Squirrel/Golmote RunDevelopment 2.34KB
  - Solidity (Ethereum) glachaud 1.12KB
  - Solution file RunDevelopment 1.06KB
  - T4 Text Templates (VB) RunDevelopment 0.08KB
  - TAP isaacs 0.44KB
  - Tcl PeterChaplin 1.65KB
  - Template Toolkit 2 gflahr 1.5KB
  - Textile Golmote 7.17KB
  - Träskämaxi 2.5KB
  - XML doc (.net) RunDevelopment 0.76KB
  - Xojo (REALbasic) Golmote 0.82KB
  - XQuery Golmote 6.01KB
  - YAML bases 3.11KB

DOWNLOAD JS | DOWNLOAD CSS



Handcrafted with ♥, by [Lea Verou](#), [Golmote](#), [James DiGioia](#), [Michael Schmidt](#) & all these awesome people

- [Home](#)
- [Download](#)
- [FAQ](#)
- [Test drive](#)
- [API docs](#)
- [Fork Prism on GitHub](#)
- [Follow Prism on Twitter](#)



1 : Vous avez le choix entre *Development version* et *Minified version*.

Avec *Minified version*, vous aurez un fichier réduit (optimisé) mais presque illisible et vous n'aurez pas la possibilité de modifier ou paramétrer le code source de ce fichier. Avec *Development version*, vous aurez un fichier non optimisé mais lisible avec la possibilité de modifier et paramétrer selon vos désirs.

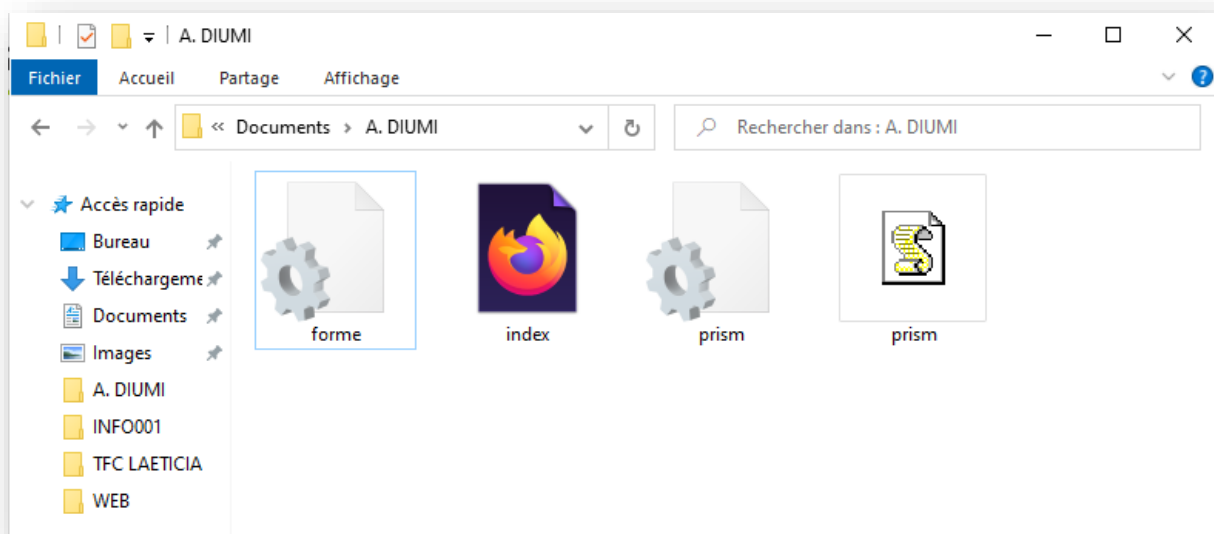
2 : Choisissez le thème de votre choix parmi les choix proposés

3 : Choisissez les langages de votre choix : ce sont les langages que Prism reconnaîtra pour la coloration syntaxique.

4 : Choisissez les plugins, un plugin est une fonctionnalité supplémentaire à ajouter à Prism, la numérotation de lignes de code par exemple.

Enfin, cliquez sur **DOWNLOAD JS** pour télécharger `prism.js` et sur **DOWNLOAD CSS** pour télécharger `prism.css`

De préférence, placez les deux fichiers dans le même dossier que votre fichier `html` comme l'illustre la figure suivante :



Après cela, il faut faire appel aux fichiers dans votre fichier `html`, le fichier `prism.css` sera inséré dans l'en-tête (`head`) comme l'illustre la figure suivante :

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title> Formation en HTML et CSS</title>
6     <link rel="stylesheet" href="forme.css" />
7     <link rel="stylesheet" href="prism.css" />
8   </head>
```

Le deuxième fichier sera inséré juste avant la fermeture de body :

```
<script src="prism.js"></script>
</body>
</html>
```

Pour spécifier le langage, ajoutez l'attribut class à la balise code comme ceci :

```
<code class="language-x">
```

Où x représente un langage donné, actuellement Prism supporte 243 langages, on écrira par exemple :

Class="language-csharp" pour le C#

Class="language-java" pour le Java

Class="language-html" pour le html

Class="language-javascript" ou language-js pour le JavaScript

Class="language-css" pour le CSS

Class="language-sql" pour le SQL

Class="language-visual-basic" ou vb ou vba ou VB.Net pour le Visual basic

Pour le programme visual basic ci-dessus, on aura :

Formation.html

```
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title> Formation en HTML et CSS</title>
    <link rel="stylesheet" href="forme.css" />
    <link href="prism.css" rel="stylesheet"/>
  </head>
  <body>
<p> Voici les codes sources du programme en Visual basic qui détermine si
  un nombre est premier ou pas
</p>
  <pre>
    <code class="language-VB.Net">
      Module Module1
      Sub Main()
        'ce programme permet de déterminer si un nombre est premier ou
pas
```



```

        Dim n, i As Integer
        Dim test As Boolean = True
        Console.WriteLine("Entrez un nombre entier positif")
5:      n = CInt(Console.ReadLine())
        If n < 0 Then
            Console.WriteLine("Le nombre doit etre positif, recommence
z")
            GoTo 5
        End If
        For i = 2 To n - 1 Step 1
            If n Mod i = 0 Then
                test = False
            End If
        Next

        If test = True Then
            Console.WriteLine("Le nombre est premier")
        Else
            Console.WriteLine("Le nombre n'est pas premier")
        End If
        Console.ReadLine()
    End Sub
End Module
</code>
</pre>
<pre>
    <script src="prism.js"></script>
</body>
</html>

```

Résultat :

Voici les codes sources du programme en Visual basic qui détermine si un nombre est premier ou pas

```
Module Module1

Sub Main()
    'ce programme permet de déterminer si un nombre est premier ou pas
    Dim n, i As Integer
    Dim test As Boolean = True
    Console.WriteLine("Entrez un nombre entier positif")
5:   n = CInt(Console.ReadLine())
    If n < 0 Then
        Console.WriteLine("Le nombre doit etre positif, recommencez")
        GoTo 5
    End If
    For i = 2 To n - 1 Step 1
        If n Mod i = 0 Then
            test = False
        End If
    Next

    If test = True Then
        Console.WriteLine("Le nombre est premier")
        Console.WriteLine("Le nombre n est pas premier")
    End If

    --- ---

End Module

End Module
```

## VI.2.3 Quelques plugins de Prism

### a) Line Numbers

Permet d'ajouter le numéro de ligne au début des lignes du code source. Pour activer cela, après avoir il suffit d'ajouter `class="line-numbers"` à la balise `<pre>` ou à l'un de ses parents.

Exemple

Formation.html

```
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title> Formation en HTML et CSS</title>
    <link href="prism.css" rel="stylesheet" />
  </head>
  <body>
    <p> Un petit programme écrit en C</p>

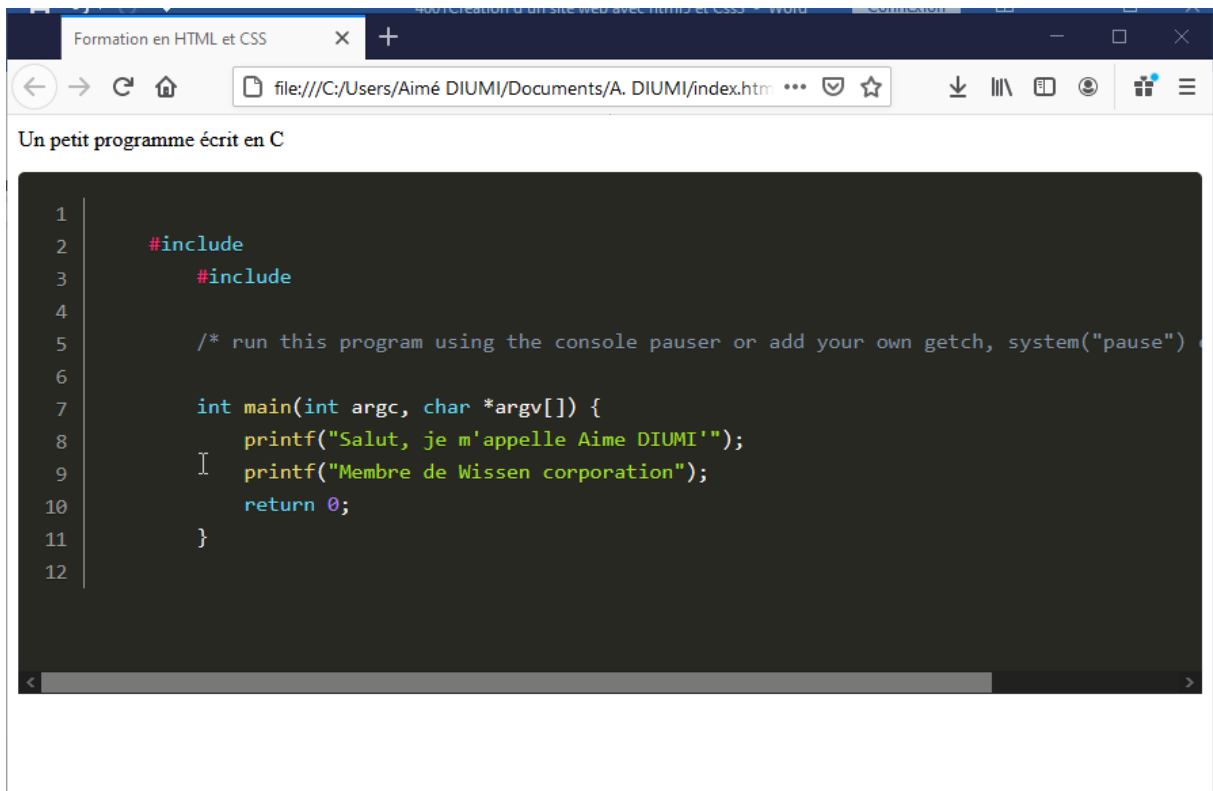
    <pre class="line-numbers">
<code class="language-C">
  #include <stdio.h>
    #include <stdlib.h>

    /* run this program using the console pauser or add your own getch, sy
stem("pause") or input loop */

    int main(int argc, char *argv[]) {
      printf("Salut, je m'appelle Aime DIUMI");
      printf("Membre de Wissen corporation");
      return 0;
    }
</code>
  </pre>

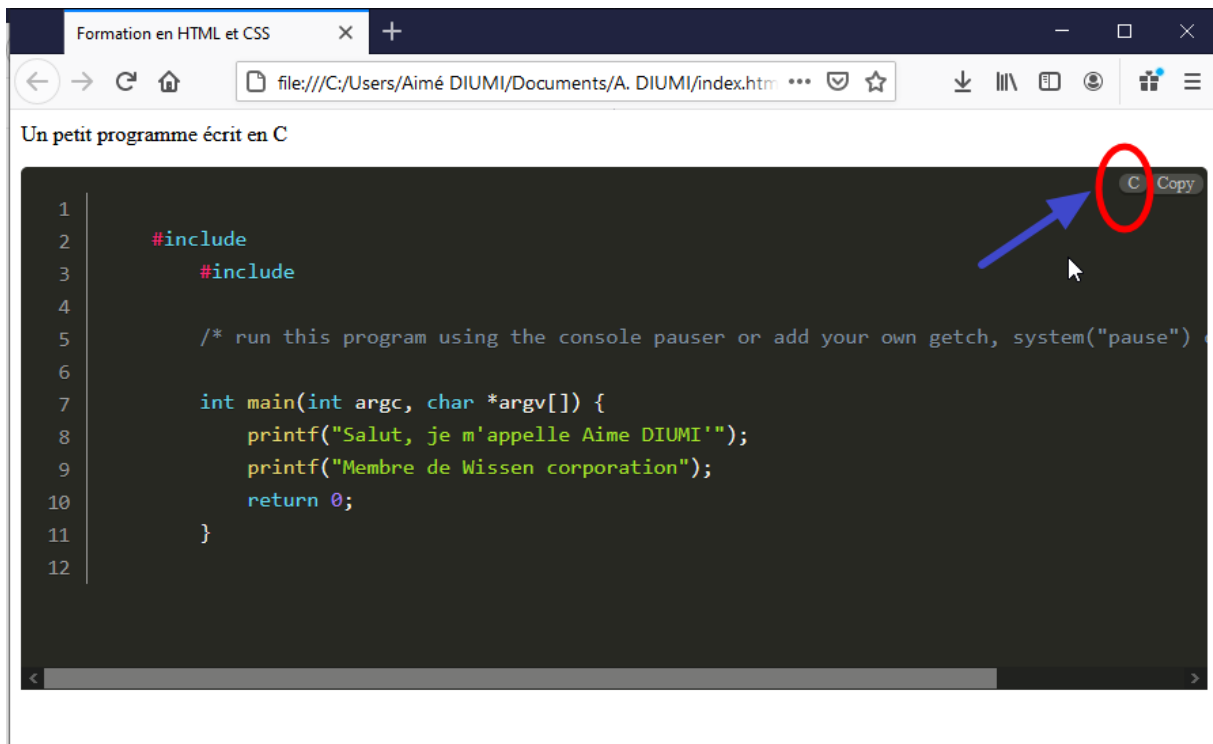
<script src="prism.js"></script>
  </body>
</html>
```

Résultat :



## b) Show language

Affiche la langue en surbrillance dans des blocs de code comme le montre la figure suivante :



### c) Copy to Clipboard Button

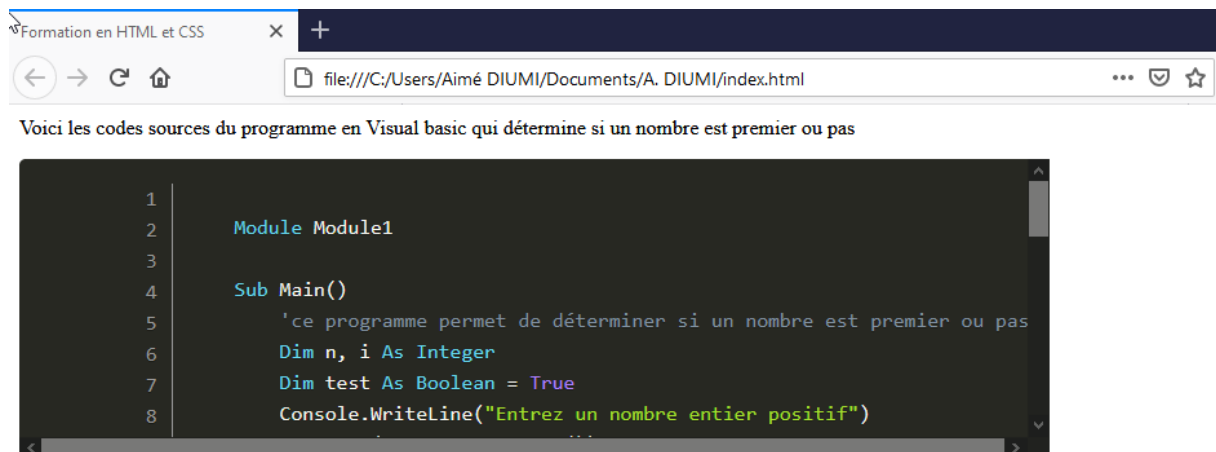
Permet d'ajouter un bouton qui copie le bloc de code si le visiteur clique dessus. Dans la figure ci-dessus, ce bouton vient juste après le nom du langage.

Si vous voulez que le code inséré ait une dimension bien précise, cela se fait grâce au CSS. Pour illustrer cela, considérez l'exemple sur le code Visual basic, il suffit de définir la largeur et la hauteur du bloc :

Forme.css

```
pre
{
    width:60%;
    height: 200px;;
}
```

Résultat :

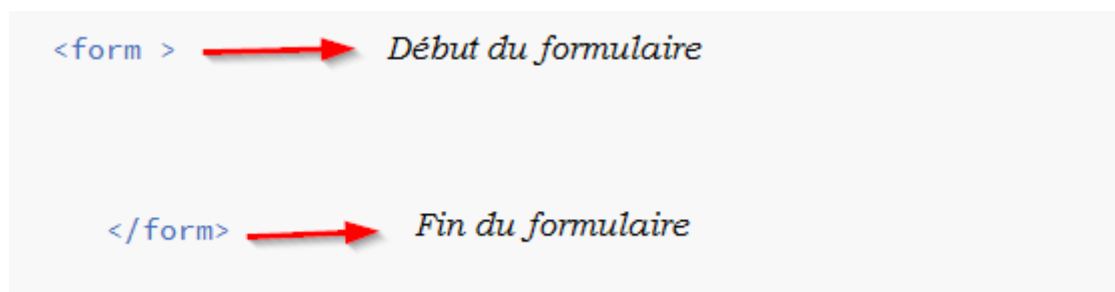


## VII. LES FORMULAIRES

On peut ajouter des formulaires dans nos pages HTML pour interagir avec nos visiteurs en les invitant à renseigner des informations : insérer des commentaires, sélectionner des options, valider un bouton etc.

Sachez cependant que les formulaires représentent les limites du langage HTML car il faut ensuite pouvoir analyser les informations que le visiteur a saisies... et cela ne peut pas se faire en langage HTML. Comme nous allons le voir, le traitement des résultats doit s'effectuer dans un autre langage, par exemple le PHP ou JavaScript.

Pour créer un formulaire, on utilise la balise `<form>` `</form>` qui est la balise principale du formulaire.



Pour indiquer comment les informations saisies par l'utilisateur seront envoyées ainsi que comment traiter ces informations (la page qui s'occupera du traitement des informations), vous devez ajouter deux attributs à la balise `<form>` :

- **method** : cet attribut indique par quel moyen les données vont être envoyées. Il existe deux solutions pour envoyer des données sur le Web :
  - **method="get"** : avec cette méthode, les informations seront envoyées dans l'adresse de la page (<http://...>) et on ne peut pas envoyer plus de 255 caractères.
  - **method="post"** : avec cette méthode, Les données saisies dans le formulaire ne transitent pas par la barre d'adresse. Elle permet d'envoyer un grand nombre d'informations.
- **action** : Lorsque vous donnez l'ordre au navigateur de transmettre les données du formulaire, il a besoin de connaître l'action qu'il doit effectuer. Cette action peut être :
  - soit l'adresse d'un programme de traitement des données situé sur le serveur, PHP par exemple.

Par exemple, `<form action = "http://www.serveur/traitement.php" >`

- soit une adresse de courrier électronique pour récupérer simplement les données.

Exemple `<form action= "mailto : votre email" >`

- Soit, lorsque les données du formulaire sont traitées en interne, cote client par Javascript par exemple, l'attribut action restera vide comme ceci :

`<form action="" >`

Donc la balise form doit être comme ceci si on retient le premier cas :

```
<form method="post" action="Traitement.php">
</form>
```

Pour method, j'ai choisi post et pour action, j'ai mis une page fictive que j'ai appelée Traitement.php, c'est-à-dire toutes les informations saisies dans le formulaire seront traitées par cette page (Traitement.php) et elle sera appelée lorsque le visiteur cliquera sur le bouton d'envoi du formulaire.

## VII.1 Zone de texte monoligne

Voici à quoi ressemble une zone de texte monoligne :

Nom:

Pour ajouter une zone de texte monoligne, on utilise la balise **input**, à la quelle on ajoute l'attribut **type= "text"** .

```
<input type="text" />
```

Etant donné que les informations saisies dans une zone de texte seront traités par un autre langage et que dans une page web, on peut avoir plusieurs zones de texte, il faut donner un nom à chaque zone de texte pour pouvoir l'identifier afin de reconnaître d'où proviennent les informations.

Pour donner un nom à un élément d'un formulaire (une zone de texte par exemple), on utilise l'attribut **name**. Pour notre exemple, on peut appeler cette zone de texte, nom\_utilisateur :

```
<input type="text" name="nom_utilisateur" />
```

Résultat :

## Les libellés

En ajoutant une zone de texte dans notre page, il sera avantageux de donner une indication au visiteur sur le type d'informations attendu dans la zone. On doit ajouter un libellé et cela se fait grâce à la balise label.

Formation.html

```
<form method="post" action="Traitement.php">
  <label> Nom: </label>
  <input type="text" name="nom_utilisateur" />
</form>
```

Résultat :



Nom:

Il faut ensuite lier le libellé avec la zone de texte, pour que lorsque l'utilisateur clique sur le libellé, le curseur se place automatiquement dans la zone de texte. Pour réaliser ce lien, il faut :

- Ajouter un attribut id à la zone de texte
- Ensuite, ajouter un attribut for ayant la même valeur que l'attribut id de la zone de texte dans le libellé (label).

Formation.html

```
<form method="post" action="Traitement.php">
  <label for="nom"> Nom: </label>
  <input type="text" name="nom_utilisateur" id="nom" />
  </form>
```

## Quelques attributs supplémentaires

- ❖ L'attribut **size** permet d'agrandir la zone de texte, en réalité, cet attribut indique le nombre de caractères visibles dans la zone et par conséquent, la largeur de la zone de texte, sa valeur par défaut est de 20.

Exemple

Formation.html

```
<form method="post" action="Traitement.php">
  <label for="nom"> Nom: </label>
  <input type="text" name="nom_utilisateur" id="nom" size="100" />
```



```
</form>
```

Résultat :

Nom:

- ❖ L'attribut **maxlength** permet de limiter le nombre de caractères maximal que l'on peut saisir dans la zone de texte.
- ❖ L'attribut **value** permet de définir une valeur par défaut pour la zone de texte.

Formation.html

```
<form method="post" action="Traitement.php">  
  
  <label for="nom"> Nom: </label>  
  <input type="text" name="nom_utilisateur" id="nom" value="Aimé DIUMI" />  
</form>
```

Résultat :

Nom:

- ❖ L'attribut **placeholder** permet de donner une indication sur le contenu de la zone. Cette indication disparaîtra dès que le visiteur aura cliqué à l'intérieur du champ.

Formation.html

```
<form method="post" action="Traitement.php">  
  
  <label for="nom"> Nom: </label>  
  <input type="text" name="nom_utilisateur" id="nom" placeholder="Veuillez saisir votre nom" />  
</form>
```

Résultat :

Nom:

## VII.2 Zone de texte avec une liste de suggestions

Il est possible d'associer une liste de suggestions à une zone de texte, la liste apparaîtra quand la zone de texte aura le focus, le visiteur pourra alors retenir une suggestion ou saisir une autre valeur de son choix.

La liste de suggestions est réalisée avec les balises `<datalist>` `</datalist>` qui indique le début et la fin de la liste de suggestions. Entre les deux balises, on insère les éléments (suggestions) avec les balises `<option value="">``</option>`, comme valeur de l'attribut `value`, on indique la suggestion.

Exemple

Formation.html

```
<datalist>
  <option value="Aimé DIUMI"></option>
  <option value="Daniel MBILA"></option>
  <option value="Néhémie IYAMBE"></option>
  <option value="Serge BOIKA"></option>
  <option value="Nathan NANDOY"></option>
</datalist>
```

Pour lier la liste de suggestions avec la zone de texte, la première chose à faire est d'ajouter l'attribut `id` à la balise `<datalist>` et lui donner une valeur au choix. Ensuite, ajouter l'attribut `list` à la zone de texte et lui donner la même valeur que celle de l'attribut `id` de la liste de suggestions.

Formation.html

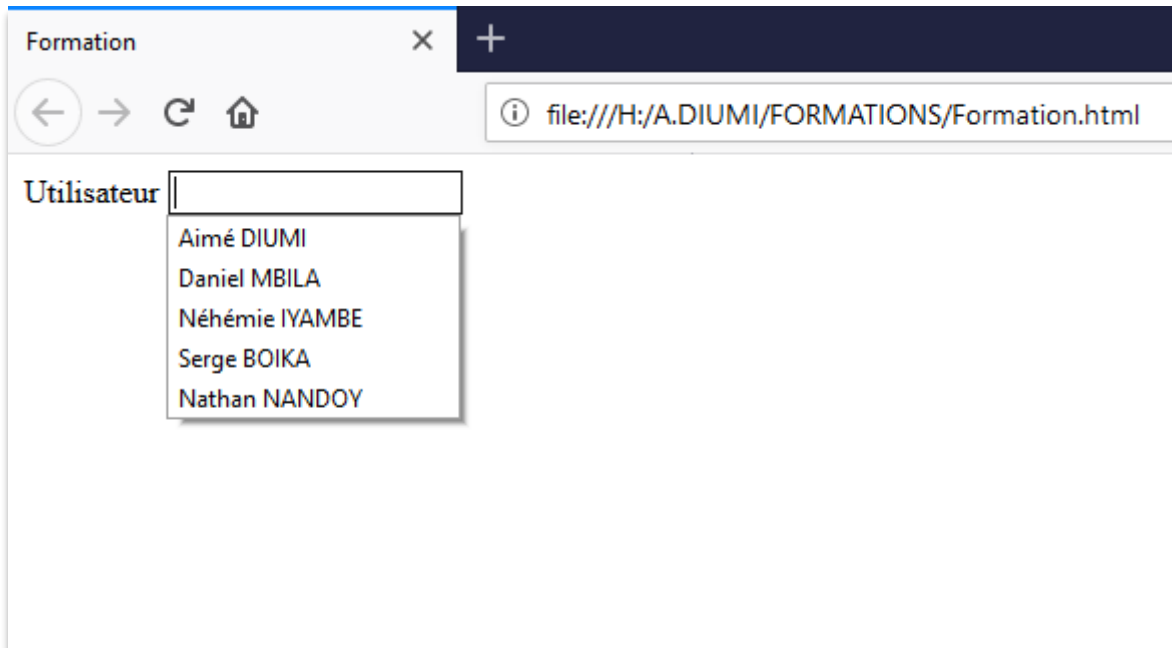
```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
  </head>
  <body>
<form action="Traitement.php" method="post">

  <label > Utilisateur </label>
<input type="text" list="noms">
<datalist id="noms">
<option value="Aimé DIUMI"></option>
<option value="Daniel MBILA"></option>
<option value="Néhémie IYAMBE"></option>
<option value="Serge BOIKA"></option>
<option value="Nathan NANDOY"></option>
```

```
</datalist>

</form>
</body>
```

Résultat :



## VII.3 Zone de mot de passe

Pour créer une zone de mot de passe, on utilise toujours la balise ***input***, à laquelle on ajoute l'attribut ***type="password"***.

Formation.html

```
<form method="post" action="Traitement.php">

  <label for="MotDePasse"> Mot de passe: </label>
  <input type="password" name="MotDePasse" id="MotDePasse" />
</form>
```

Résultat :

Quand j'essaie de saisir quelque chose, les caractères sont masqués :

Mot de passe:

Ajoutons les deux zones de saisie vues jusqu'ici dans notre formulaire :

Formation.html

```
<form method="post" action="Traitement.php">
  <p>
    <label for="nom"> Nom : </label>
    <input type="text" name="nom_utilisateur" id="nom" /> <br/> </p>

  <p> <label for="MotDePasse"> Mot de passe: </label>
    <input type="password" name="MotDePasse" id="MotDePasse" /> </p>
</form>
```

Résultat :

J'ai rempli les deux champs :



## VII.4 Zone de texte multiligne

Si vous ne l'avez pas remarqué, on ne peut écrire que sur une seule ligne avec la zone de texte que nous avons créée avec la balise **input**. Pour créer une zone de texte multiligne, on utilise la balise `<textarea>` `</textarea>`

Formation.html

```
<form method="post" action="Traitement.php">

  <label for="commentaires" > Commentaires : </label>
  <textarea name="commentaire1" id="commentaires" > </textarea>
</form>
```

Résultat :



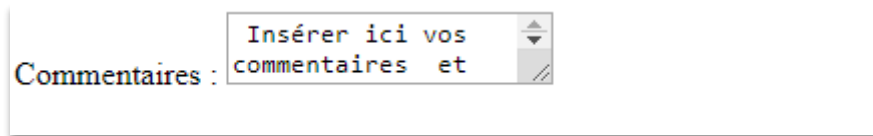
Pour ajouter une valeur par défaut à la zone, il suffit de mettre cette valeur entre la balise ouvrante et fermante :

## Formation.html

```
<form method="post" action="Traitement.php">

<label for="commentaires" > Commentaires : </label>
<textarea name="commentaire1" id="commentaires"> Insérer ici vos commentaires et suggestions </textarea>
</form>
```

Résultat :



Commentaires :

Pour changer les dimensions de la zone de saisie, nous avons deux possibilités :

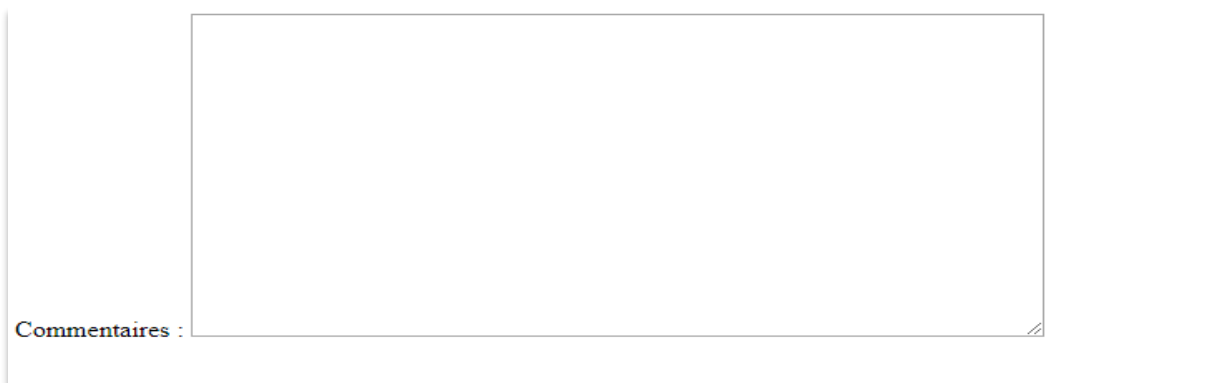
- ❖ Ajouter les attributs **rows** et **cols** à la balise **<textarea>**. Ils indiquent respectivement le nombre de lignes et de colonnes de texte qui peuvent être affichées simultanément.

## Formation.html

```
<form method="post" action="Traitement.php">

<label for="commentaires" > Commentaires : </label>
<textarea name="commentaire1" id="commentaires" rows="15" cols="70"> </textarea>
</form>
```

Résultat :



Commentaires :

- ❖ Appliquer les propriétés CSS *width* et *height* à la balise textarea.

Exemple

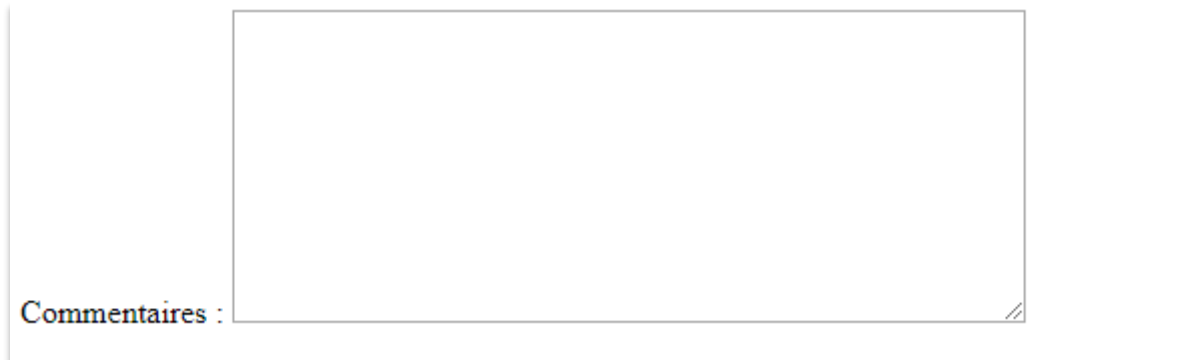
Formation.html

```
<form method="post" action="Traitement.php">
  <label for="commentaires" > Commentaires : </label>
  <textarea name="commentaire1" id="commentaires" > </textarea>
</form>
```

Forme.css

```
#commentaires
{
  width: 50%;
  height: 150px;
}
```

Résultat :



Commentaires :

A screenshot of a web form. On the left, there is a label 'Commentaires :'. To its right is a large, empty text area (textarea) with a thin border and a small cursor icon in the bottom right corner. The text area is rectangular and occupies most of the width of the form container.

## VII.5 Zones de texte enrichies

Les zones de texte enrichies vérifient que le contenu respecte le format requis et signalent l'erreur si ce n'est pas le cas tout en empêchant la validation du formulaire.

Toutes les zones que nous verrons dans cette section ont la balise `input` en commun, c'est la valeur de l'attribut `type` qui sera différente pour chaque zone.

### VII.5.1 L'email

Pour créer une zone pour la saisie d'une adresse e-mail, on utilise **`type="email"`**.

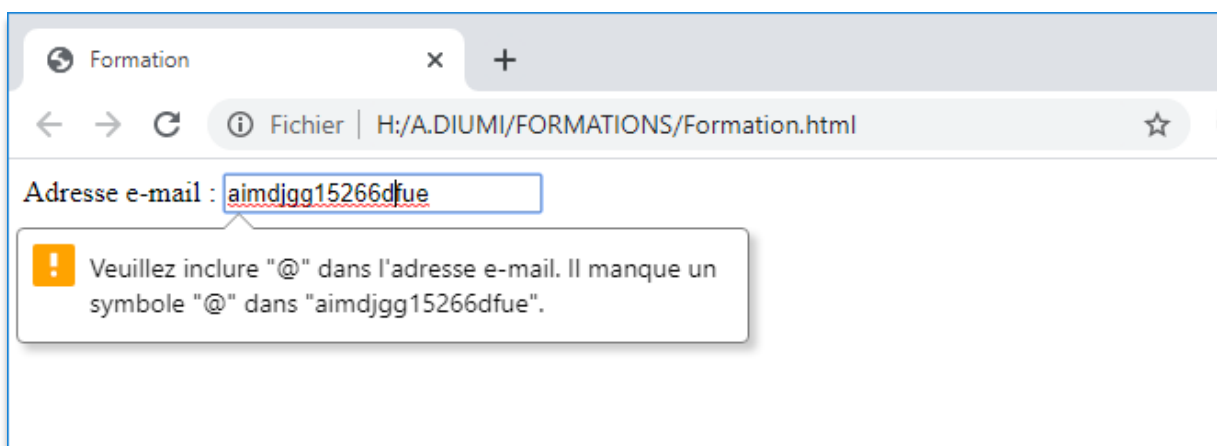
Formation.html

```
<form method="post" action="Traitement.php">
  <label for="email" > Adresse e-mail : </label>
  <input type="email" name="AdresseEmail" id="email" />
</form>
```

Résultat :

Adresse e-mail :

Le champ vous semblera a priori identique mais votre navigateur sait désormais que l'utilisateur doit saisir une adresse e-mail. Il peut afficher une indication si l'adresse n'est pas un e-mail, c'est ce que fait Google Chrome par exemple (figure suivante).



## VII.5.2 Une URL

Pour créer une zone qui recevra une adresse absolue (commençant généralement par http://), on utilise **type="url"**.

Formation.html

```
<label for="AdresseUrl" > Saisissez l'url : </label>
<input type="url" name="Adresseurl" id="AdresseUrl" />
</form>
```

## VII.5.3 Numéro de téléphone

On utilise **type="tel"** pour créer une zone pour la saisie d'un numéro de téléphone.

Formation.html

```
<form method="post" action="Traitement.php">
  <label for="phone" > Numéro téléphone : </label>
  <input type="tel" name="phone" id="phone" />
</form>
```

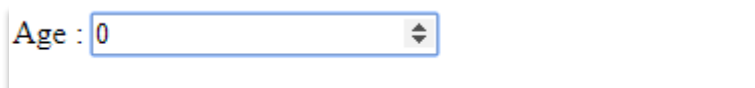
## VII.5.4 Nombre

On utilise **type="number"** pour créer une zone permettant de saisir un nombre.

Formation.html

```
<form method="post" action="Traitement.php">
  <label for="age" > Age : </label>
  <input type="number" name="age" id="age" />
</form>
```

Résultat :



### Attributs supplémentaires :

- **min** permet de définir la valeur minimale autorisée dans la zone
- **max** définit la valeur maximale autorisée.



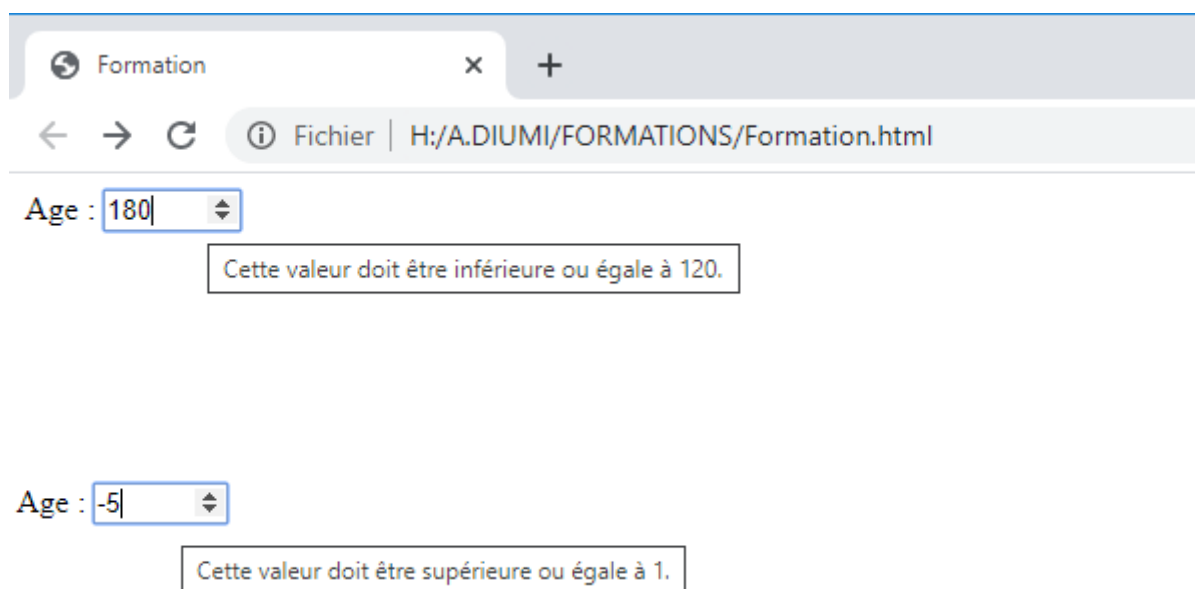
Par exemple

Formation.html

```
<form method="post" action="Traitement.php">

  <label for="age" > Age : </label>
  <input type="number" name="age" id="age" min="1" max="120"/>
</form>
```

Les valeurs autorisées vont de 1 à 120, si le visiteur introduit une valeur invalide, le navigateur affiche une notification comme le montre la figure suivante :



- **step** permet de définir le pas d'incréméntation ou de décrémentation. Si la valeur initiale est 0 et que vous définissez 2 comme pas de déplacement (incréméntation ou décrémentation), en montant, vous aurez les valeurs 0,2,4,6 etc... et en descendant, vous aurez 0, -2, -4 etc.

## VII.5.5 Un curseur

Le type **range** permet de sélectionner un nombre avec un curseur (aussi appelé slider).

Formation.html

```
<form method="post" action="Traitement.php">

  <label for="intensité" > Intensité: </label>
```

```
<input type="range" name="intensite" id="intensité" />
</form>
```

Résultat :



Les attributs vus pour le type number sont aussi applicables ici.

Il est possible d'utiliser une datalist pour montrer les différentes étapes. La procédure est la même que celle utilisée pour la zone de texte.

Exemple

Formation.html

```
<form method="post" action="Traitement.php" >
  <input type="range" list="valeurs" min="0" max="100" step="10" value="30" />
  <datalist id="valeurs">
    <option value="0" label="0"> </option>
    <option value="10" label="10"></option>
    <option value="20" label="20"></option>
    <option value="30" label="30"> </option>
    <option value="40" label="40"></option>
    <option value="50" label="50"></option>
    <option value="60" label="60"> </option>
    <option value="70" label="70"></option>
    <option value="80" label="80"></option>
    <option value="90" label="90"> </option>
    <option value="100" label="100"> </option>
  </datalist>
</form>
```

L'attribut label dans les options de la liste permet d'afficher un texte, mais il n'est pas encore totalement supporté.

Résultat :

Sur Google chrome



Sur Internet Explorer



## VII.5.6 La couleur

On utilise le type **color** pour créer un champ permettant de saisir (sélectionner) une couleur.

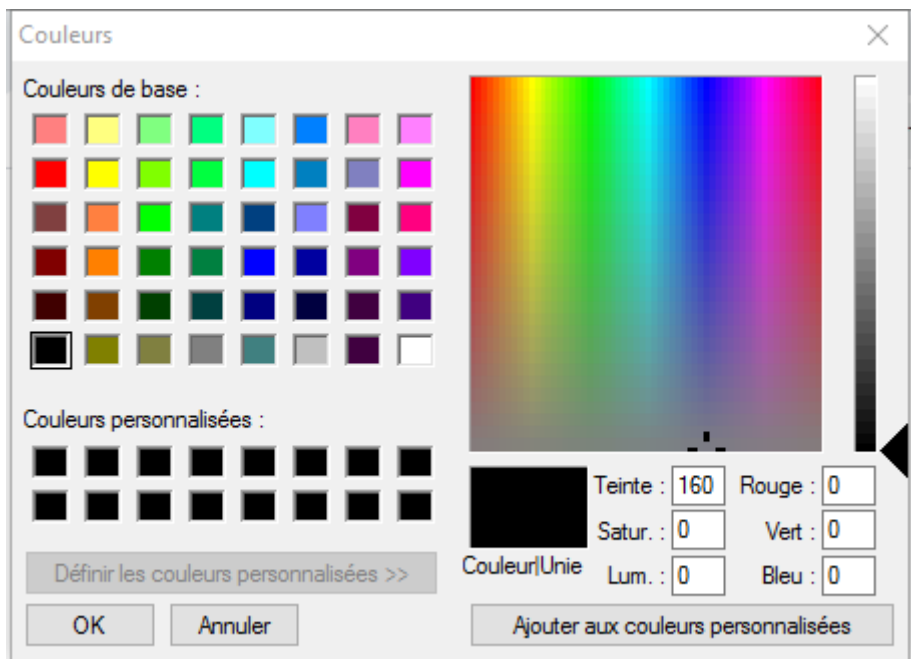
Formation.html

```
<form method="post" action="Traitement.php">  
  
  <label for="couleur" > Choisissez une couleur: </label>  
  <input type="color" name="couleur" id="couleur" />  
  </form>
```

Résultat :



Si vous cliquez sur le fond noir (couleur par défaut), la fenêtre suivante s'ouvre pour vous permettre de sélectionner une couleur :



## VII.5.7 Date

Différents types de champs de sélection de date existent :

- date : pour la date (05/08/1985 par exemple) ;
- time : pour l'heure (13:37 par exemple) ;
- week : pour la semaine ;
- month : pour le mois ;
- datetime : pour la date et l'heure (avec gestion du décalage horaire) ;
- datetime-local pour la date et l'heure (sans gestion du décalage horaire).


Exemple :




Formation.html

```
<form method="post" action="Traitement.php">  
  <label for="DateNaissance" > Date de naissance: </label>  
  <input type="date" name="DateNaissance" id="DateNaissance" />  
</form>
```

Résultat :

Date de naissance:

Date de naissance:  

mai 2020   

lun.	mar.	mer.	jeu.	ven.	sam.	dim.
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

## VII.5.8 Recherche

Pour créer un créer une zone de recherche, on utilise le type **search**.

## VII.6 Les cases à cocher

Les éléments d'options proposent au visiteur de faire un choix parmi une liste de possibilités.

Voici à quoi ressemblent les cases à cocher :



Langues parlées:

- Anglais
- Français
- Lingala
- Swahili
- Kikongo
- Tshiluba
- Kitetela

Pour créer une case à cocher, on utilise la balise `input` à laquelle on ajoute l'attribut **type="checkbox"**. Pour ajouter du texte à côté de la case à cocher, on utilise la balise `label`.

Exemple

Pour les cases ci-dessus, le code html est le suivant :

Formation.html

```
<form method="post" action="Traitement.php">
  <p> Langues parlées: </p>
  <input type="checkbox" name="Anglais" id="Anglais" /> <label for="Anglais" > Anglais </label> <br/>
  <input type="checkbox" name="Français" id="Français" /> <label for="Français" > Français </label> <br/>
  <input type="checkbox" name="Lingala" id="Lingala" /> <label for="Lingala" > Lingala </label> <br/>
  <input type="checkbox" name="Swahili" id="Swahili" /> <label for="Swahili" > Swahili </label> <br/>
  <input type="checkbox" name="Kikongo" id="Kikongo" /> <label for="Kikongo" > Kikongo </label> <br/>
  <input type="checkbox" name="Tshiluba" id="Tshiluba" /> <label for="Tshiluba" > Tshiluba </label> <br/>
  <input type="checkbox" name="Kitetela" id="Kitetela" /> <label for="Kitetela" > Kitetela </label>
```

```
</form>
```

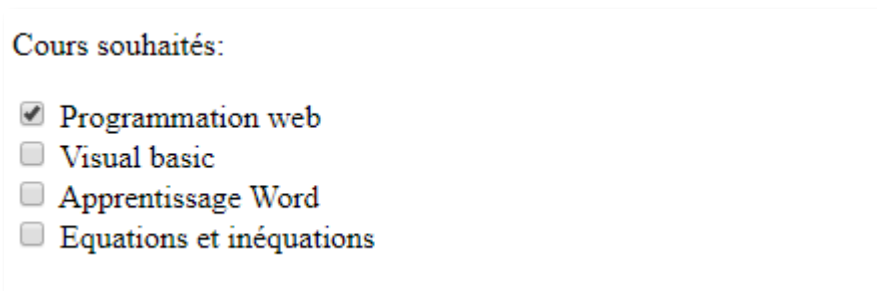
Pour faire en sorte qu'une case à cocher soit cochée (sélectionnée) par défaut, on utilise l'attribut **checked**.

Exemple :

Formation.html

```
<form method="post" action="Traitement.php">
  <p> Cours souhaités: </p>
  <input type="checkbox" name="Informatique1" id="Informatique1" checked /
> <label for= "Informatique1" >Programmation web </label> <br/>
  <input type="checkbox" name= "Informatique2" id="Informatique2" /> <label
for="Informatique2" > Visual basic </label> <br/>
  <input type="checkbox" name="Informatique3" id= "Informatique3" /> <label
for="Informatique3" > Apprentissage Word </label> <br/>
  <input type="checkbox" name="math1" id="math1" /> <label for= "math1"
> Equations et inéquations </label>
</form>
```

Résultat :



Cours souhaités:

- Programmation web
- Visual basic
- Apprentissage Word
- Equations et inéquations

## VII.7 Les zones d'options

Les zones d'options permettent de faire un et un seul choix parmi une liste de possibilités. Elles doivent être organisées en groupes pour permettre l'unicité du choix. Les zones d'un même groupe possèdent le même nom (même valeur de l'attribut name) mais de valeurs différentes (l'attribut value).

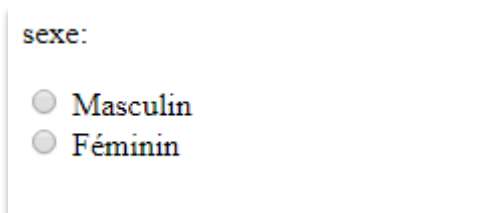
Pour créer une zone d'option, on utilise toujours la balise **input** avec le type **radio**.

## Exemple

Formation.html

```
<form method="post" action="Traitement.php">
  <p> sexe: </p>
  <input type="radio" name="sexe" value="Masculin" id="sexe1" /> <label for
= "sexe1" >Masculin</label> <br/>
  <input type="radio" name="sexe" value="Féminin" id="sexe2" /> <label for
="sexe2" > Féminin </label>
</form>
```

Résultat :



sexe:

Masculin

Féminin

## VII.8 Les listes déroulantes

### VII.8.1 Généralités

Pour créer une liste déroulante, on utilise la balise `<select>` `</select>` à laquelle on ajoute l'attribut **name** pour donner un nom à la liste déroulante. Les deux balises indiquent le début et la fin de la liste déroulante.

Puis, à l'intérieur du `<select>` `</select>`, nous allons placer plusieurs balises `<option>` `</option>` (une par choix possible). On ajoute à chacune d'elles un attribut **value** pour pouvoir identifier ce que le visiteur a choisi.

## Exemple

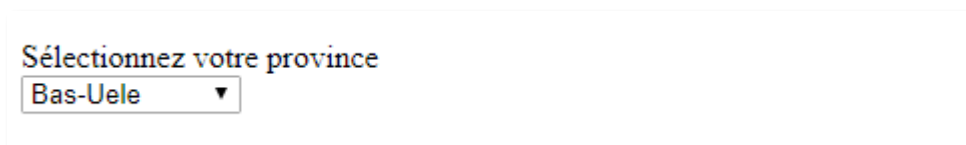
Formation.html

```
<form method="post" action="Traitement.php">
  <p>
    <label for="provinces">Sélectionnez votre province </label><br />
    <select name="provinces" id="provinces">
      <option value="Bas-Uele">Bas-Uele</option>
      <option value="Equateur">Equateur</option>
      <option value="Haut-Katanga">Haut-Katanga</option>
      <option value="Haut-Lomami">Haut-Lomami</option>
    </select>
  </p>
</form>
```

```
<option value="Haut-Uele">Haut-Uele</option>
<option value="Ituri">Ituri</option>
<option value="Kasaï">Kasaï</option>
<option value="Kasaï-central">Kasaï-central</option>
<option value="Kasaï-Oriental">Kasaï-Oriental</option>
<option value="Kinshasa">Kinshasa</option>
<option value="Kongo-Central">Kongo-Central</option>
<option value="Kwango">Kwango</option>
<option value="Kwilu">Kwilu</option>
<option value="Lomami">Lomami</option>
<option value="Lualaba">Lualaba</option>
<option value="Mai-Ndombe">Mai-Ndombe</option>
<option value="Maniema">Maniema</option>
<option value="Mongala">Mongala</option>
<option value="Nord-Kivu">Nord-Kivu</option>
<option value="Nord-Ubangi">Nord-Ubangi</option>
<option value="Sankuru">Sankuru</option>
<option value="Sud-Kivu">Sud-Kivu</option>
<option value="Sud-Ubangi">Sud-Ubangi</option>
<option value="Tanganyika">Tanganyika</option>
<option value="Tshopo">Tshopo</option>
<option value="Tshuapa">Tshuapa</option>
</select>
</p>
</form>
```

Résultat :

Avant que la liste soit déroulée



Sélectionnez votre province

Bas-Uele ▼

Liste déroulée :





L'attribut **selected** permet de sélectionner une option par défaut.

Par exemple, je veux que Sankuru soit sélectionnée par défaut :

Formation.html

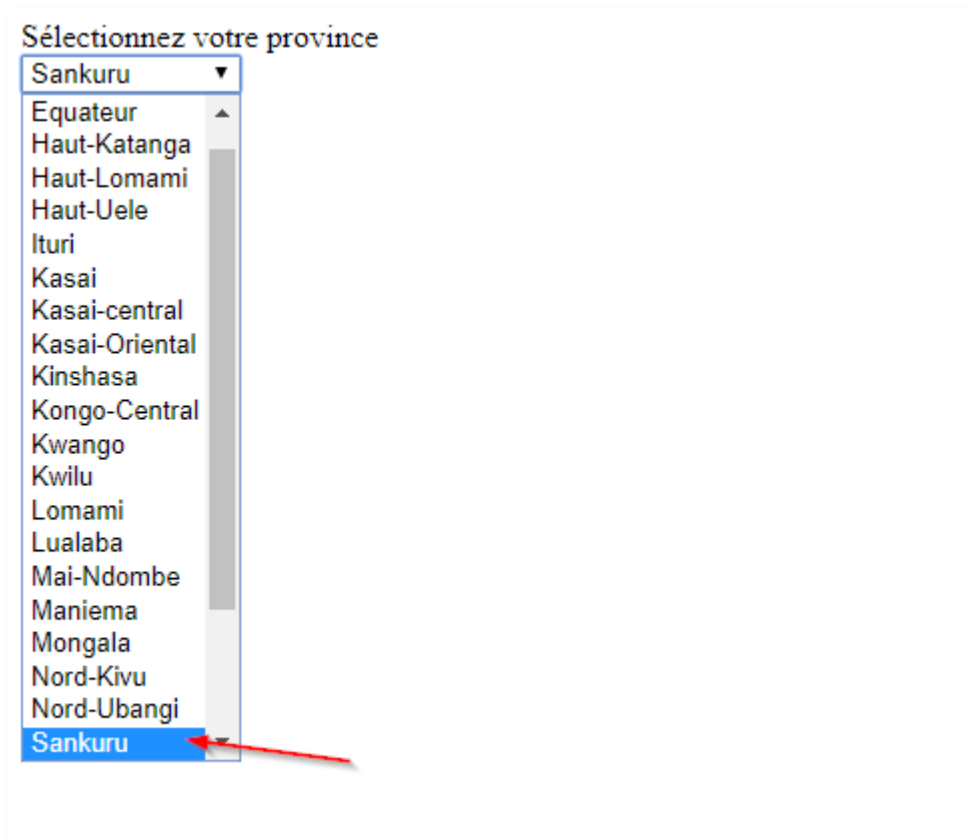
```
<form method="post" action="Traitement.php">
  <p>
    <label for="provinces">Sélectionnez votre province </label><br />
    <select name="provinces" id="provinces">
      <option value="Bas-Uele">Bas-Uele</option>
      <option value="Equateur">Equateur</option>
      <option value="Haut-Katanga">Haut-Katanga</option>
      <option value="Haut-Lomami">Haut-Lomami</option>
      <option value="Haut-Uele">Haut-Uele</option>
      <option value="Ituri">Ituri</option>
      <option value="Kasaï">Kasaï</option>
      <option value="Kasaï-central">Kasaï-central</option>
      <option value="Kasaï-Oriental">Kasaï-Oriental</option>
      <option value="Kinshasa">Kinshasa</option>
      <option value="Kongo-Central">Kongo-Central</option>
      <option value="Kwango">Kwango</option>
      <option value="Kwilu">Kwilu</option>
      <option value="Lomami">Lomami</option>
      <option value="Lualaba">Lualaba</option>
      <option value="Mai-Ndombe">Mai-Ndombe</option>
```

```

<option value="Maniema">Maniema</option>
  <option value="Mongala">Mongala</option>
  <option value="Nord-Kivu">Nord-Kivu</option>
  <option value="Nord-Ubangi">Nord-Ubangi</option>
  <option value="Sankuru" selected>Sankuru</option>
  <option value="Sud-Kivu">Sud-Kivu</option>
  <option value="Sud-Ubangi">Sud-Ubangi</option>
  <option value="Tanganyika">Tanganyika</option>
  <option value="Tshopo">Tshopo</option>
  <option value="Tshuapa">Tshuapa</option>
</select>
</p>
</form>

```

Résultat :



La balise `<optgroup>` `</optgroup>` permet de grouper les options.

Exemple

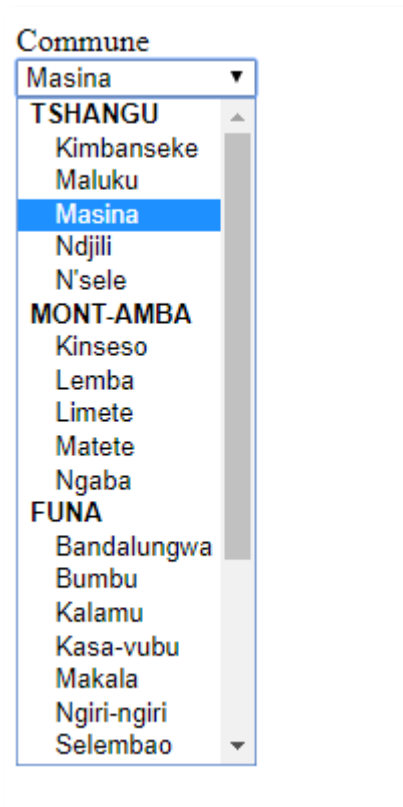
Affichons les communes de la ville province de Kinshasa en les groupant d'après les districts (Tshangu, Funa, Mont-Amba et Lukunga).

## Formation.html

```
<form method="post" action="Traitement.php">
  <p>
    <label for="provinces">Commune </label><br />
    <select name="communes" id="communes">
      <optgroup label="TSHANGU">
        <option value="Kimbanseke">Kimbanseke</option>
        <option value="Maluku">Maluku</option>
        <option value="Masina">Masina</option>
        <option value="Ndjili">Ndjili</option>
        <option value="N'sele">N'sele</option>
      </optgroup>

      <optgroup label="MONT-AMBA">
        <option value="Kinseso">Kinseso</option>
        <option value="Lemba">Lemba</option>
        <option value="Limete">Limete</option>
        <option value="Matete">Matete</option>
        <option value="Ngaba">Ngaba</option>
      </optgroup>
      <optgroup label="FUNA" >
        <option value="Bandalungwa">Bandalungwa</option>
        <option value="Bumbu">Bumbu</option>
        <option value="Kalamu">Kalamu</option>
        <option value="Kasa-vubu">Kasa-vubu</option>
        <option value="Makala">Makala</option>
        <option value="Ngiri-ngiri">Ngiri-ngiri </option>
        <option value="Selembao">Selembao</option>
      </optgroup>
      <optgroup label="LUKUNGA" >
        <option value="Barumbu">Barumbu</option>
        <option value="Gombe">Gombe</option>
        <option value="Kinshasa">Kinshasa</option>
        <option value="Kintambo" >Kintambo</option>
        <option value="Lingwala">Lingwala</option>
        <option value="Mont-Ngafula">Mont-Ngafula</option>
        <option value="Ngaliema">Ngaliema</option>
      </optgroup>
    </select>
  </p>
</form>
```

Résultat :



## VII.8.2 Options de la liste

### a) L'attribut size

Par défaut, seul un élément (sélectionné par défaut) est visible dans la liste, il faut la dérouler pour voir les autres éléments. Il est possible de changer ce comportement avec l'attribut **size** qui permet d'indiquer le nombre d'options de la liste.

Si **size="1"** : le navigateur affichera une liste déroulante classique, c'est-à-dire un seul élément sera visible à l'affichage de la page.

Si **size="n"**, avec n un entier quelconque : le navigateur affichera une liste déroulée qui fait apparaître le nombre d'options spécifié par l'attribut size.

Exemple

Formation.html

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel="stylesheet" href="forme.css" />
```

```

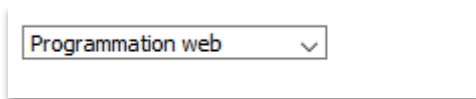
</head>
<body>

<form method="post" action="Traitement.php">
  <p>
    <select name="cours" id="cours" size="1">
      <option value="Programmation_web">Programmation web</option>
      <option value="Visual_basic">Visual basic</option>
      <option value="Excel">Excel </option>
      <option value="Programmation_en_java">Programmation en Java</option>
      <option value="Les_mathématiques_faciles"> Les mathématiques faciles</option>
      <option value="Mysql"> MySQL</option>
    </select>
  </p>
</form>

</body>
</html>

```

Résultat :



A screenshot of a web browser displaying a single dropdown menu. The menu is open, showing the selected option 'Programmation web' and a small downward-pointing arrow on the right side of the menu box.

Changeons maintenant la valeur de l'attribut size :

Formation.html

```

<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
    <link rel="stylesheet" href="forme.css" />
  </head>
  <body>

  <form method="post" action="Traitement.php">
    <p>
      <select name="cours" id="cours" size="6">
        <option value="Programmation_web">Programmation web</option>
        <option value="Visual_basic">Visual basic</option>
        <option value="Excel">Excel </option>
        <option value="Programmation_en_java">Programmation en Java</option>
        <option value="Les_mathématiques_faciles"> Les mathématiques faciles</option>
      </select>
    </p>
  </form>

```

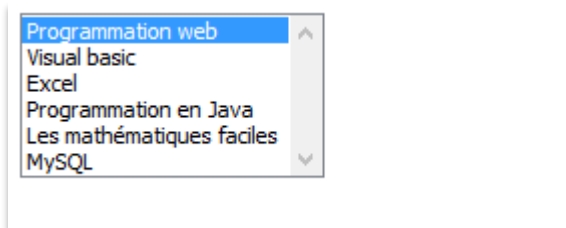
```

        <option value="Mysql"> MySQL</option>
    </select>
</p>
</form>

</body>
</html>

```

Résultat :



### *b) L'attribut multiple*

Vous l'avez sans doute remarqué : le visiteur ne peut choisir qu'un seul élément dans la liste déroulante, mais il est tout à fait possible de permettre au visiteur de sélectionner plusieurs éléments de la liste déroulante. Cela se fait grâce à l'attribut **multiple** de la balise **<select>**.

Pour sélectionner plusieurs éléments de la liste déroulante, l'utilisateur doit maintenir la touche Ctrl enfoncée et cliquer sur les éléments de la liste avec la souris.

Exemple

Formation.html

```

<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Formation</title>
  </head>
  <body>
<form method="post" action="Traitement.php">
  <p>
    <label for="cours" >Sélectionnez les cours que vous maitrisez </label><br />
    <select name="cours" id="cours" multiple>
      <option value="Programmation_web">Programmation web</option>
      <option value="Visual_basic">Visual basic</option>
      <option value="Excel">Excel </option>

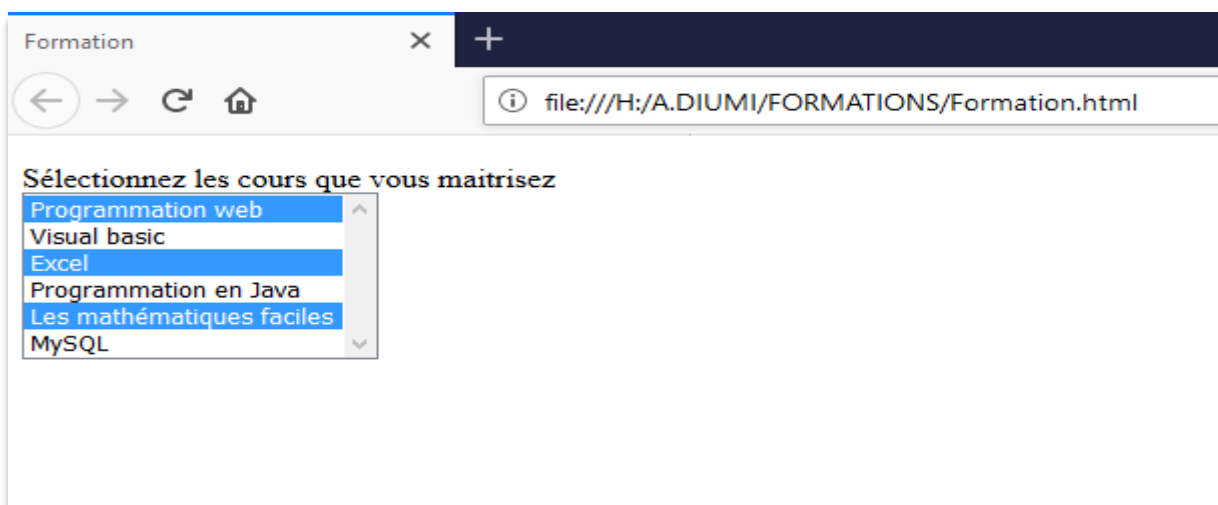
```

```

        <option value="Programmation_en_java">Programmation en Java</option
    >
        <option value="Les_mathématiques_faciles"> Les mathématiques facile
s</option>
        <option value="Mysql"> MySQL</option>
    </select>
</p>
</form>
</body>
</html>

```

Je peux sélectionner plusieurs éléments comme le montre la figure suivante :



## VII.9 Le transfert des fichiers

Pour permettre le transfert de fichiers du poste de l'utilisateur vers un ordinateur de type serveur, on utilise la balise `<input type="file">`.

Exemple

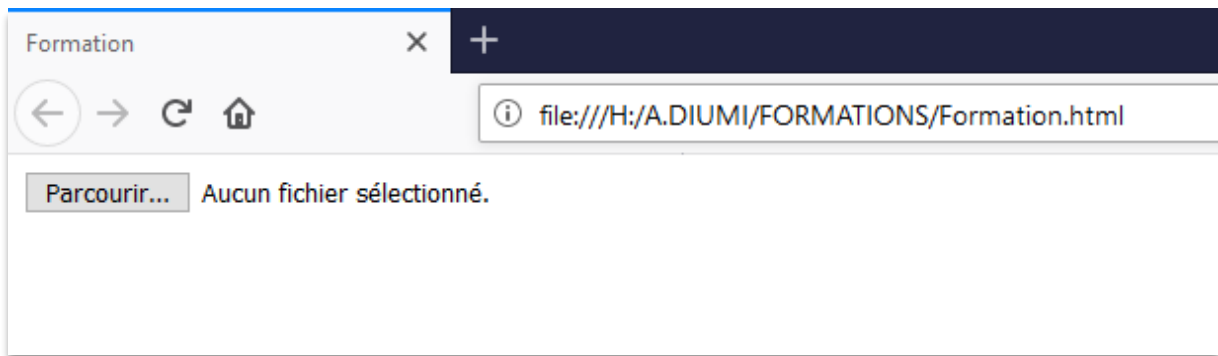
Formation.html

```

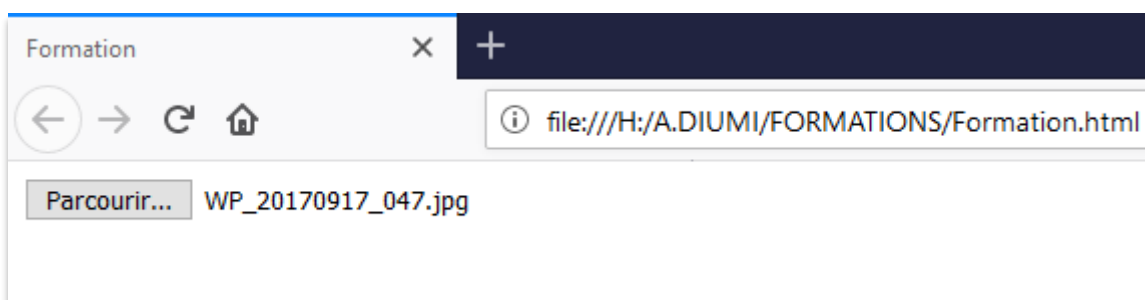
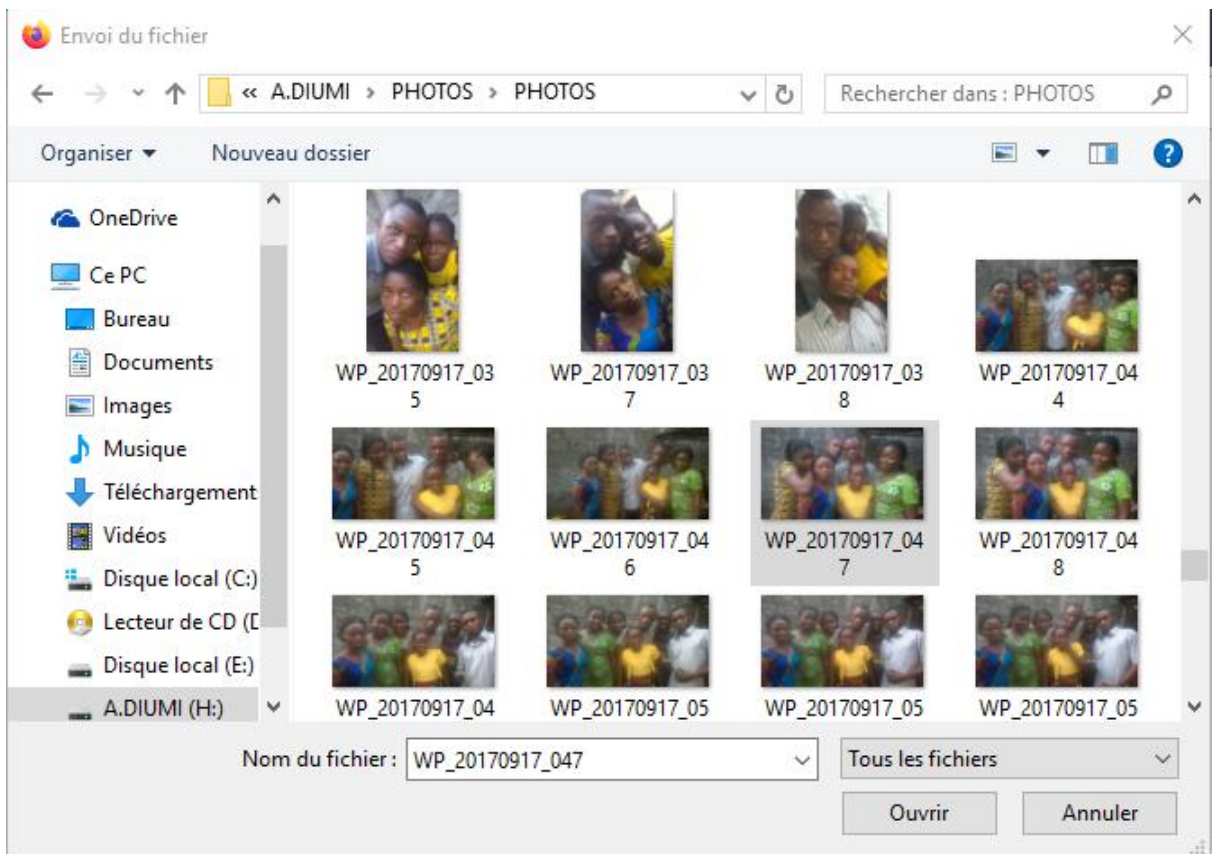
<!doctype html>
<html>
    <head>
        <meta charset="utf-8">
        <title> Formation</title>
    </head>
    <body>
<form action="Traitement.php" method="post" enctype="multipart/form-data">
<input type="file" name="fichier_photo">
</form>
</body>
</html>

```

Résultat :



Un clic sur le bouton « Parcourir » ouvre l'exploitateur de fichiers de votre ordinateur pour sélectionner le fichier à envoyer comme le montrent les figures suivantes :





Quand on utilise ce type de balise, Il faut impérativement utiliser les attributs `method="post"` `enctype="multipart/form-data"` dans la balise `<form>` pour le transfert au bon format du fichier.

## VII.10 Regrouper les champs

Si le formulaire contient plusieurs champs, il est alors nécessaire de grouper certains champs. On utilise la balise `<fieldset>` `</fieldset>` pour grouper les champs et on ajoute une légende (le titre du groupe) avec la balise `<legend>` `</legend>`.

Exemple

Formation.html

```
<form method="post" action="Traitement.php">
  <fieldset >
    <legend> Identité </legend>

    <fieldset>
      <label for="prenom" > Prénom </label><br/>
      <input type="text" name="prenom" id="prenom" /> <br/>
      <label for="nom"> Nom </label> <br/>
      <input type="text" name="nom" id="nom" /> <br/>
      <label for="postnom" > Postnom </label><br/>
      <input type="text" name="postnom" id="postnom" /> <br/>
      <label for="dateNaissance" > Date de naissance </label><br/>
      <input type="date" name="dateNaissance" id="dateNaissance" /> <br/>
    </fieldset>

    <fieldset>
      <legend > Sexe</legend>
      <input type="radio" name="sexe" value="Masculin" id="sexe1" /> <label for="sexe1" >Masculin</label> <br/>
      <input type="radio" name="sexe" value="Féminin" id="sexe2" /> <label for="sexe2" > Féminin </label>
    </fieldset>
  </fieldset>

  <fieldset>
    <legend > Localisation </legend>
    <p><label for="avenue" >Avenue </label>
    <input type="text" name="avenue" id="avenue" />
    <label for="no" >No </label>
    <input type="text" name="numero" id="no" size="5" /> <p/>
    <p> <label for="quartier" > Quartier </label>
    <input type="text" name="quartier" id="quartier" /> <br/> </p>
    <label for="provinces">Commune </label>
    <select name="communes" id="communes">
```

```

<optgroup label="TSHANGU">
  <option value="Kimbanseke">Kimbanseke</option>
  <option value="Maluku">Maluku</option>
<option value="Masina">Masina</option>
  <option value="Ndjili">Ndjili</option>
  <option value="N'sele">N'sele</option>
</optgroup>

<optgroup label="MONT-AMBA">
<option value="Kinseso">Kinseso</option>
  <option value="Lemba">Lemba</option>
  <option value="Limete">Limete</option>
  <option value="Matete">Matete</option>
  <option value="Ngaba">Ngaba</option>
</optgroup>

<optgroup label="FUNA" >
  <option value="Bandalungwa">Bandalungwa</option>
  <option value="Bumbu">Bumbu</option>
  <option value="Kalamu">Kalamu</option>
  <option value="Kasa-vubu">Kasa-vubu</option>
  <option value="Makala">Makala</option>
  <option value="Ngiri-ngiri">Ngiri-ngiri </option>
  <option value="Selembao">Selembao</option>
</optgroup>

<optgroup label="LUKUNGA" >
  <option value="Barumbu">Barumbu</option>
  <option value="Gombe">Gombe</option>
  <option value="Kinshasa">Kinshasa</option>
  <option value="Kintambo" >Kintambo</option>
  <option value="Lingwala">Lingwala</option>
  <option value="Mont-Ngafula">Mont-Ngafula</option>
  <option value="Ngaliema">Ngaliema</option>
</optgroup>
</select>
</fieldset>

</form>

```

Résultat :

**Identité**

Prénom

Nom

Postnom

Date de naissance

**Sexe**

Masculin

Féminin

**Localisation**

Avenue  No

Quartier

Commune

## VII.11 Compléments sur les champs

- ❖ L'attribut **autofocus** permet de placer automatiquement le curseur dans un champ (zone de saisie). Par exemple, si je veux que le curseur soit placé sur le champ « Postnom », il suffit d'ajouter autofocus dans le champ de Postnom comme ceci :

Formation.html

```
<input type="text" name="postnom" id="postnom" autofocus /> <br/>
```

Résultat :

Le curseur se place automatiquement dans la zone de saisie du Postnom

Identité

Prénom

Nom

Postnom

Date de naissance

Sexe

Masculin  
 Féminin

Localisation

Avenue  No

Quartier

Commune

- ❖ Pour rendre un champ obligatoire, on utilise l'attribut **required**. C'est-à-dire que le visiteur doit renseigner une valeur pour ce champ avant l'envoi du formulaire.

## VII.12 Le bouton d'envoi

Pour créer le bouton d'envoi, on utilise toujours la balise input. Pour l'attribut type, il y a quatre valeurs possibles :

- **type="submit"** : il envoie les données du formulaire au serveur pour un traitement par le script spécifié dans l'attribut action de la balise <form>
- **type="reset"** : remise à zéro (réinitialisation) du formulaire.
- **type="image"** : équivalent du bouton submit, présenté cette fois sous forme d'image. On ajoute l'attribut src pour indiquer l'URL(l'adresse) de l'image.
- **type="button"** : bouton générique, qui n'aura (par défaut) aucun effet. En général, ce bouton est géré en JavaScript pour exécuter des actions sur la page.

L'attribut **value** permet de changer le texte écrit sur le bouton

## Exemple

Ajoutons les boutons « Envoyer » et « Annuler » à notre formulaire. Le reste de code ne change pas, j'ajoute seulement le code des boutons.

Formation.html

```
<p>
  <fieldset>
    <input type="submit" value="Envoyer" />
    <input type="reset" value="Annuler" />
  </fieldset>
</p>
```

Résultat :

Identité

Prénom

Nom

Postnom

Date de naissance

Sexe

Masculin  
 Féminin

Localisation

Avenue  No

Quartier

Commune

Quand le visiteur saisira les informations, il doit cliquer sur le bouton « Envoyer » pour que les informations soient traitées, mais il faudra un autre langage (PHP par exemple) pour traiter ces informations.