POSITIONNEMENT EN CSS

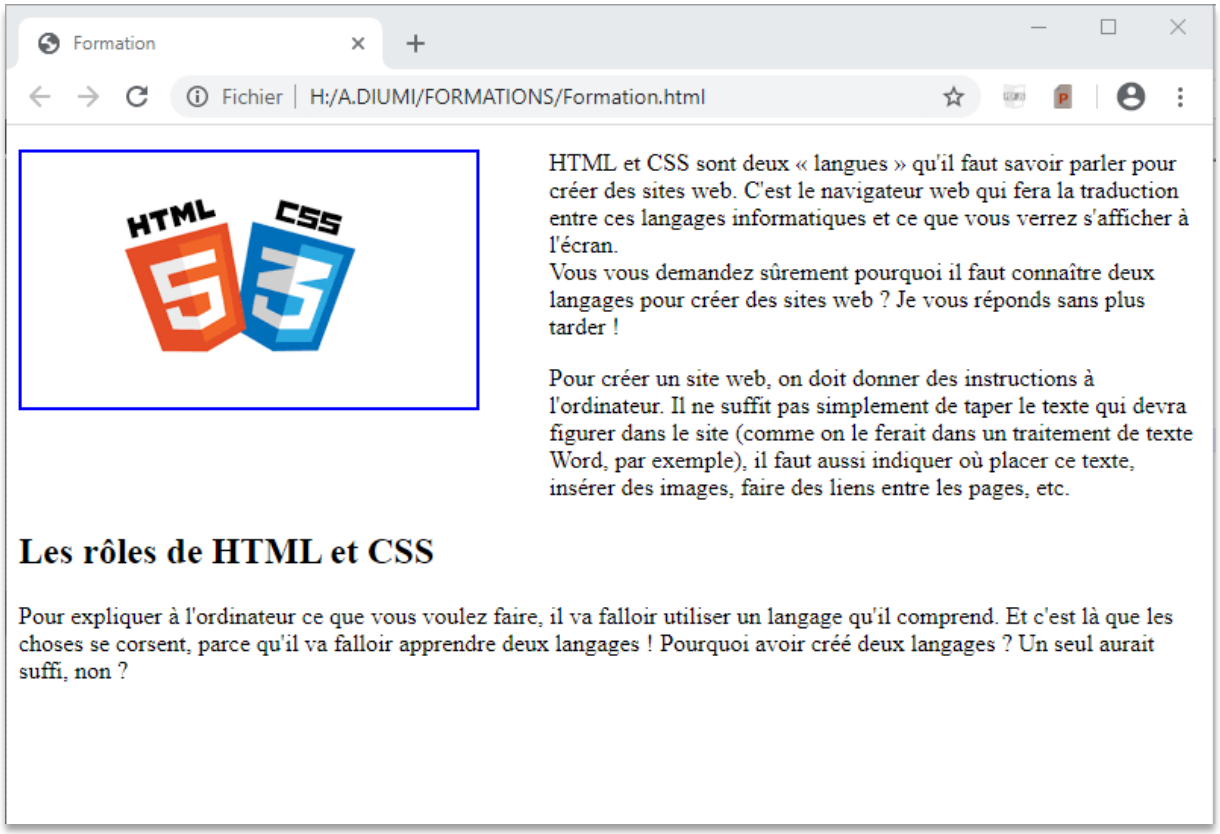
Aimé DIUMI DIKOLO

www.wiscorp.com

Table des matières

I. LE POSITIONNEMENT FLOTTANT

Il est possible avec Css 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>
 <img class= "image" src="images.png"   alt="oups" />
<p >HTML et CSS sont deux « langues » qu'il faut savoir parler pour créer des sit
es 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 d
es 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 ex
emple), 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 utilise
r 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>
 <img class= "image" src="images.png"   alt="oups" />
<p >HTML et CSS sont deux « langues » qu'il faut savoir parler pour créer des sit
es 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 d
es sites web ? Je vous réponds sans plus
tarder ! </p>

<p  class="para">Pour créer un site web, on doit donner des instructions à l'ordi
nateur. 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 ex
emple), 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 utilise
r 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

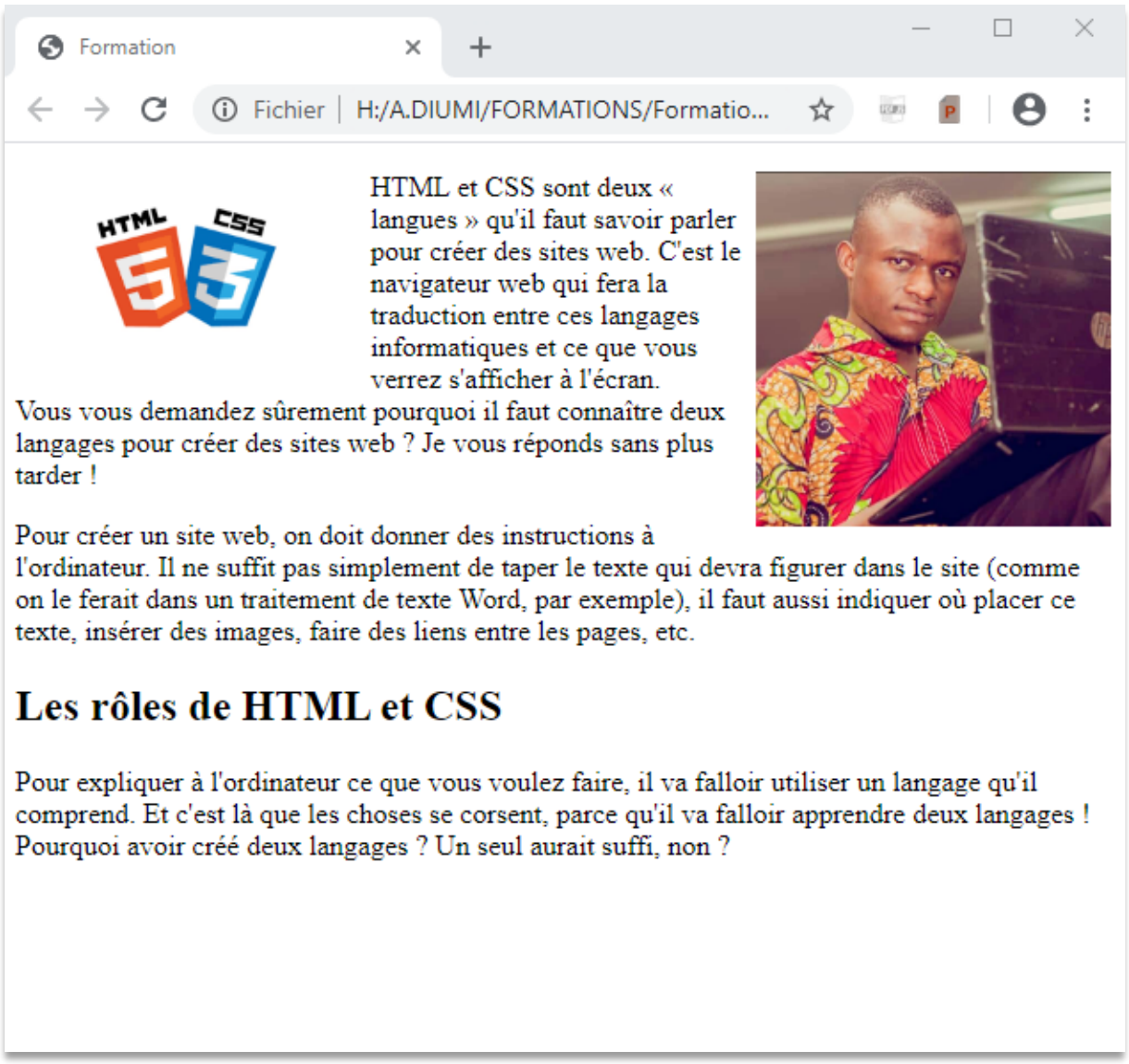
```
<!doctype html>
<html>
    <head>
        <meta charset="utf-8">
        <title> Formation</title>
      <link rel=stylesheet   href="forme.css"  />
    </head>
    <body>

<img class= "image" src="images.png"  alt="oups" width="200" />
<img class="image2" src="photoDiumi.jpg" alt="Désolé" width="200" />
<p >HTML et CSS sont deux « langues » qu'il faut savoir parler pour créer des sit
es 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 d
es 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 ex
emple), 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 Css :

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 événementiel
le de troisième génération ainsi qu'un environnement de développement intégré, cr
éé par Microsoft pour son modèle COM. <br/>Visual basic est directement dérivé du
 BASIC (acronyme pour Beginner's All purpose Symbolic Instruction Code ) et perme
t le développement rapide d'applications, la création d'interfaces utilisateur gr
aphiques, 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 é
diteur 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 d
éveloppement intégré (EDI) ou IDE en anglais, pour Integrated development Environ
ment. </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 conn
aissances sans les partager, c'est se mettre au niveau de celui qui n'a pas d'idé
es ». Wissen Corporation est là pour assurer votre formation, votre encadrement s
uivant vos désirs. </p>
</section>
</body>
</html>
```

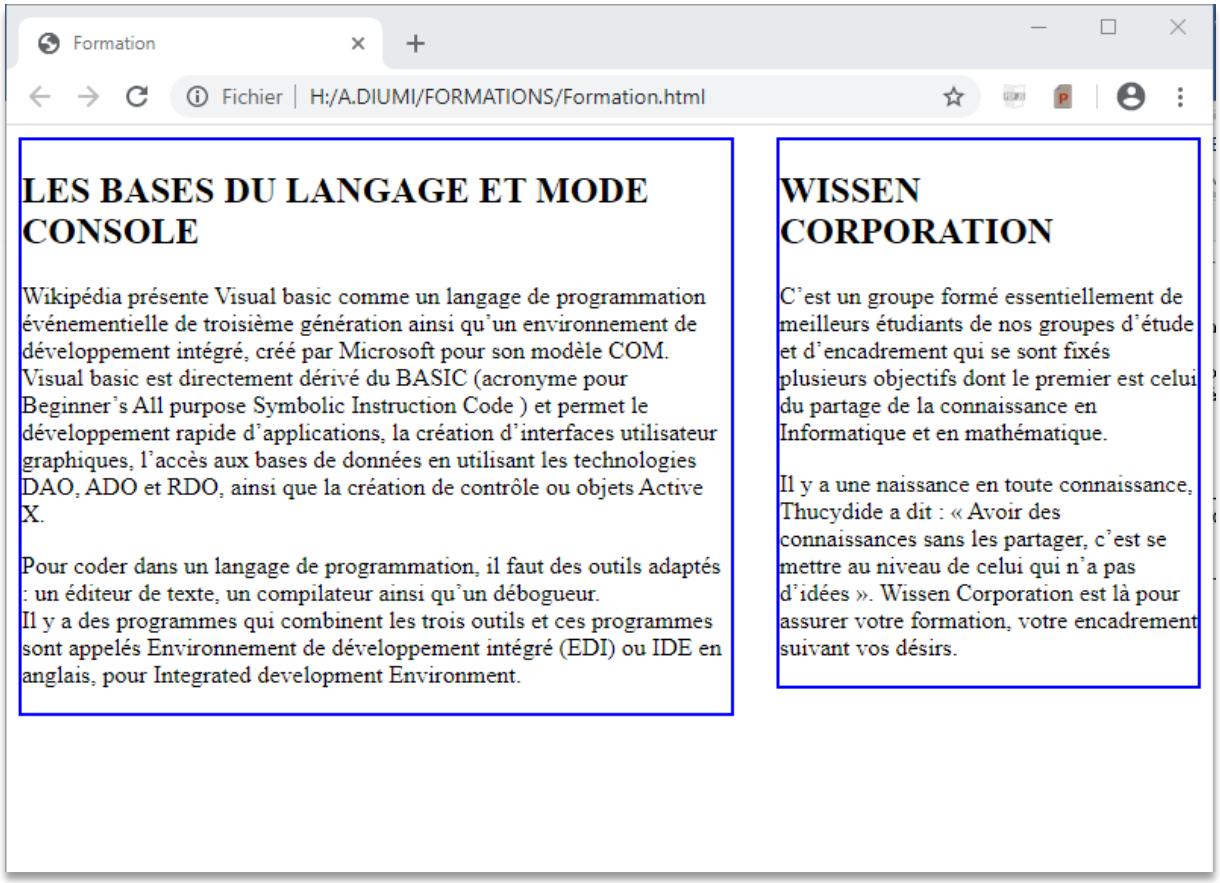
J'aimerai que les deux sections (bloc1 et bloc2) soient côte à côte, il suffit de flotter bloc1 comme ceci :

```
.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.

II. POSITIONNEMENT AVEC LA PROPRIETE *Display*

La plupart des balises peuvent être ranger 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

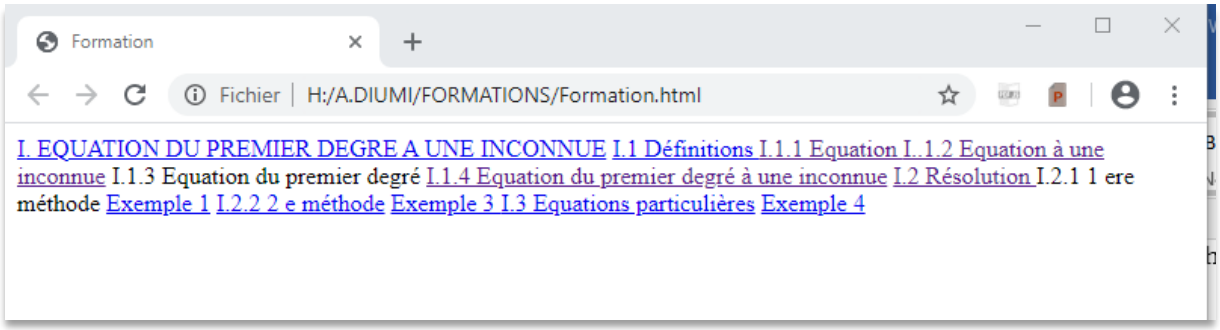
Exemple1

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 :



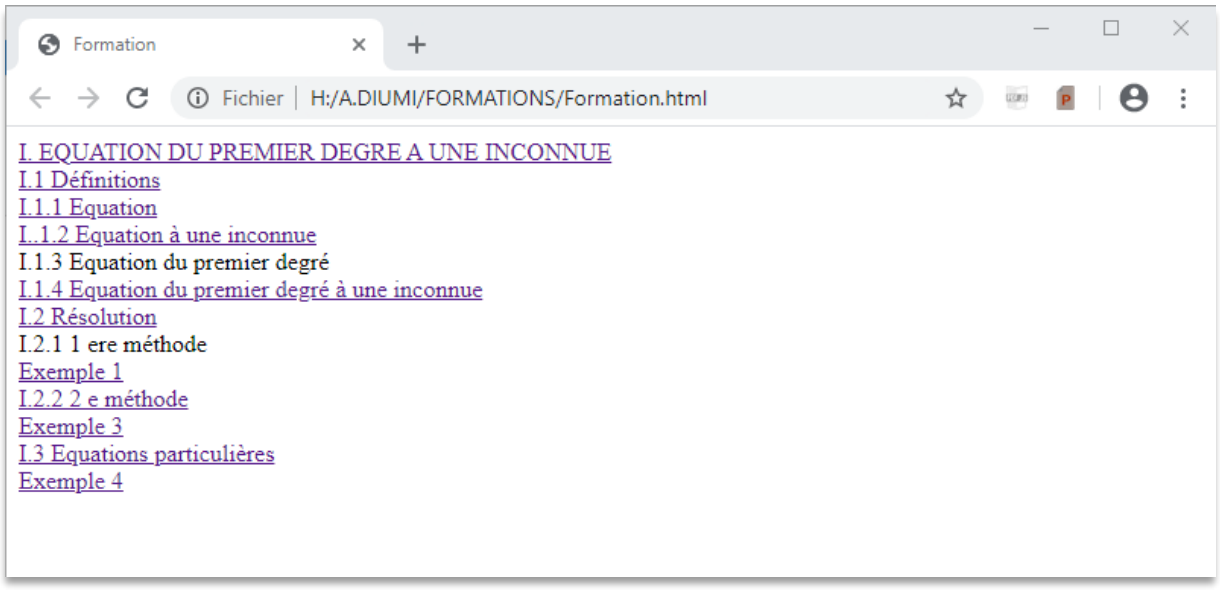
Pour faire en sorte qu'ils (les liens) se placent les uns en dessous des autres, on peut utiliser
, mais avec cette balise, les liens seront toujours de type inline et il n'y aura pas moyen de les redimensionner. La meilleure solution est de les

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.

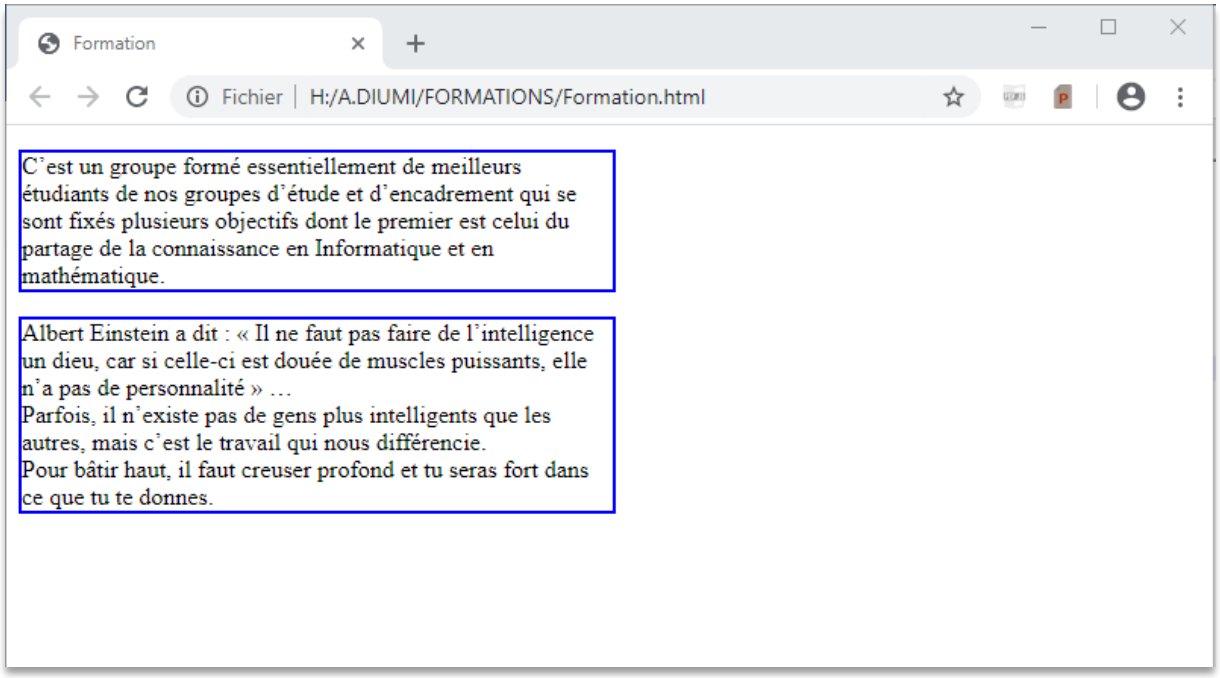
Exemple 2

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 l
e premier est celui du partage de la connaissance en Informatique et en mathémati
que. </p>
 <p class="para2">
Albert Einstein a dit : « Il ne faut pas faire de l’intelligence un dieu,
car si celleci est douée de muscles puissants, elle n’a pas de personnalité » …<b
r/>
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 donne
s. </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 … = 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>
```

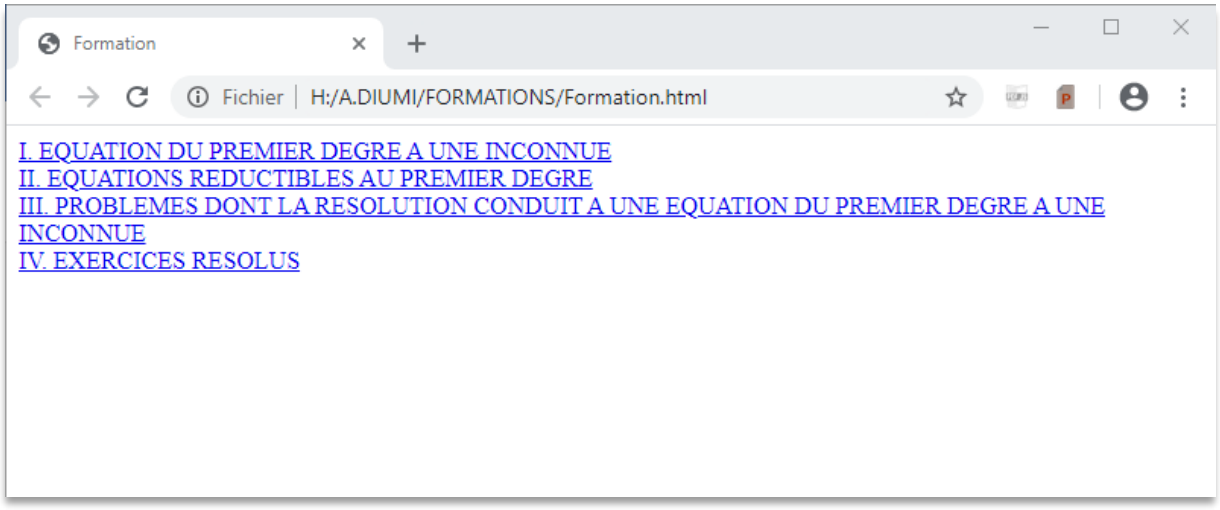
```
<a href="#LIII" >III. PROBLEMES DONT LA RESOLUTION CONDUIT A UNE EQUATION DU PREM
IER DEGRE A UNE INCONNUE</a><br/>
<section class="sous">
<a href="#Ex12" >Exemple 12 </a> <br/>
<a href="#Ex13" >Exemple 13</a> <br/>
</section>
<a href ="#LIV">IV. EXERCICES RESOLUS</a> <br/>
</body>
</html>
```

J'aimerai afficher au départ seulement les quatre principaux points et cacher les sous points :

Forme.css

```
.sous
{
    display: none;

}
```

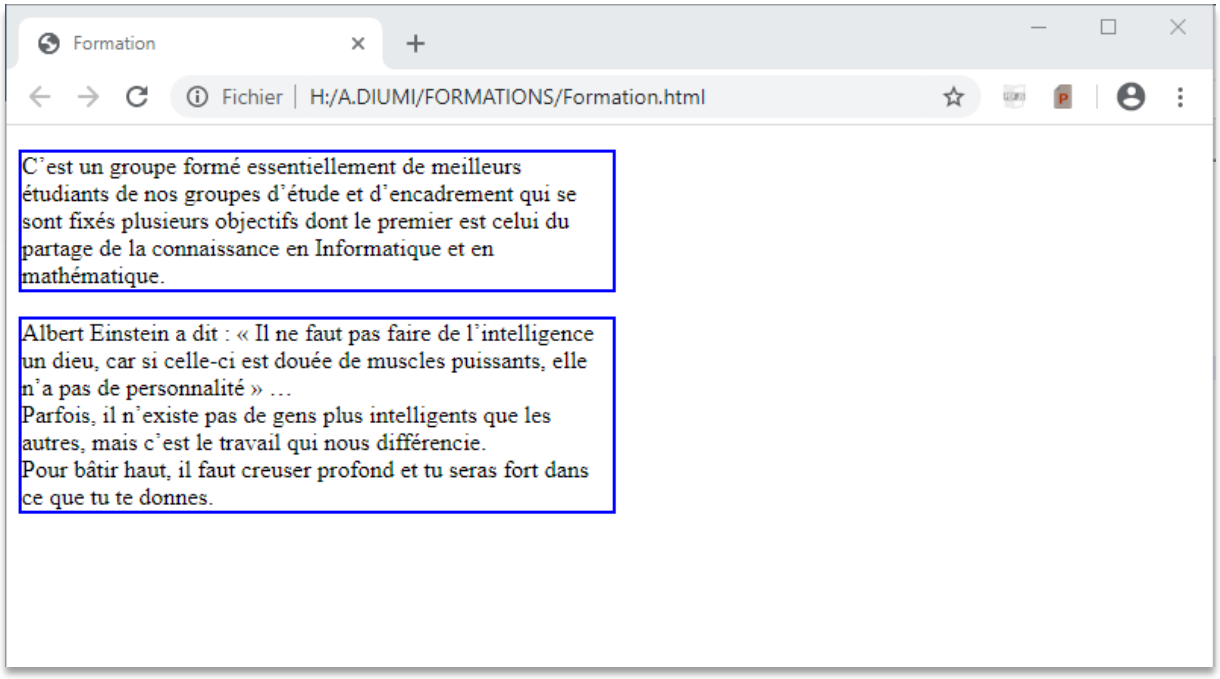
Résultat :



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 :



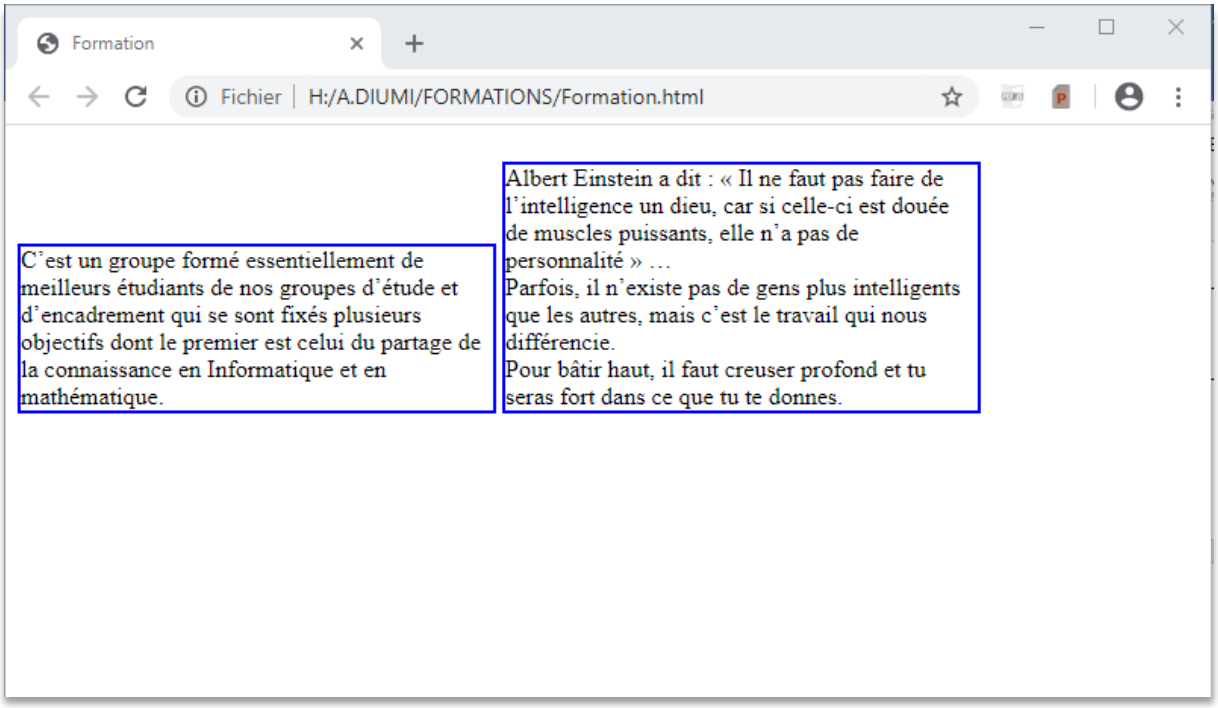
Pour que les deux blocs se positionnent l'un à côté de l'autre et qu'on ait la possibilité de les redimensionner (leur donner des tailles précises), on doit les transformer en type inline-block.

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 :

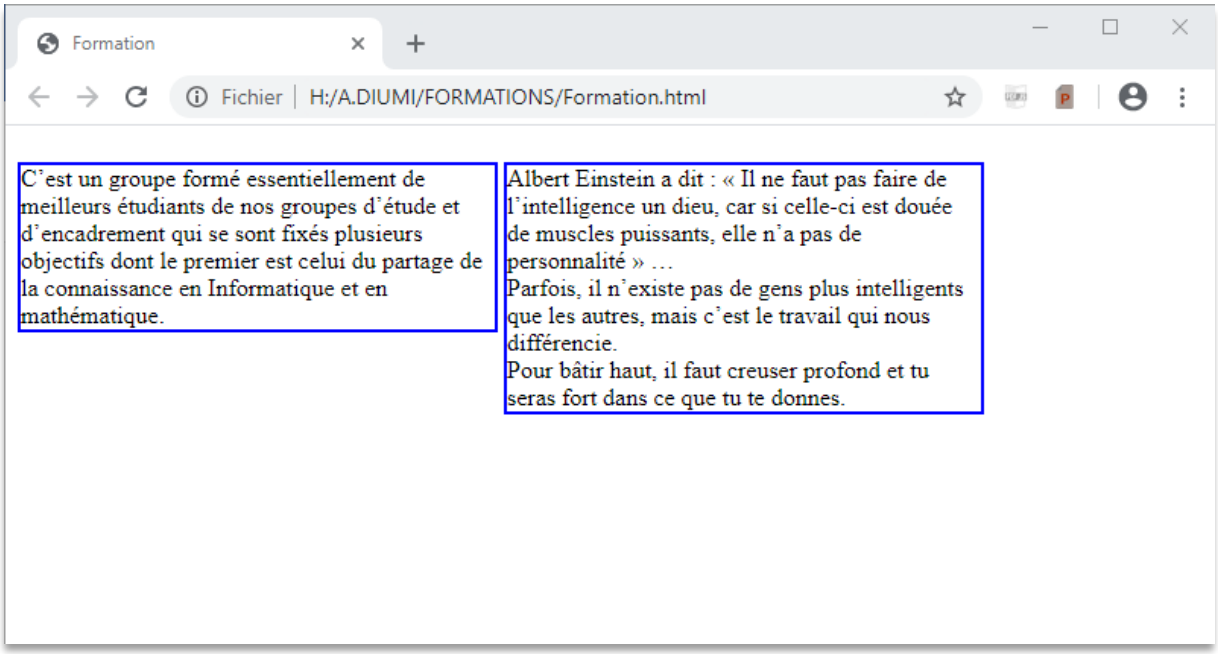
- ***baseline*** : aligne de la base de l'élément avec celle de l'élément parent (par défaut).Voir exemple ci-dessus.
- ***top*** : aligne en haut ;

forme.css

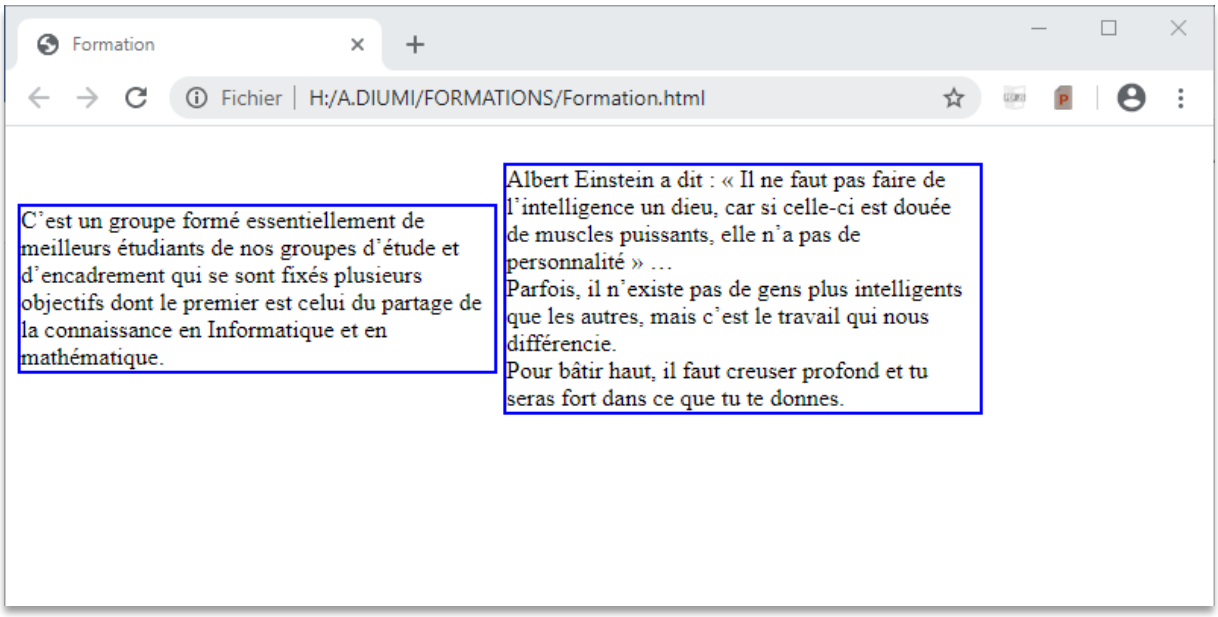
```
.para1
{
    display: inline-block;
    vertical-align: top;
    border: 2px blue solid;
    width: 40%;
}
```

```css
.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. POSITIONNEMENT AVEC FLEXBOX

III.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 positionnement 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 donne
s. </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>
```

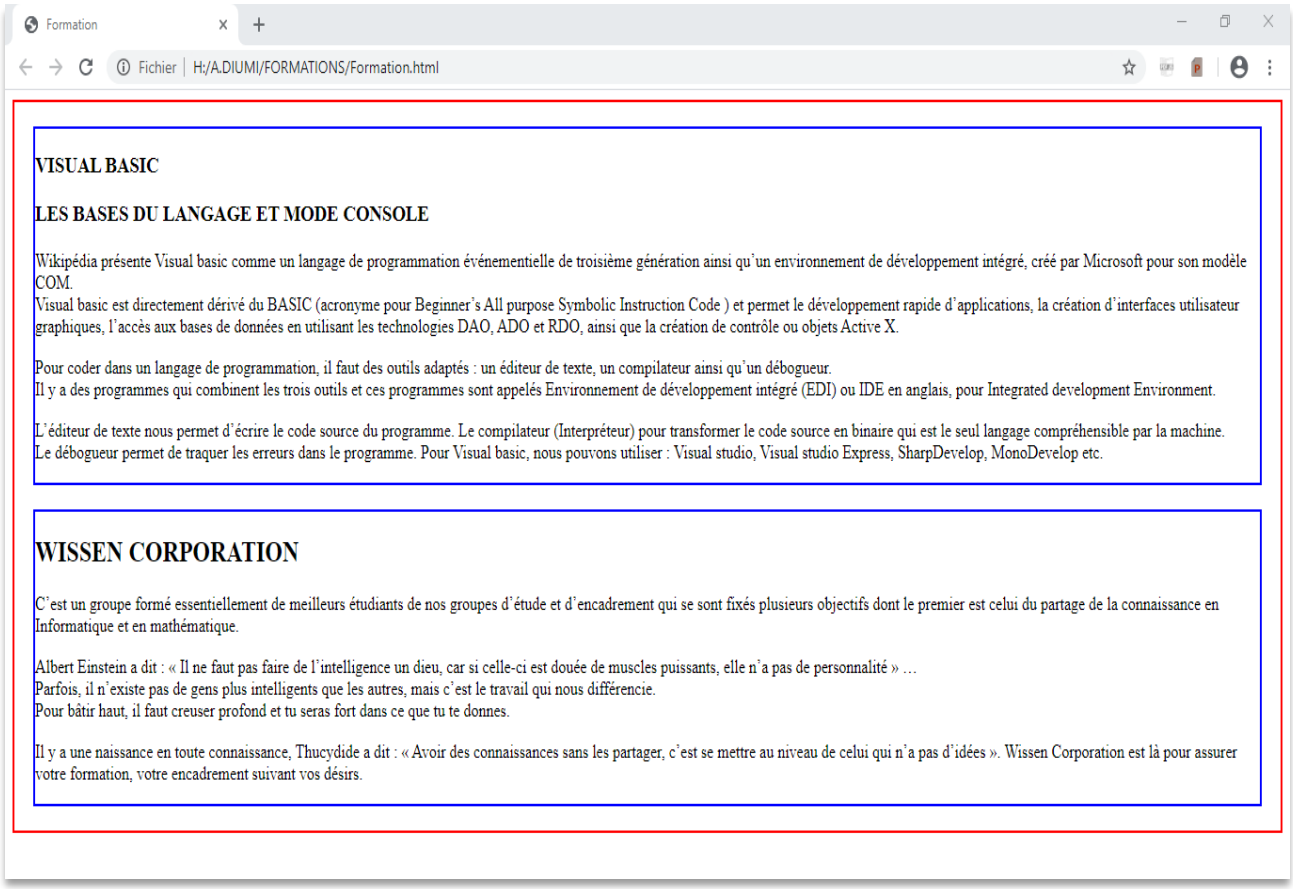
Ajoutons une bordure à chaque sous-section et à la section principale :

Forme.css

```
.bloc
{
border: 2px red solid;

}

.sousBloc1, .sousBloc2
{
    border: 2px blue solid;
```

```
    margin: 20px;
}
```

Résultat :



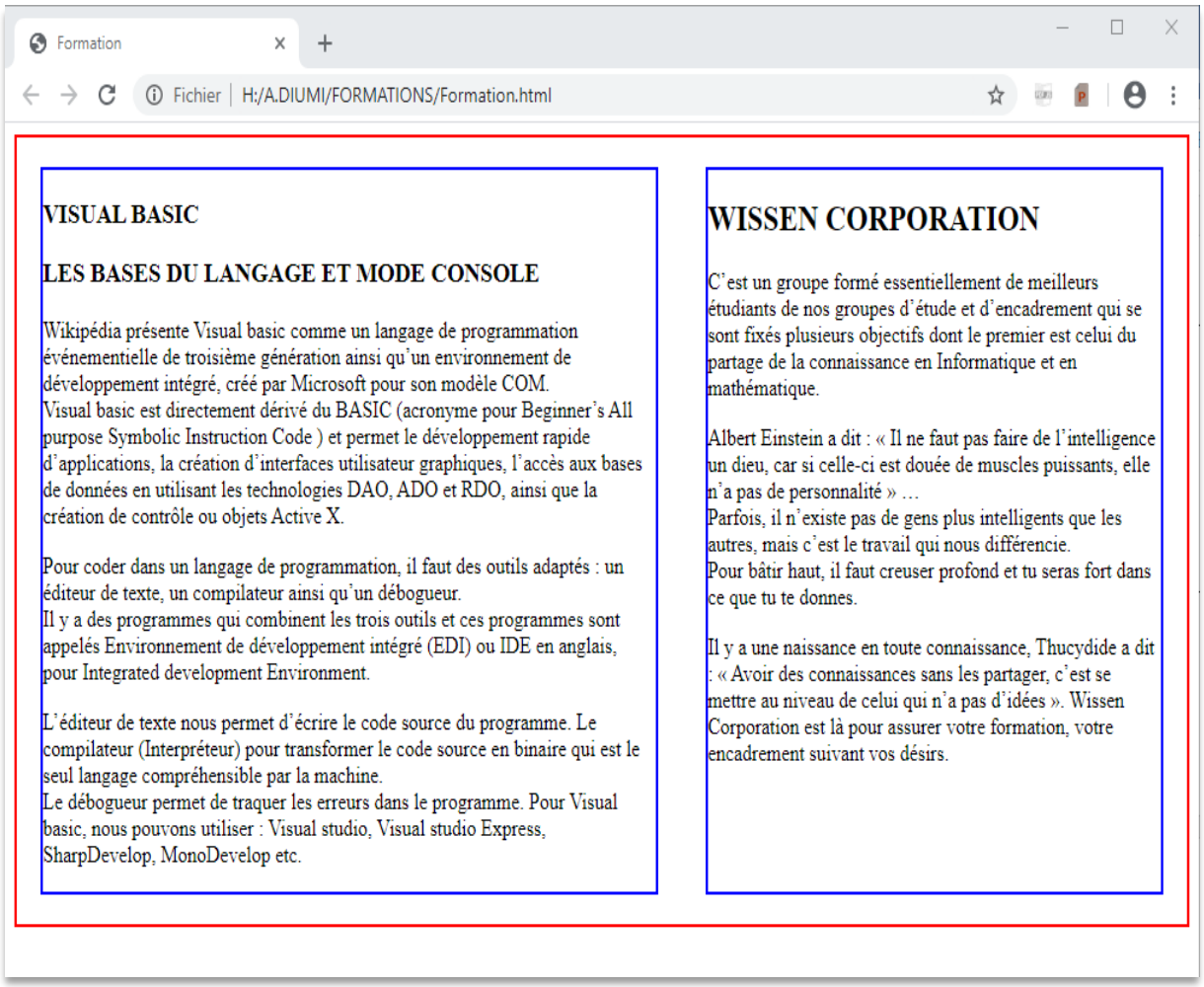
Si nous appliquons ***display : flex*** au bloc parent, les sous-bloc (les enfants) seront repartis en colonnes. Comme dans notre exemple, les deux sous-blocs seront repartis en colonne, c'est-à-dire seront positionnés l'un à côté de l'autre.

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 forcement toute la largeur.

Pour chaque sous-bloc, la largeur peut être gérée par la propriété ***flex*** :

```
.sousBloc
{
    flex: nombre;
}
```

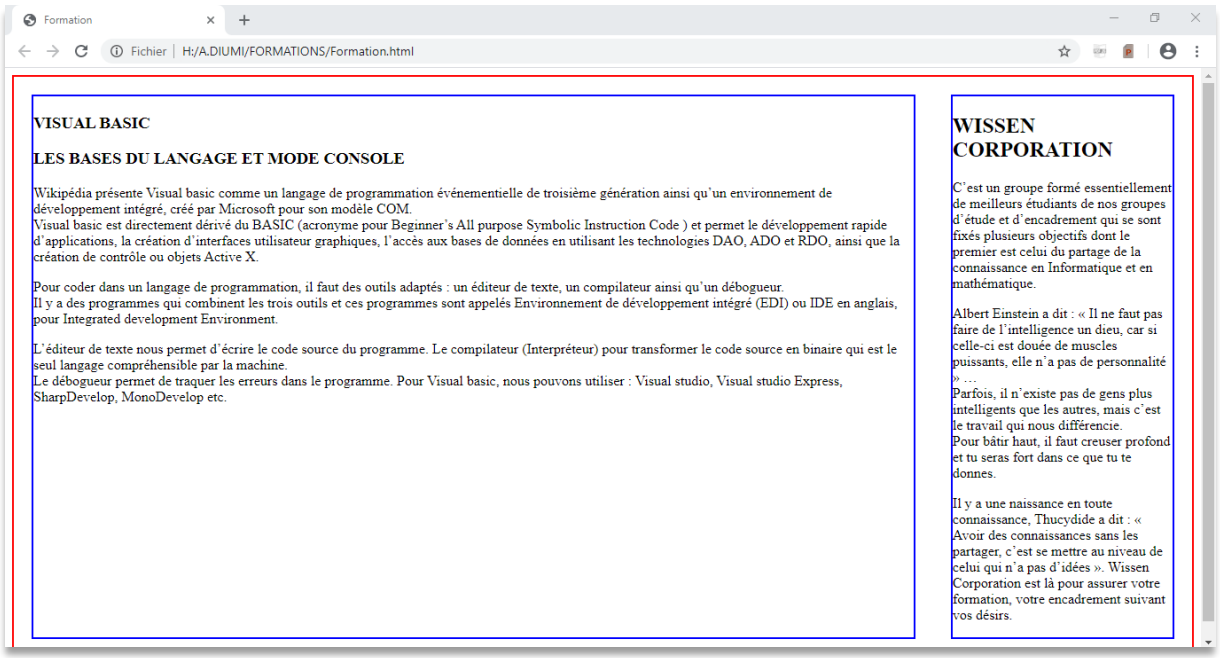
Le nombre donne le rapport de largeur (ou de hauteur) entre les sous-blocs. En mettant 1 à tous les sous-blocs, ils seront tous de la même taille.

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 $^4/_5$ au premier sous-bloc et le reste ($^1/_5$) 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.2 Propriétés liées à 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 progra
mme.
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>
```

III.2.1 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.

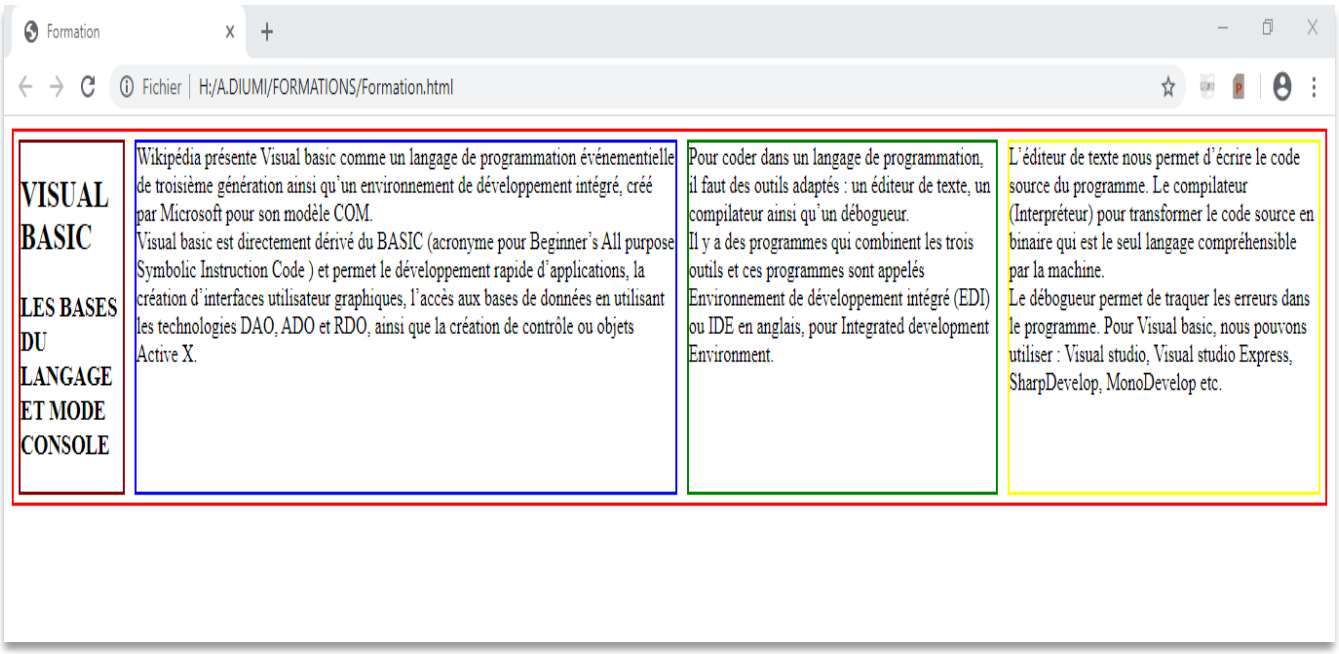
Forme.css

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…

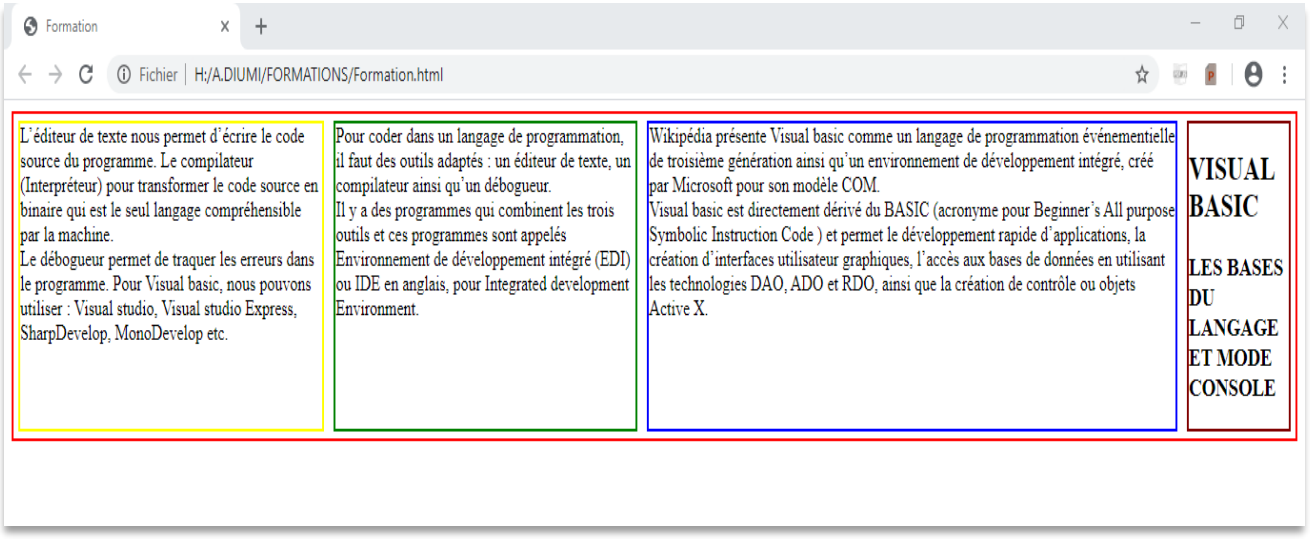
- ***row-reverse*** : alignement les sous blocs verticalement (sur une ligne) en commençant de la droite vers la gauche.

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 :



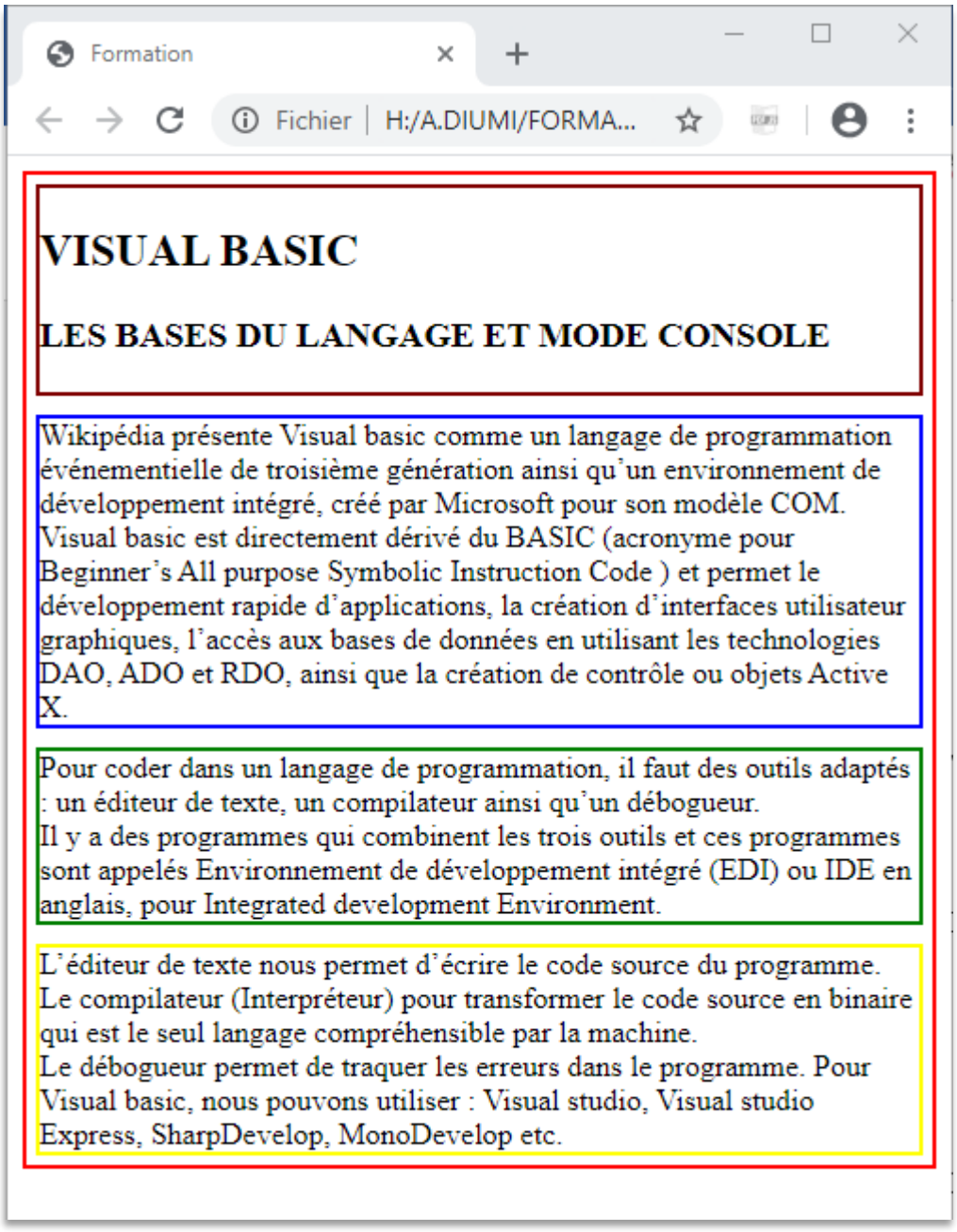
Vous l'avez sans doute remarqué : le premier sous-bloc est placé à droite.

- ***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 :



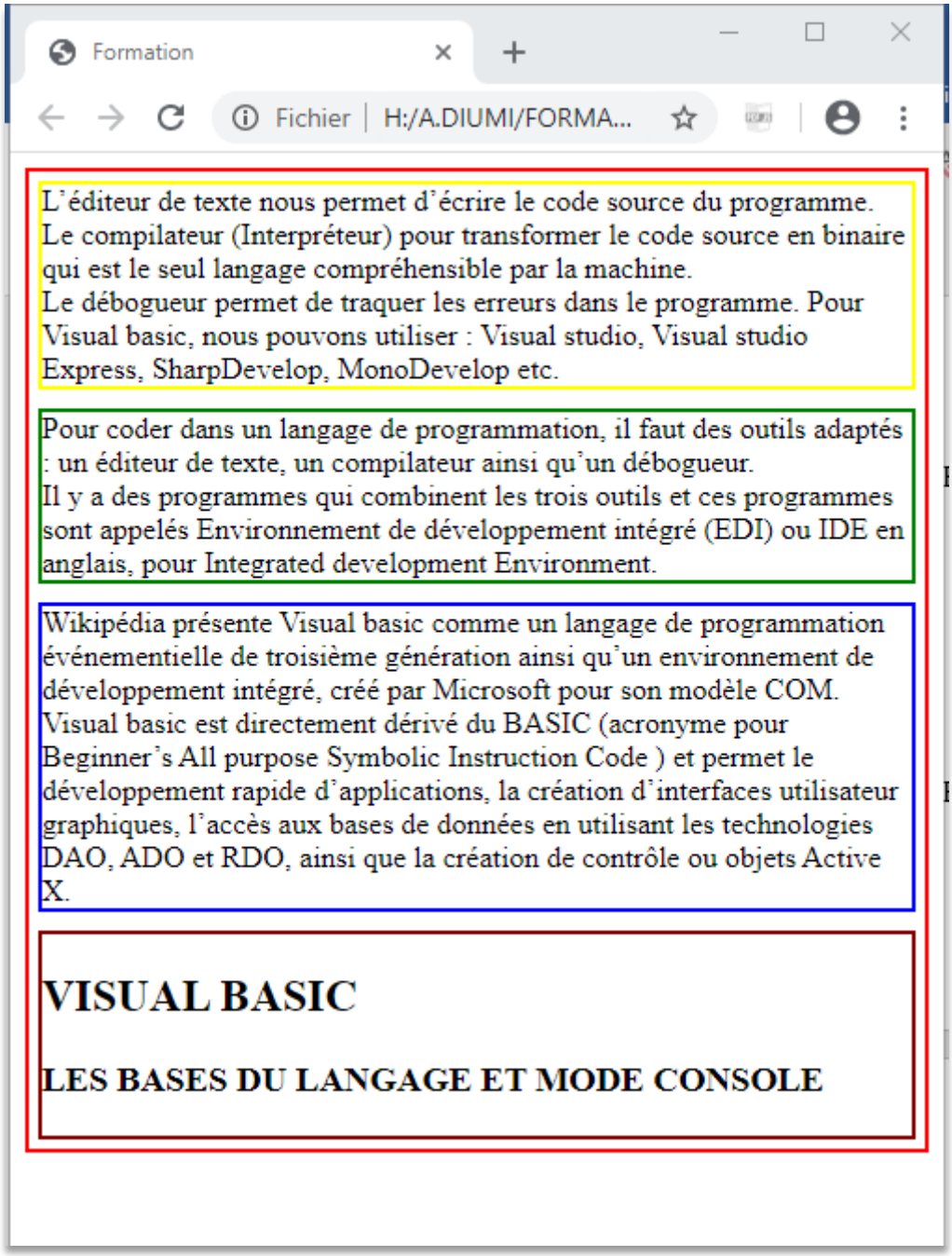
- ***column-reverse*** : positionne les sous blocs sur une colonne en partant du bas vers le haut.

Forme.css

```
.bloc
{
display: flex;
flex-direction: column-reverse;
border: 2px red solid;
```

```
}
```

Résultat :


Formation
Fichier | H:/A.DIUMI/FORMA...
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.
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.
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.
VISUAL BASIC
LES BASES DU LANGAGE ET MODE CONSOLE

III.2.2 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.

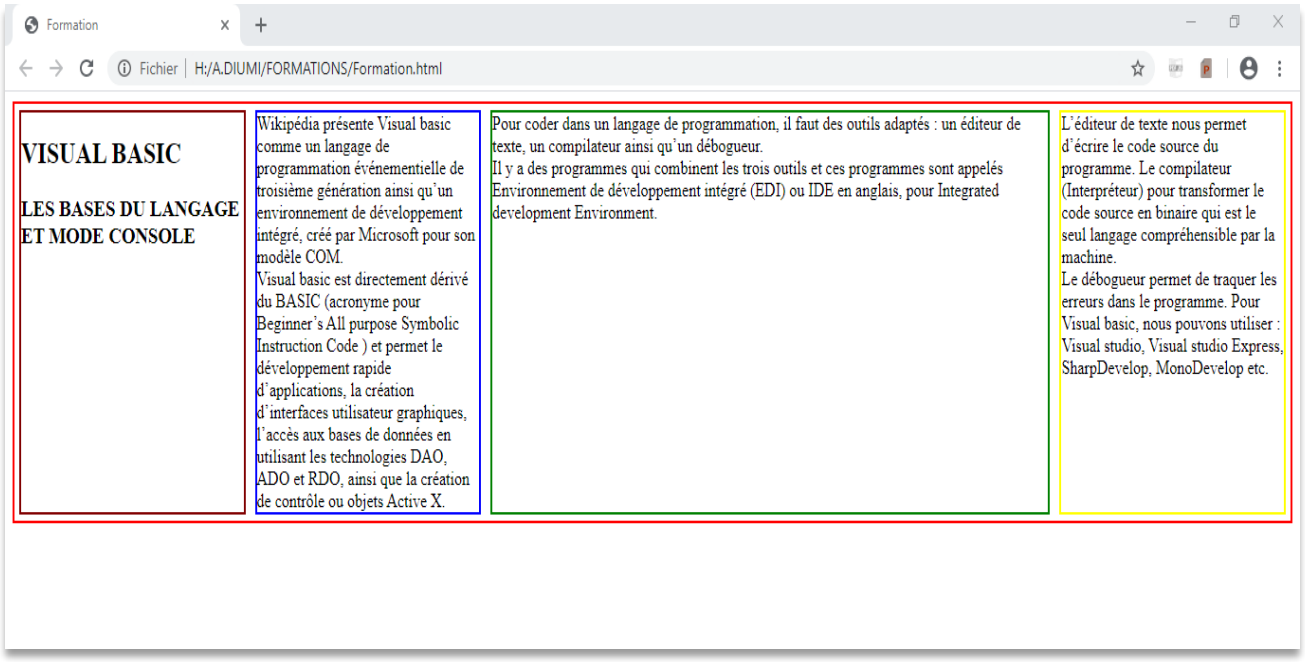
Avec ***nowrap*** :

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.

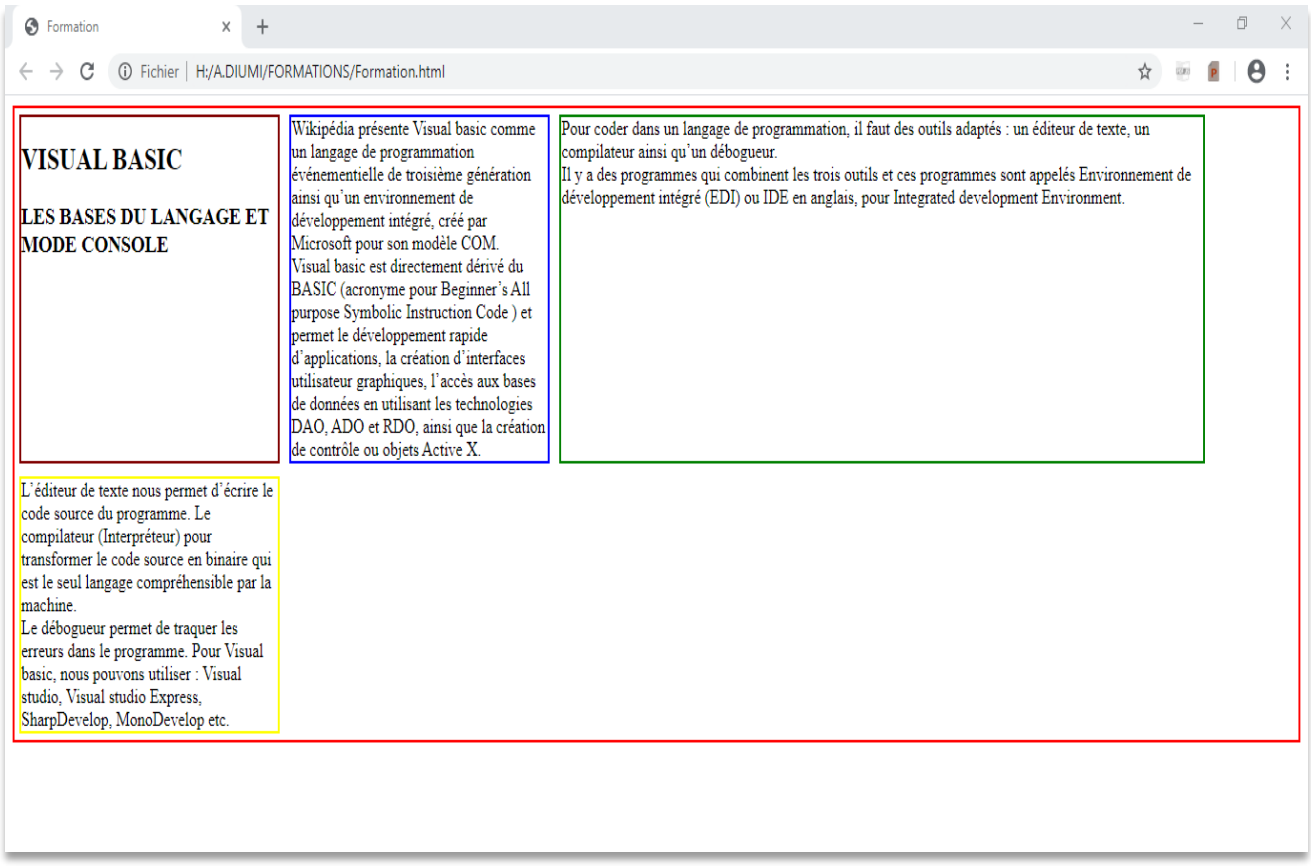
Avec wrap :

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.

III.2.3 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

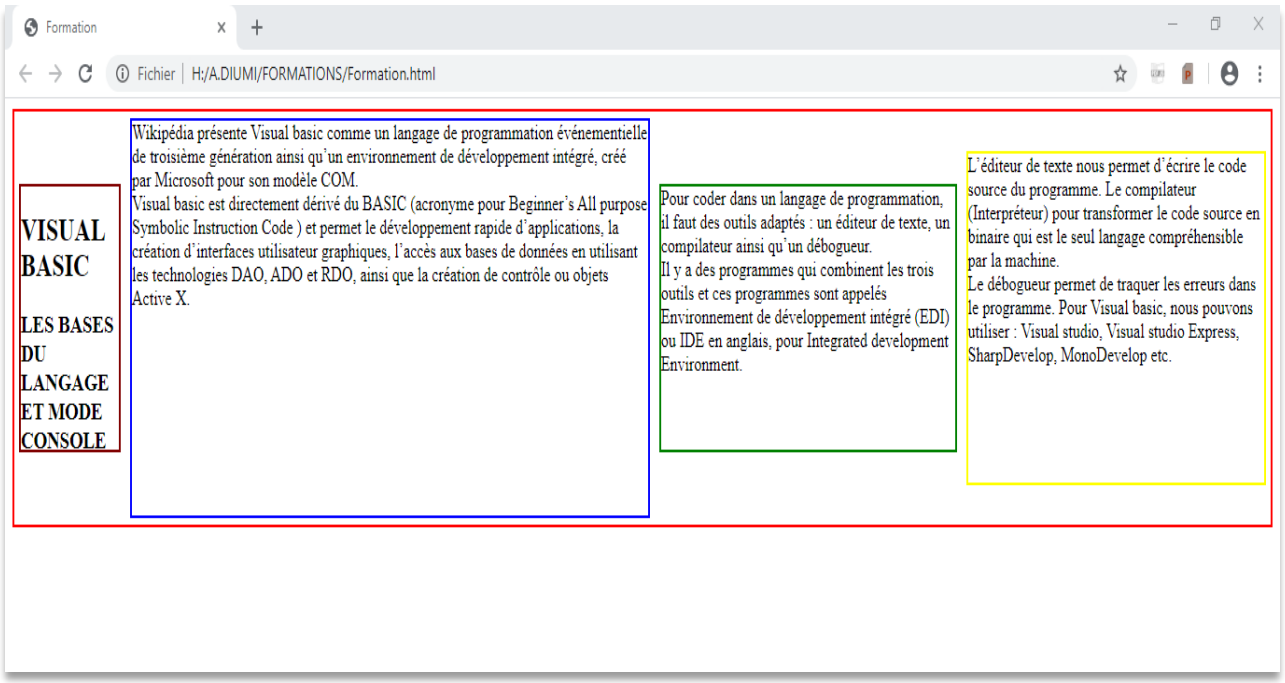
Exemple1

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 :



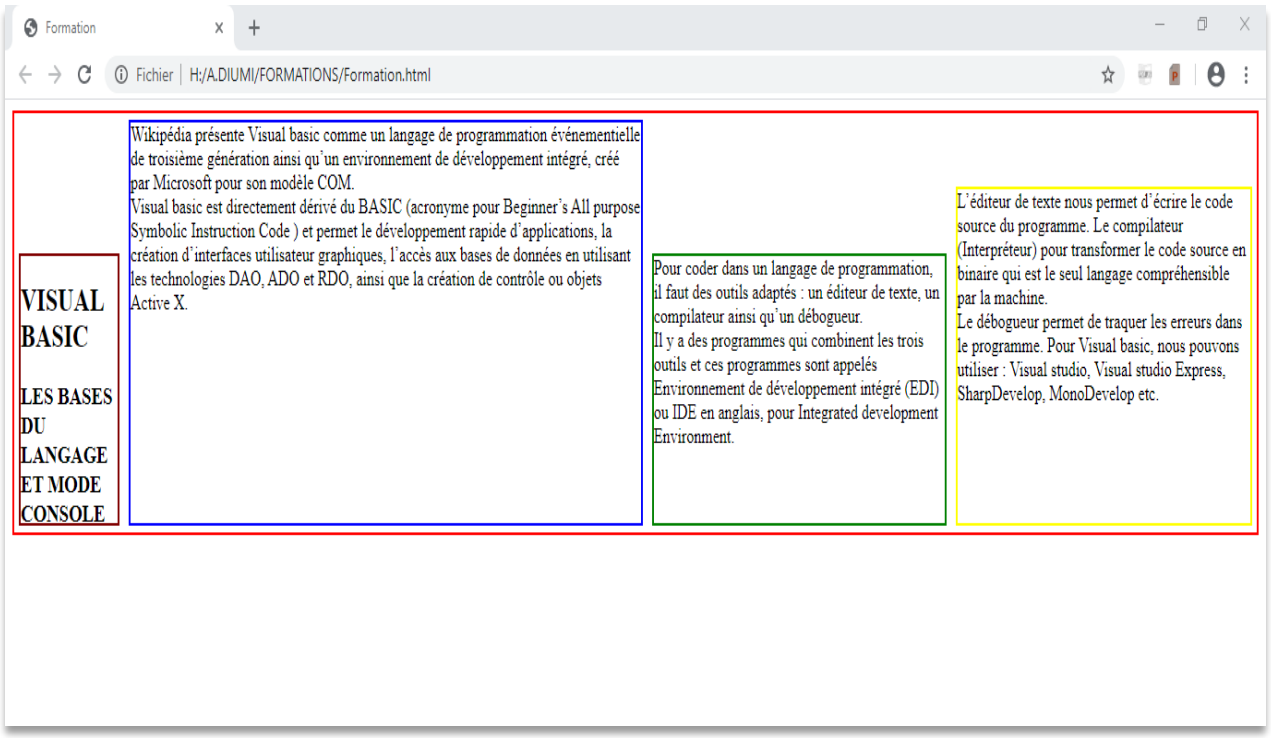
Exemple 2

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 :

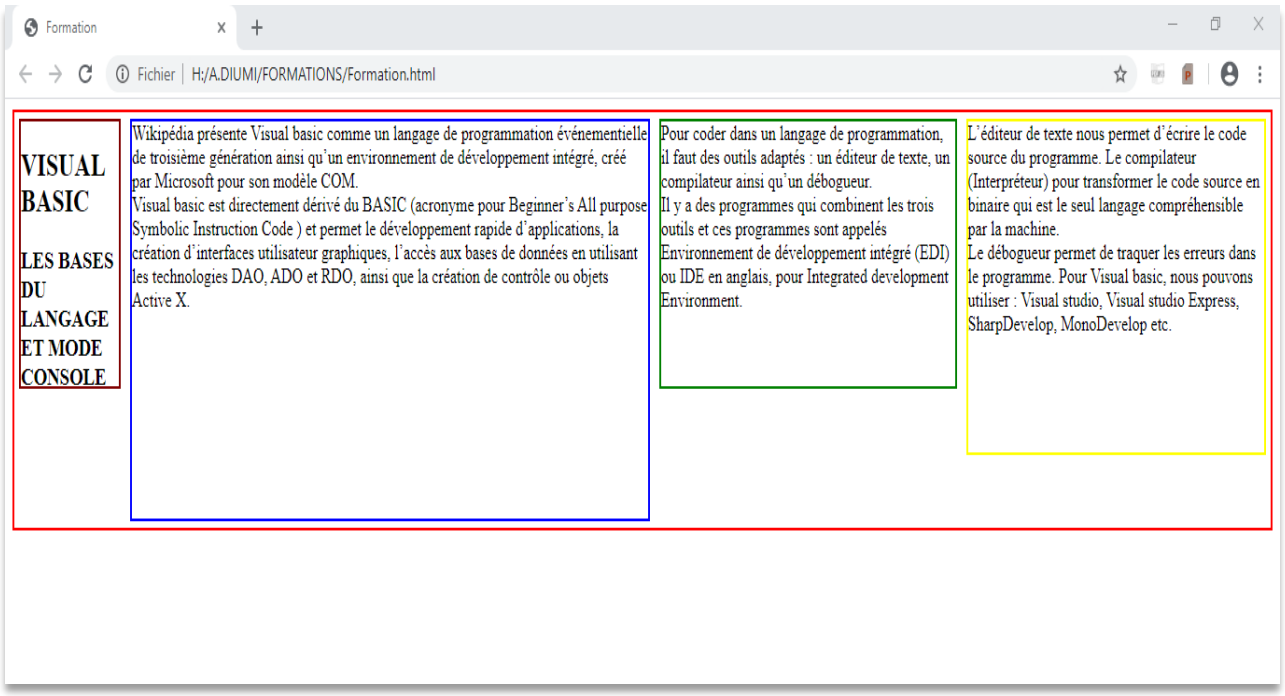


Exemple3

Forme.css

```
.bloc
{
display: flex;
align-items: flex-start;
border: 2px red solid;
}
```

Résultat :



III.2.4 La propriété order

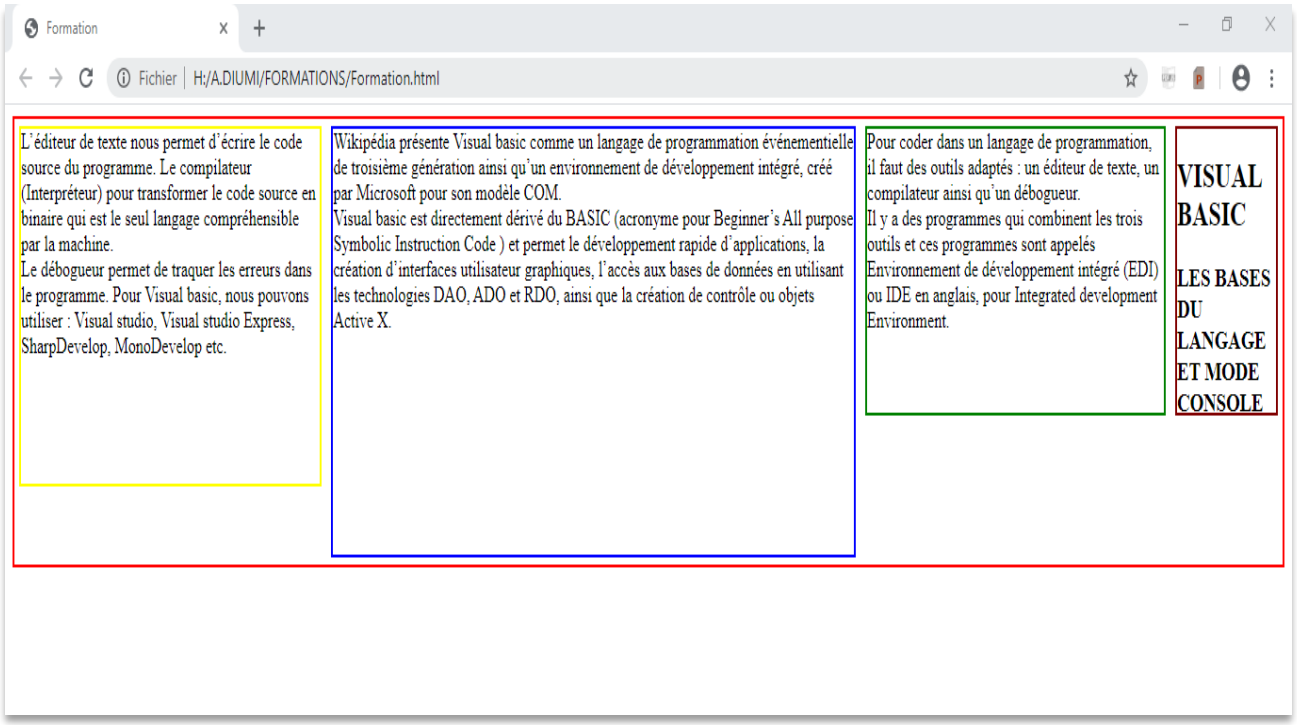
Cette propriété permet de choisir l'ordre d'affichage de sous blocs quel que soit l'ordre dans html. Dans notre exemple, le premier sous bloc est le titre, donc c'est lui qui s'affiche le premier par défaut, changeons l'ordre d'affichage :

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.

IV. LE POSITIONNEMENT ABSOLU

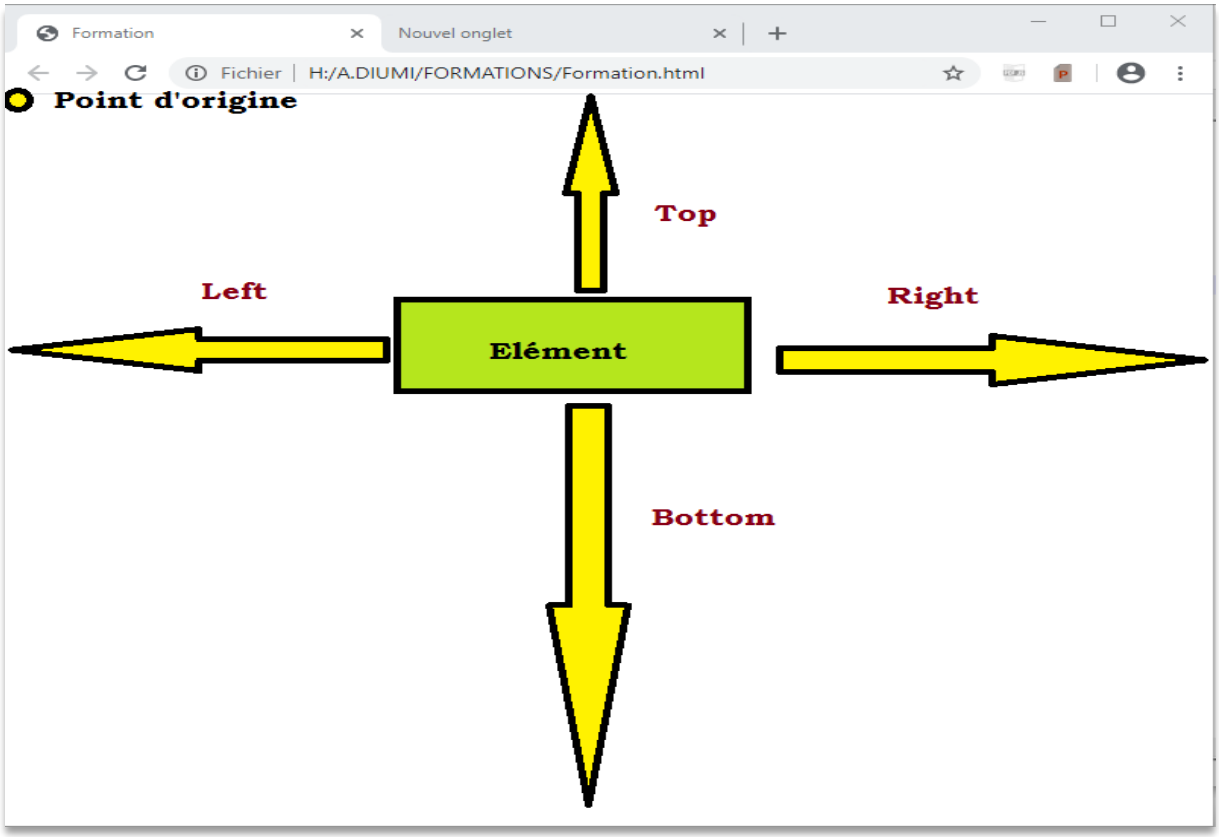
Le positionnement absolu nous permet de placer un élément n'importe où dans la page web. On utilise la propriété position à laquelle on attribue la valeur absolute comme ceci :

```
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.



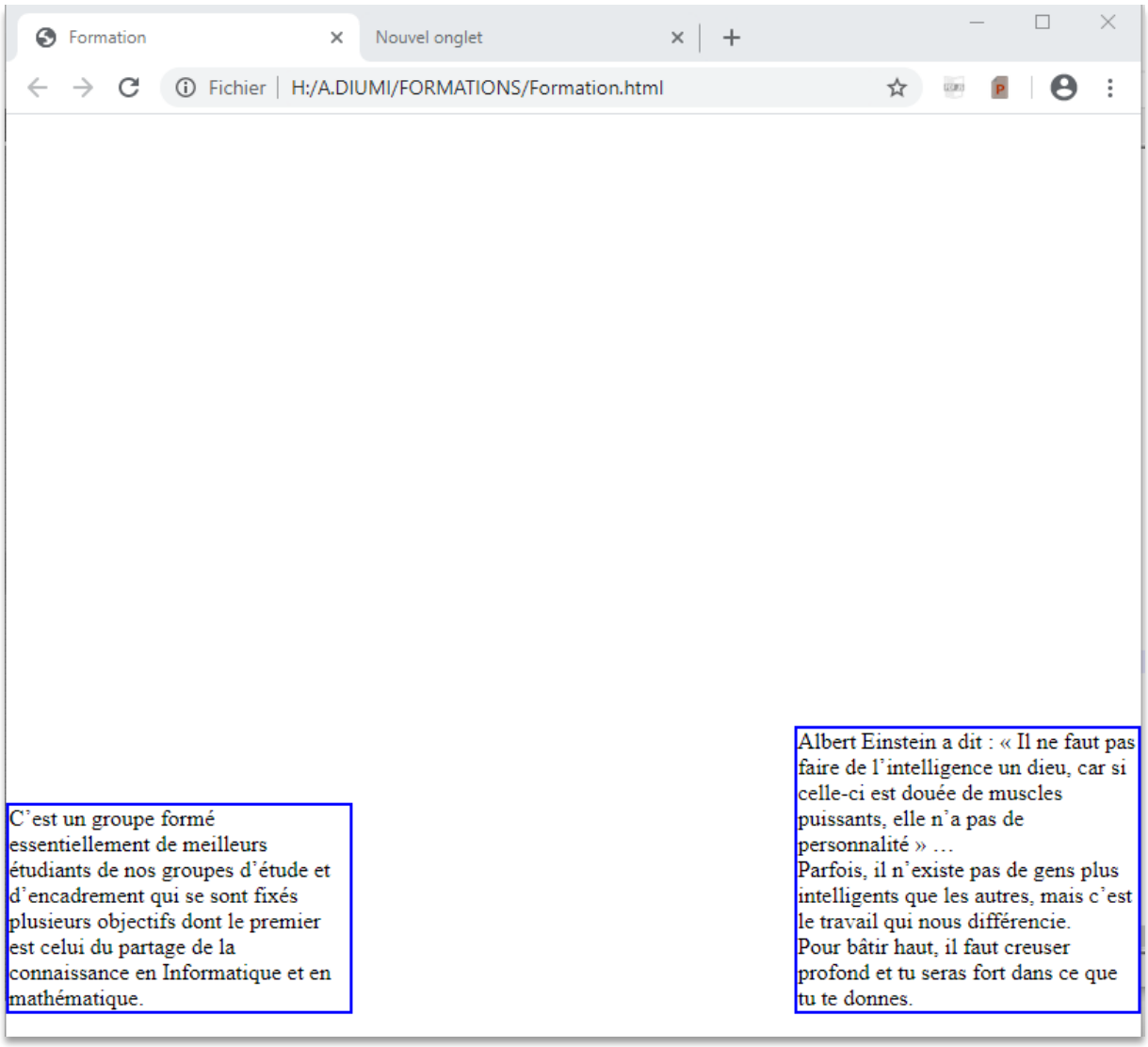
Maintenant, vous libre de positionner votre élément partout où vous voulez en donnant au moins une valeur à ces quatre propriétés.

Reprenons l'exemple de deux paragraphes, si j'ai besoin que le premier se place au coin inférieur gauche et le deuxième au coin inferieur 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 :



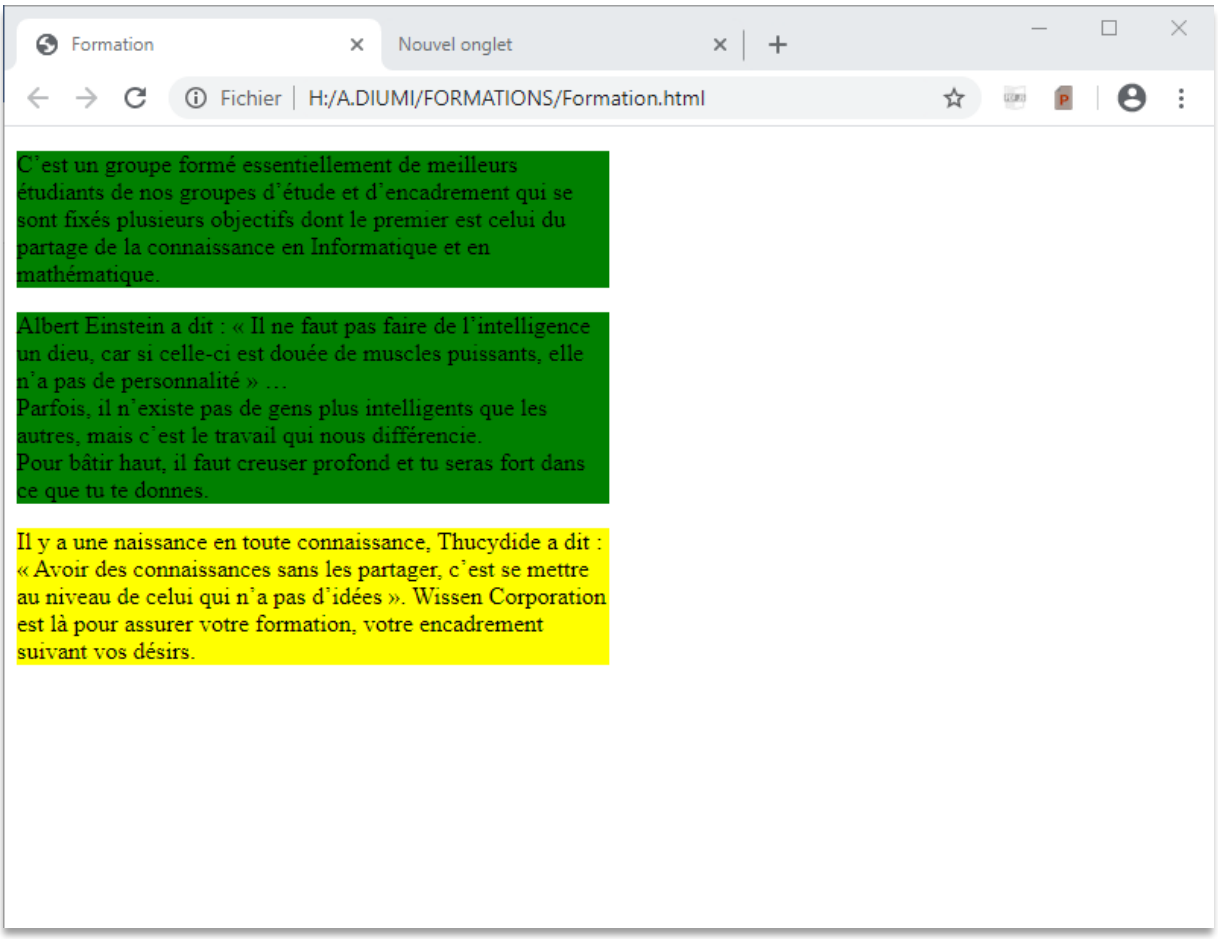
Les éléments positionnés en absolu sont placés par-dessus le reste des éléments de la page. En d'autres termes, un élément positionné en absolu est positionné de façon indépendante de tout texte qui vient avant ou après dans le document, Voyons cela avec un exemple

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 l
e premier est celui du partage de la connaissance en Informatique et en mathémati
que. </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 donne
s. </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 celui qui
 n’a pas d’idées ». Wissen Corporation est là pour assurer votre formation, 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 :

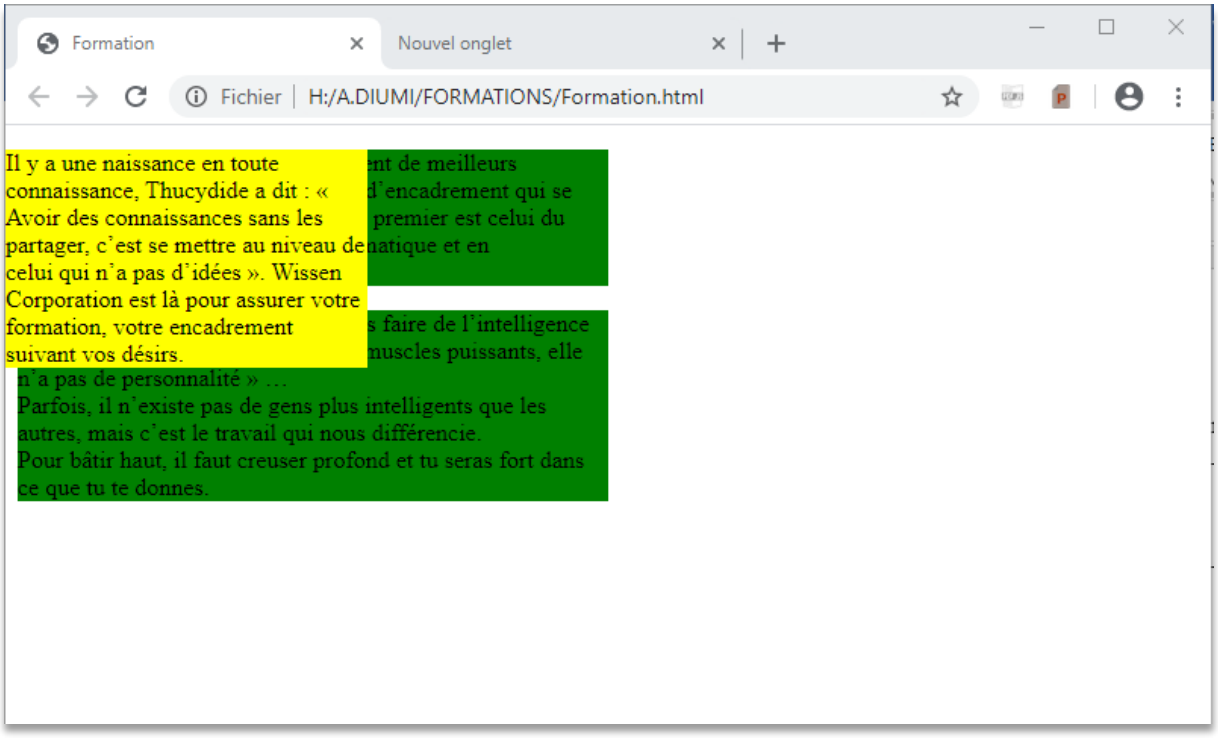


Positionnons le dernier paragraphe (ayant un fond jaune) en absolu en le plaçant au coin supérieur gauche qui revient par défaut au premier paragraphe :

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.

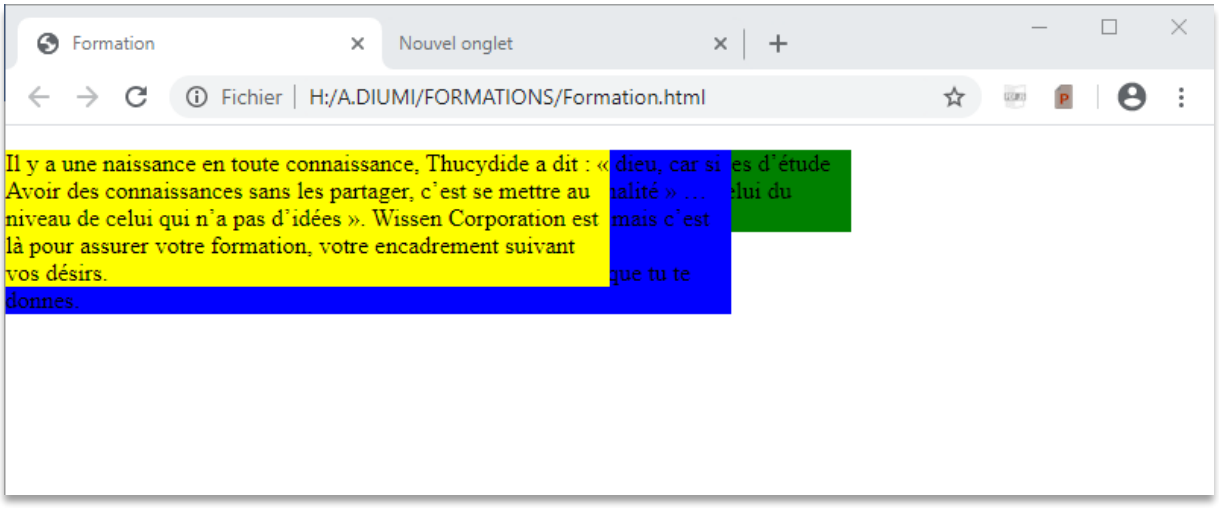
Par ailleurs, si vous placez deux éléments en absolu vers le même endroit, ils risquent de se chevaucher. Dans ce cas, utilisez la propriété ***z-index*** pour indiquer quel élément doit apparaître au-dessus des autres. L'élément ayant la valeur de ***z-index*** la plus élevée sera placé par-dessus les 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.

V. 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>
```

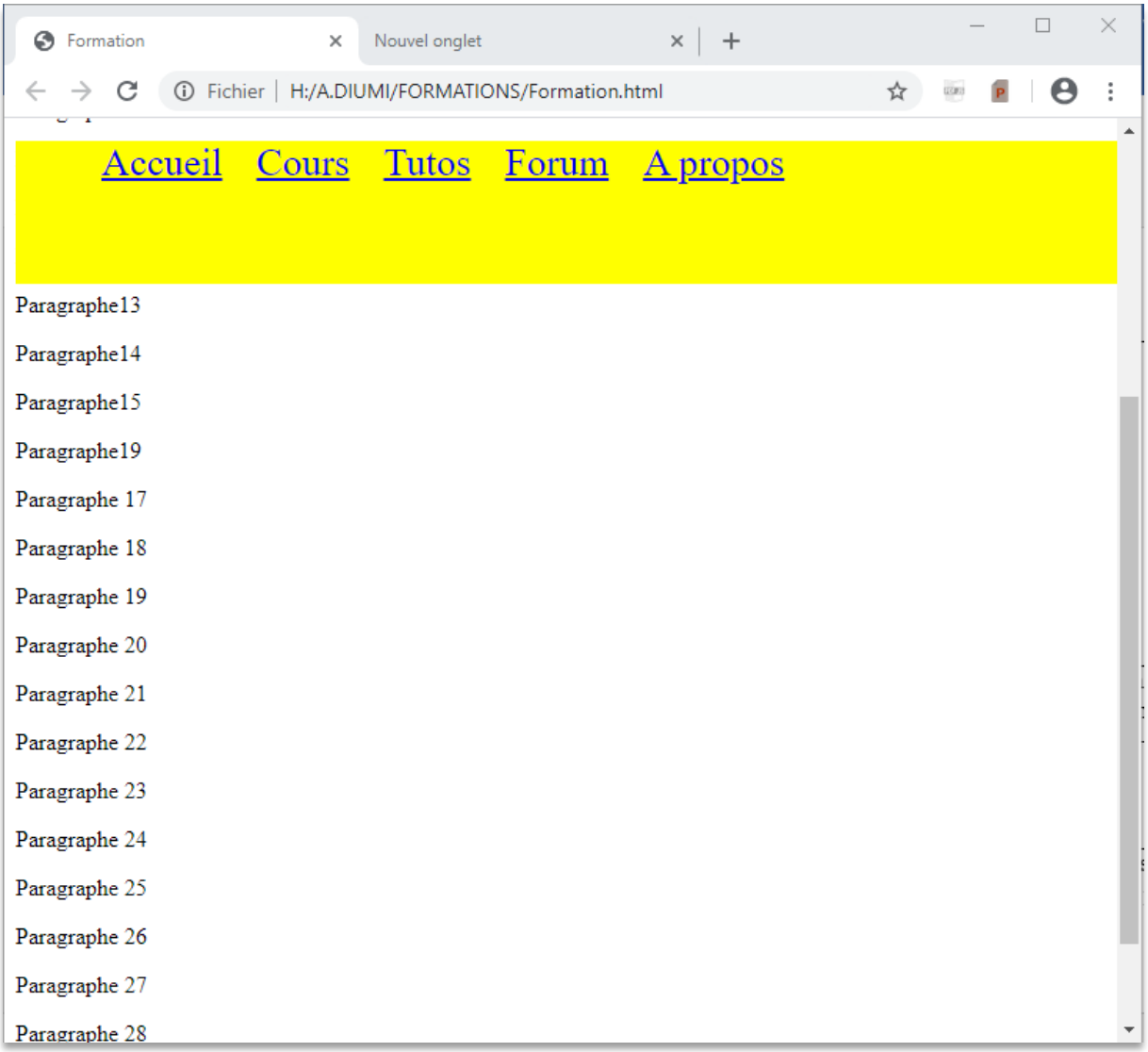
Forme.css

```
ul
{
    list-style-type: none;
    background-color:yellow;
    height: 100px;
}

li
{
    display: inline-block;
    margin-left: 20px;
    font-size: 1.7em;
}
```

J'aimerai 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).



VI. 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 :

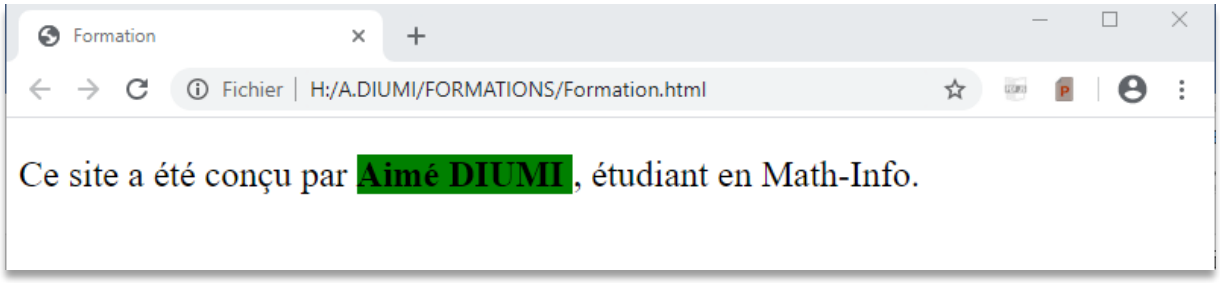
Formation.html

```
<p> Ce site a été conçu par <span class="nom"> Aimé DIUMI </span>, étudiant en Ma
th-Info. </p>
```

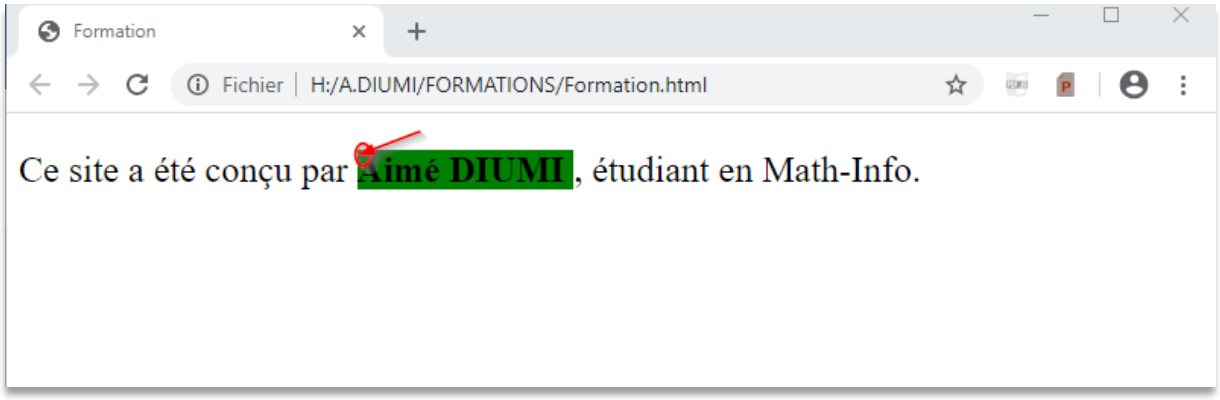
J'ai ajouté une couleur de fond au nom Aimé DIUMI pour qu'il soit visible :

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 :

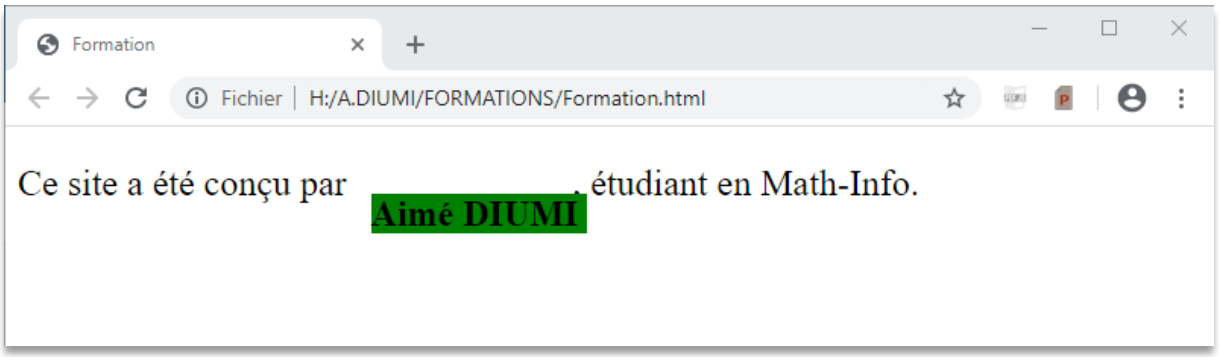


Si on applique les propriétés top, left, right et bottom, l'élément va se déplacer par rapport à cette position. Essayons de tester cela avec le décalage suivant :

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.

VII. POSITIONNEMENT MULTI COLONNE

Le module de positionnement multicolonnes (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 :