

# RÉALISER UN SITE WEB AVEC HTML5 ET CSS3

## 1. Un site internet « moderne » : accessible et adaptable.

Un site « web » moderne doit concilier plusieurs caractéristiques : être accessible à tous (quel que soit le navigateur, quelle que soit la personne, quel que soit le média de visualisation), être facile à référencer par les moteurs de recherche (sémantique de qualité) et facile à mettre à jour ou faire évoluer.

La solution existe : HTML (HyperText Markup Language) pour le contenu (l'information brute), aujourd'hui dans sa version 5, et CSS (Cascading Style Sheet) pour la mise en page et le design (CSS3 aujourd'hui).

## 2. Qu'est-ce qu'un site internet ?

Il se compose d'un ensemble de pages (qui sont des fichiers texte enregistrés au format HTML de suffixe .htm) liées entre elles par des **liens hypertexte**.

Un fichier HTML décrit le contenu d'une page du site ; le navigateur (Firefox, IE, Safari, Opera, etc.) télécharge puis « lit » ce fichier, en interprète les instructions et reconstitue la page (en téléchargeant aussi les fichiers liés : images, fontes, etc.)

Chaque navigateur peut interpréter à sa façon le code HTML, ce qui explique les différences de rendu.

Le **W3C** essaie de normaliser les instructions pour limiter les écarts d'interprétation. Le langage évoluant, certaines instructions ne sont pas encore implémentées dans tous les navigateurs.

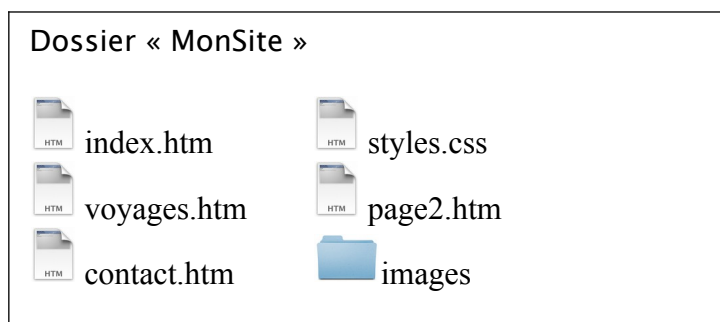
## 3. Où placer ses fichiers ?

Toutes les pages, images, feuilles de style doivent être placées dans un dossier unique (dossier « racine ») pour une organisation simple et cohérente.

Le fichier HTML, ouvert automatiquement par un navigateur (lorsque le site est sur un serveur distant en particulier), doit être intitulé `index.htm` ; tous les autres documents seront reliés à ce fichier de référence par des liens.

Pour éviter tout problème avec les liens, on utilisera uniquement des liens relatifs (et non pas absolus) dans le code HTML.

*Exemple :* si dans le code de la page `index.htm` on a l'instruction suivante : `<a href='images/photo1.jpg'>`, le lien mènera vers l'image intitulée « *photo1.jpg* » qui se trouve dans le dossier « *images* » qui doit être dans le même dossier `index.htm`, quel que soit l'endroit où se trouve ce dossier. C'est un lien relatif (relatif à l'emplacement de `index.htm`). Les liens absolus seront réservés aux liens externes (par exemple `<a href='http://www.tonsiteweb.com/index.htm'>`).



## 4. Exemples types de fichiers HTML5 et CSS3

### Un fichier HTML5

```
<!DOCTYPE html>

<html>
<!-- N'apparaît pas sur la page web -->
<meta charset="utf-8" />
<title>Page 1</title>
<!-- S'affiche dans la barre de fenêtre -->
<link rel="stylesheet" type="text/css" href="styles.css">
</head>

<body>
  <div id="wrapper">
    <!-- L'enveloppe de la partie centrale de ma page -->
    <header>
      <h1>Mon Site Web</h1>
    </header>
    <article>
      <p>Hello World</p>
      <p class="centre">C'est mon site !</p>
      <p><a href="page2.htm">Cliquez ici</a> pour voir la page 2</p>
      
    </article>
    <footer>© 2013</footer>
  </div>
</body>

</html>
```

**Balise** (pointe vers `<html>`)

**Commentaire** (pointe vers `<!-- N'apparaît pas sur la page web -->`)

**Paragraphe** (pointe vers `<p>Hello World</p>`)

**Lien hypertexte** (pointe vers `<a href="page2.htm">Cliquez ici</a>`)

### Un fichier CSS3

```
#wrapper h1
{
  font-family : Georgia, serif ;
  color : #000; /*Les titres h2 seront en noir */
  font-size : 25px;
  text-shadow: 0px 1px 1px black, 0px 0px 10px black;
}

.centre
{
  text-align : center; /*Le texte sera centré avec la classe .centre*/
}
```

**Sélecteur** (pointe vers `#wrapper h1`)

**Commentaire** (pointe vers `/*Les titres h2 seront en noir */`)

**Propriété** (pointe vers `font-family`)

**Sélecteur de** (pointe vers `.centre`)

#### Mon Site Web

Ma vie au lycée Janot

**SPECIAL I.S.N**

- Accueil
- Mes livres
- Contact
- No CSS

#### Accueil

Bonjour !



Une page WEB sans CSS (gauche) et avec CSS (droite)

## 5. Quelques liens utiles

**<http://www.w3.org>**

Le site du World Wide Web Consortium : la référence de la standardisation du web.

**<http://validator.w3.org/>**

Pour vérifier si le site créé respecte bien les standards du W3C.

**<http://html5doctor.com/>**

Le glossaire (glossary) recense les balises de HTML5.

**<http://www.w3schools.com/css3/default.asp>**

Les références des sélecteurs CSS

**<http://www.alsacreations.com/>**

Tutoriels et articles intéressants.

**<http://www.proftnj.com/RGB3.htm>**

Pour choisir des couleurs et connaître leur code.

**<http://fr.openclassrooms.com>**

Des tutoriels et des cours, entre autres sur HTML et CSS.

## 6. Réalisation du site : le contenu (fichier HTML)

### 6.1. Structure du fichier HTML

Le fichier HTML comporte tout le **contenu** du site : les informations proprement dites, le contenu lisible et informatif (y compris les images) mais **pas de données sur le design** qui sera géré séparément par les « CSS » (voir plus loin).

La 1<sup>ère</sup> information à donner en début de document HTML est le **doctype** : il annonce au navigateur quelle version de HTML est utilisée. `<!DOCTYPE>` pour du HTML version 5. La syntaxe du fichier HTML est basée sur des **balises** (« tags ») qui structurent le document.

Elles vont par deux (sauf exceptions), pour ouvrir et fermer une section.

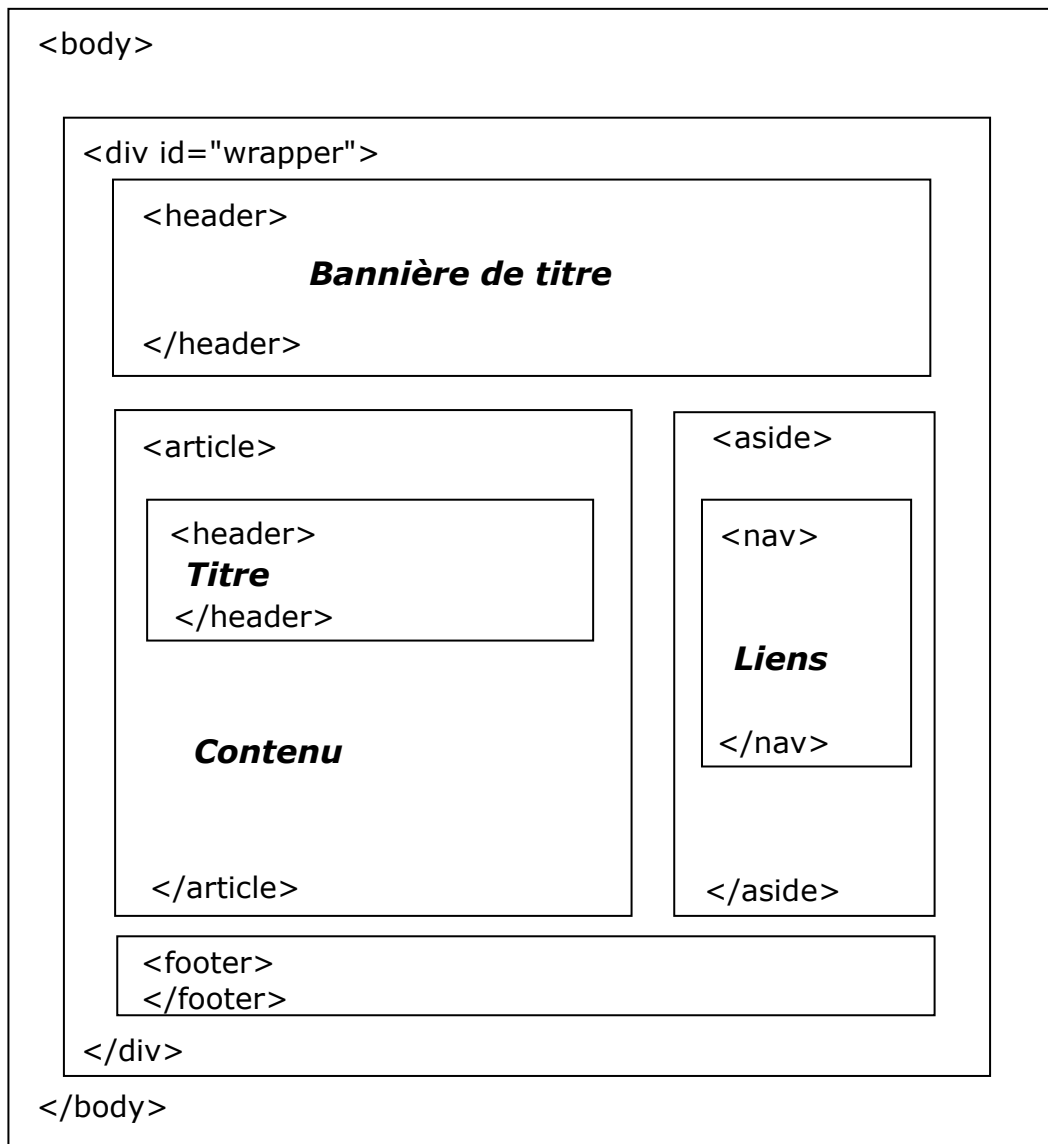
- Le document doit se poursuivre par `<html>` (il se terminera par `</html>` ; les balises ouvrante et fermante se différencient par le « / »).
- `<head>` et `</head>` contiennent des informations pour le navigateur et qui n'apparaîtront pas sur la page web :
  - L'encodage : il précise quel jeu de caractère est utilisé (accents, glyphes spéciaux) `<meta charset= "utf-8">` ;
  - Les infos destinées aux moteurs de recherche (« meta balises ») ;
  - La désignation des ressources externes (feuilles de style CSS, fontes, etc.)
- `<body>` et `</body>` qui contiennent tout le contenu affiché sur la page web.

### 6.2. Balises de contenu

Chacune apporte des informations hiérarchiques précises et joue un rôle précis dans la logique du site et dans sa sémantique. Elles permettent un référencement précis et facilitent la "lisibilité" du code.

Une page web peut être considérée comme un empilement de "boîtes" ou divisions ; chacune d'elle est délimitée par des balises.

Exemple de structure :



- `<header></header>` : regroupe l'introduction à un contenu (titres par ex.)
- `<aside></aside>` : éléments qui ne sont pas en rapport avec le contenu informatif du site. Par exemple des menus, de la publicité.
- `<nav></nav>` : contient les éléments dédiés à la navigation dans le site (liens internes).
- `<article></article>` : élément de contenu qui pourrait être extrait du site sans perdre de sens; c'est le "contenu brut" qui apporte une information.
- `<footer></footer>` : "pied de page" (liens, crédits, mentions légales etc.)
- `<div></div>` : balise générique qui délimite une "boîte" quand aucune autre balise ne se justifie.

A l'intérieur de ces balises « structurantes », le contenu est lui-même organisé sous forme de balises.

Il y a deux familles de balises : les balises de type « **bloc** » (block) et les balises de type « **ligne** » (inline).

Les éléments des balises de type **bloc** sont par défaut placés l'un sous l'autre par le navigateur. C'est le cas par exemple des balises de paragraphe `<p>` : deux paragraphes successifs seront affichés l'un au dessous de l'autre. Cette particularité permet de prévoir des dimensions (hauteur, largeur, marges internes appelées *padding*) et de choisir relativement facilement un placement particulier sur la page web apparente.

Les éléments **en ligne** sont prévus pour « enrichir » une partie d'un texte (lien, renforcement etc.) et ne modifient pas la structure du texte, ou bien pour constituer un menu sur une seule ligne par exemple.

Balise de type « ligne » ( <i>inline</i> )	Balise de type « bloc » ( <i>block</i> )
<code>&lt;a&gt;</code> : désigne un lien hypertexte.	<code>&lt;div&gt;</code> : conteneur générique
<code>&lt;img&gt;</code> : inclut une image dans le document	<code>&lt;h1&gt;</code> , <code>&lt;h2&gt;</code> , ..., <code>&lt;h6&gt;</code> : les six niveaux de titres, hiérarchiquement organisés entre eux. Un titre <code>&lt;h2&gt;</code> sera affecté à un titre de niveau hiérarchique inférieur à un titre <code>&lt;h1&gt;</code>
<code>&lt;em&gt;</code> : met en emphase une partie de texte	<code>&lt;ul&gt;</code> : listes à puces (ou sans puces !)
<code>&lt;strong&gt;</code> : met en gras une partie de texte	<code>&lt;p&gt;</code> : paragraphe de texte.

Les instructions données dans le fichier HTML ne comportent pas d'informations sur le **design** (mise en page, apparence, couleurs, typographie etc.) ; ce n'est pas son rôle, qui doit se limiter au **contenu informatif**.

## 7. L'aspect du site : le design (fichier CSS)

En l'absence de précisions sur le design (mise en page, couleurs, fontes etc.), les pages d'un site sont toutes semblables et sans intérêt graphique. Les **feuilles de style en cascade** (CSS) permettent de spécifier tous les attributs de mise en page **sans toucher au contenu**. L'intérêt est multiple : on pourra personnaliser des feuilles de style suivant l'utilisation fixe (ordinateur de bureau) ou nomade (smartphone, tablette), suivant l'humeur du jour ou les spécificités temporaires (période de soldes pour un site de vente par exemple).

Les CSS listent les propriétés et les attributs de chaque balise utilisée par le code HTML. Le fichier qui contient ce informations est ouvert, lu et stocké en mémoire par le navigateur en début d'interprétation du fichier HTML.

### 7.1. L'hérédité

Dans les feuilles de style en cascade (CSS), toute balise (qualifiée d'*enfant*) hérite des propriétés de la balise **à l'intérieur de laquelle elle se trouve "enfermée"** (celle-ci étant la balise "*parent*"). Si l'on veut une **autre propriété**, il faut la préciser dans le sélecteur de la balise enfant.

Ex. : si on définit la fonte `{font-family: 'Georgia';}` dans le sélecteur *body*, tout le texte de la page sera en *Georgia* sans qu'il soit nécessaire de le préciser pour les autres balises, puisque ces dernières sont "à l'intérieur" des balises `<body>`.

Si on veut une autre fonte pour les paragraphes par exemple, il faudra créer un sélecteur `p {font-family: 'Times', 'Garamond', serif}` qui "prendra la main" sur cette propriété.

### 7.2. La syntaxe des sélecteurs

Syntaxe générale : `selecteurCSS {proprieteCSS: valeurCSS;}`

p, h1, h2 {font-size : 2px;}	Les trois sélecteurs p, h1 et h2 ont les mêmes propriétés
header p {font-size : 2px;}	La propriété concernera les balises <p> qui se trouvent à l'intérieur de la balise <header>
#debut p {font-size : 2px;}	La propriété concernera les balises <p> qui se trouvent à l'intérieur de la balise <div id="debut">
h1+p {margin-top : 5px;}	Sélecteur adjacent. La propriété s'applique uniquement à la balise <p> qui <b>suit immédiatement</b> une balise <h1>.
<b>Classes</b>	
p.penche {font-style : italic;}	La propriété concernera les balises <p class="penche">, et pas les autres.
.penche {font-style : italic;}	Toutes les balises à qui on ajoute class="penche" seront concernées : par ex. <h3 class="penche"> ou <p class="penche">
<b>Pseudo-classes</b>	
a:hover {color : yellow;}	Les liens <a> se colorent en jaune au survol.
a:link {text-decoration : none;}	Le lien pas encore visité n'a pas de soulignement.
a:visited {font-style : bold;}	Les liens déjà visités apparaissent en gras.

## 7.3. Quelques propriétés et valeurs

### 7.3.1. Sélecteurs généraux

Propriété	Valeurs
border-style	Solid, dotted, dashed, double...
border-top-style (bottom, left, right)	Idem
border-color	idem
padding	4 valeurs (top right bottom left)
margin	Idem
margin-top (bottom, left, right)	
background-image	url(.....);
background-color	couleur
float	Right, left

#### Exemple :

```

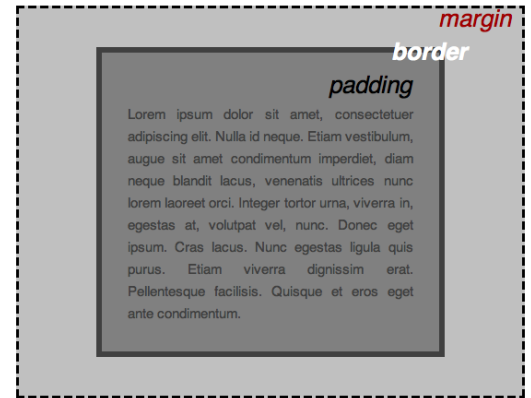
article img {
    margin-top: 30px;
    margin-bottom: 20px;
}
/* toutes les images de la section
   article auront une marge supérieure
   de 30 pixels et inférieure de 20 px */

```

**Remarque 1 :** si les marges sont différentes autour d'un élément, on spécifie les quatre. Exemple : {padding :1px 2px 5px 6px;} signifie que la marge interne haute sera de 1px, la droite de 2px, la basse de 5px et la gauche de 6px.

**Remarque 2 : Padding, border et margin :**

La "boîte" ou conteneur est la partie sombre du centre. Le **padding** et l'espace entre le contenu et la bordure du conteneur, **border** est la bordure elle-même (son épaisseur est paramétrable) et **margin** est l'espace créé autour de la boîte.

**7.3.2. Sélecteurs de type texte**

Propriété	Valeurs
font-family	'Georgia', 'Times', serif, sans...
font-size	unité
font-style	Normal, italic, oblique
font-weight	Normal, bold, light
width	unité
text-align	Left, right, center, justify
text-decoration	None, underline, overline...
background-color	couleur
line-height	unité
text-shadow	couleur, longueur, none
list-style-type (puces)	Disc, square,..., none

**Exemple :**

```
#principal h1 {
    font-family: 'Merienda', 'Georgia', sans;
    color:#fff;
    font-size:20px;
    text-align:center;
}
```

**7.3.4. Les unités**

Unité relative	Définition
%	Valeur en pourcentage
em	Proportion de la taille de la fonte courante. Ex. : 1.2em vaut 1,2 fois la taille standard.
Unité absolue	
px	Pixel
pc (pica)	1 pc égale 12 pt
pt (point)	1 pt = 1/72 <sup>e</sup> de pouce

**Exemple :**

```
#footer {
    width:900px;
    height:20px;
    font-size : 1.5em;
}
```

/\* le conteneur « footer » aura 900 px de large et 20 de haut. Le texte fera 1,5 fois la taille définie dans le conteneur parent \*/

### 7.3.5. Quelques propriétés spécifiques à CSS3

<b>Ombres</b>		
Box-shadow	<code>2px 2px 2px black inset;</code>	Les deux premières valeurs sont le décalage horizontal et vertical (obligatoires), les deux autres sont le flou et la couleur (facultatifs) ; <code>inset</code> crée une ombre interne à la boîte (externe par défaut).
Text-shadow	<code>2px 2px 2px black;</code>	idem
<b>Coins arrondis</b>		
Border-radius	<code>10px 10px 10px 10px</code>	Rayons en partant de l'angle en haut à gauche, puis dans le sens horaire.
<b>Transparence</b>		
opacity	<code>0.5</code>	La valeur doit être comprise entre 0 (totalement transparent) et 1 (opaque).

### 7.3.6. Les fontes autres que les fontes de l'utilisateur

Le navigateur peut télécharger, à l'ouverture de la page, une ou plusieurs fontes qui ne sont pas présentes dans le système de l'utilisateur. Ceci nécessite donc une connexion internet, ce qui en limite l'utilisation en « interne ».

Exemple : **<http://www.google.com/webfonts>**

Après avoir choisi la fonte, cliquer sur "use" et recopier les lignes de code HTML (entre les balises `<head>`) et CSS (dans la feuille de style), comme indiqué sur le site.



## 7.4. La mise en page

Le positionnement des conteneurs est au centre des préoccupations en CSS. Il en existe plusieurs : dans le flux, relatif, absolu, fixe et flottant.

### Le positionnement « dans le flux » :

Dans les conditions « normales », chaque nouveau bloc (défini par une balise de type « bloc », `<p>` par ex.) succède au précédent en se plaçant juste au dessous. Les blocs occupent toute la largeur possible dans leur conteneur.

Chaque élément « en ligne » se place sur la même ligne que ce qui l'entoure.

On dit que les conteneurs se placent dans le flux courant : les uns après les autres en fonction de l'ordre avec lequel ils apparaissent dans le fichier HTML.

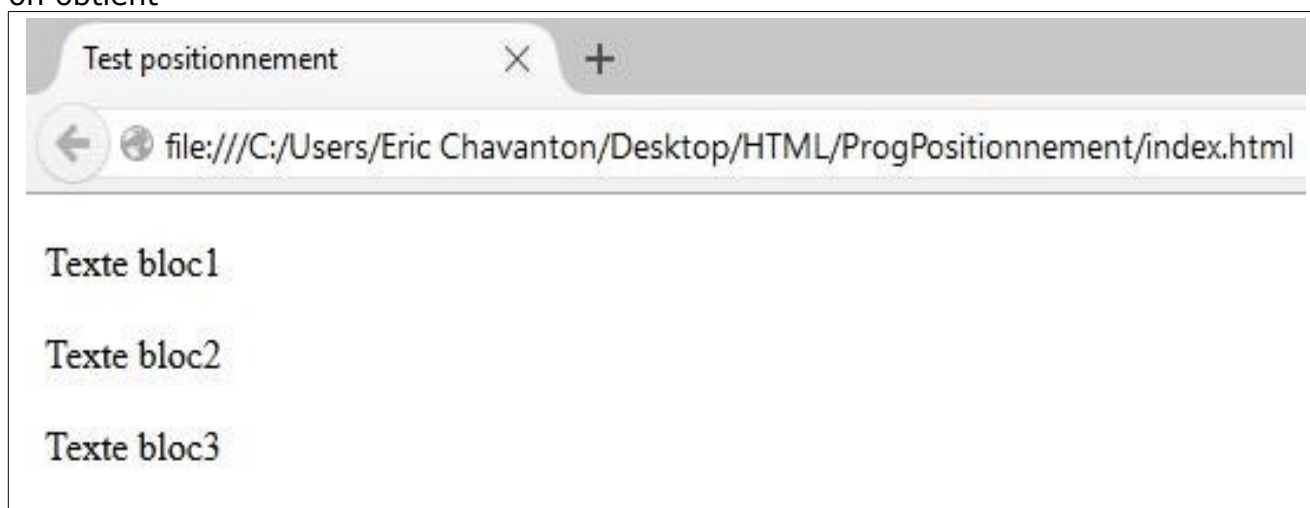
Exemple :

Le fichier HTML se résume à :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" /> <!--définition des caractères internationaux-->
    <title>Test positionnement</title> <!--s'affiche dans la barre de la fenêtre-->
    <link rel="stylesheet" type="text/css" href="styles.css">
  </head>

  <body>
    <p>Texte bloc1</p>
    <p>Texte bloc2</p>
    <p>Texte bloc3</p>
  </body>
</html>
```

En positionnement dans le flux, c'est à dire sans instruction dans le fichier CSS associé on obtient



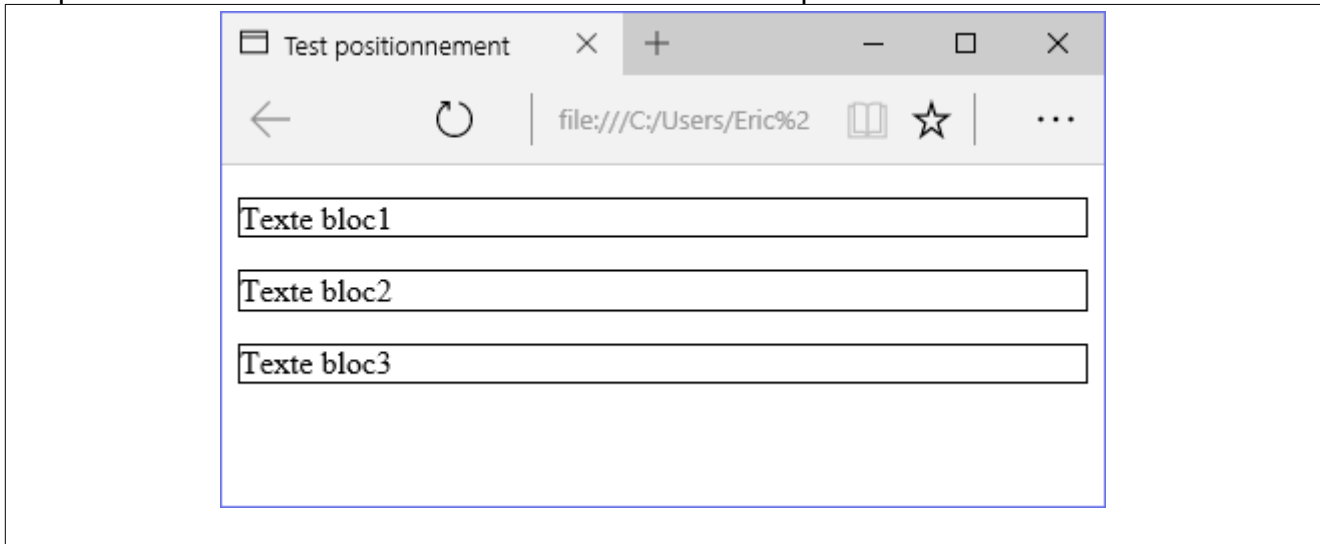
Afin de mieux visualiser les « blocs », on ajoute au fichier HTML les lignes suivantes :

```
<body>
  <p id="paragraphe1">Texte bloc1</p>
  <p id="paragraphe2">Texte bloc2</p>
  <p id="paragraphe3">Texte bloc3</p>
</body>
```

Et on complète le fichier CSS avec :

```
#paragraphe1
{
    border-style:solid;
    border-width: 1px;
}
#paragraphe2
{
    border-style:solid;
    border-width: 1px;
}
#paragraphe3
{
    border-style:solid;
    border-width: 1px;
}
```

On peut visualiser alors les blocs obtenus et leur occupation :



On s'aperçoit que les 3 blocs se retrouvent l'un en dessous de l'autre, dans l'ordre dans lequel ils apparaissent dans le fichier HTML, et qu'ils occupent la largeur totale de la place disponible.

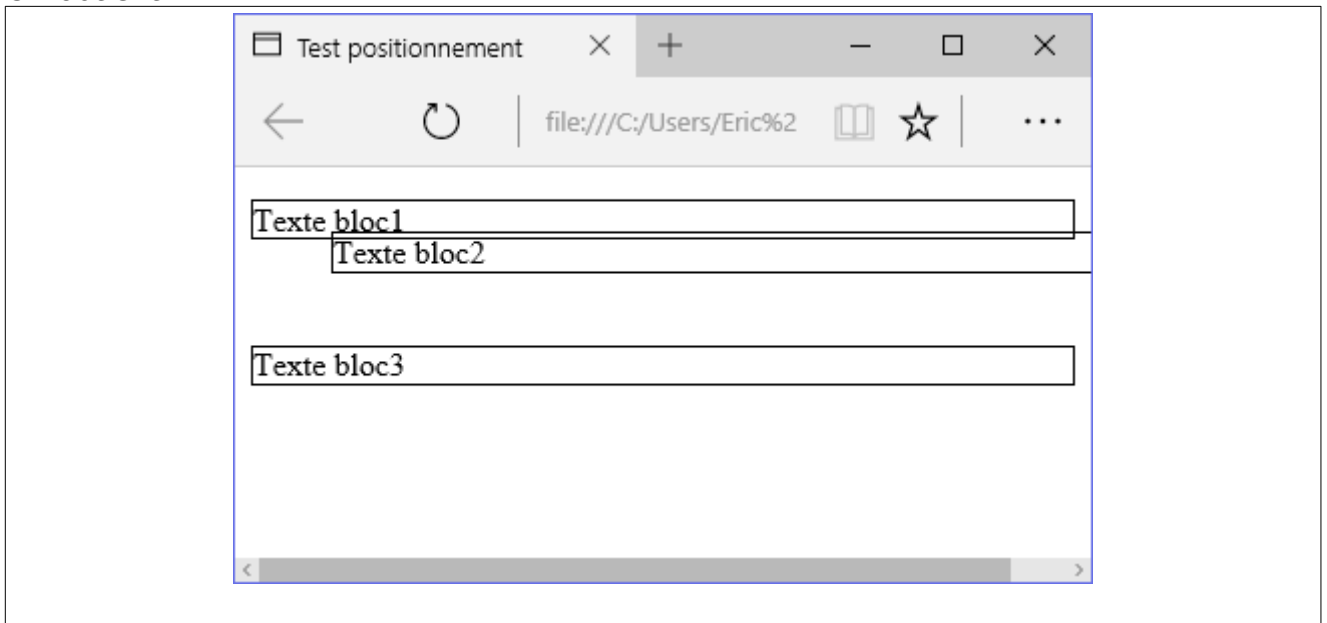
**Le positionnement « relatif » :**

Le conteneur déclaré « relatif » peut être décalé de la position qu'il aurait normalement dans le flux courant.

*Exemple :*

<pre>#paragraphe2 {     border-style:solid;     border-width: 1px;     position: relative ;     left: 40px ;     bottom: 20 px ; }</pre>	<p>Le conteneur appelé « paragraphe2 » sera décalé de 40 pixels vers la gauche (par rapport à la marge gauche du conteneur parent) et de 20 pixels vers le haut par rapport à la place qu'il aurait occupé si cette position relative n'avait pas été définie.</p>
--	--

On obtient :



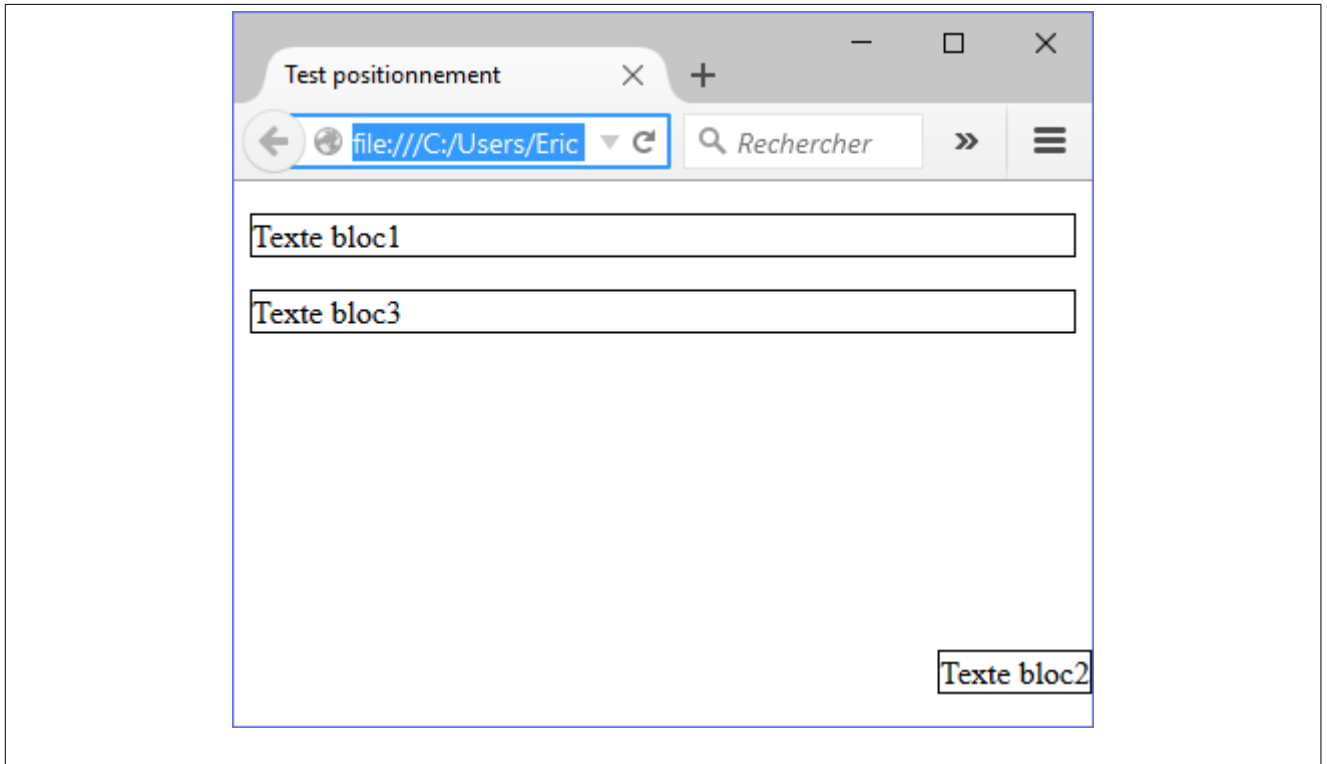
**Le positionnement « absolu » :**

Cette fois-ci, on va pouvoir réellement positionner un conteneur là où l'on veut sur la page, sachant que le point de coordonnées 0,0 est en haut à gauche de l'écran.

*Exemple :*

<pre>#paragraphe2 {     border-style:solid;     border-width: 1px;     position: absolute ;     right: 0px;     bottom: 0px; }</pre>	<p>Le conteneur appelé « paragraphe2» sera affiché en bas à droite de son conteneur parent.</p> <p><i>Ce qui ne veut pas forcément dire en bas à droite de l'écran ! Tout dépend du conteneur parent dans lequel il se trouve...</i></p>
--	--

On obtient :



**Le positionnement « flottant » :**

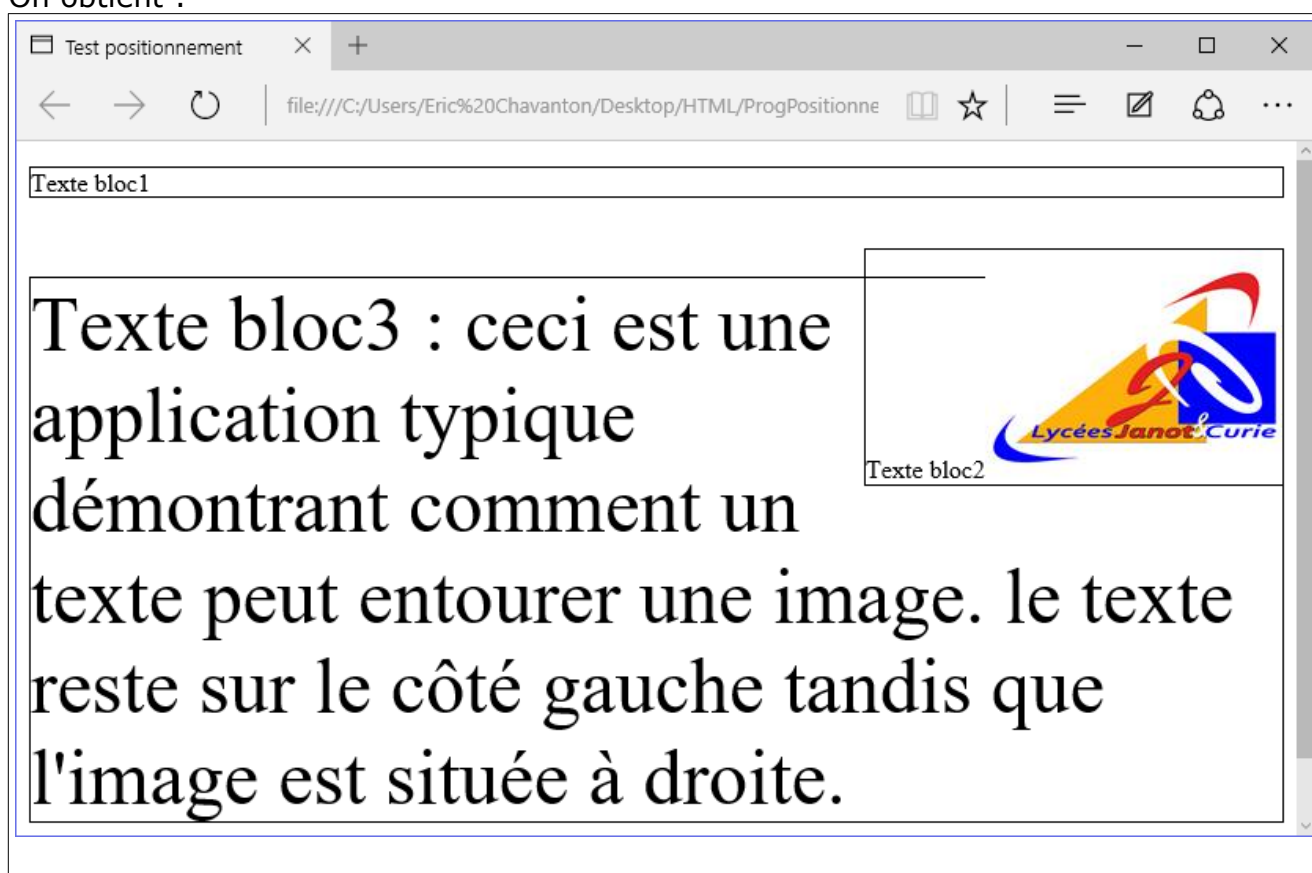
Le conteneur est « retiré » du flux normal pour prendre place à gauche (left) ou à droite (right) du conteneur qui le contient. L'élément qui le suit dans le code HTML occupera alors l'espace laissé libre, en épousant sa forme.

Pour bien démontrer l'aspect pratique de ce positionnement, nous allons prendre l'exemple d'une image autour de laquelle on souhaite mettre du texte,

**Exemple :**

<pre>#paragraphe2 {     border-style:solid;     border-width: 1px;     float: right; }</pre>	<p>Le conteneur appelé « paragraphe2 » sera affiché à droite de son conteneur parent.</p>
--	---

On obtient :

**En conclusion :**

On peut noter que cette partie de la conception en HTML/CSS, est la base de la mise en page d'un site. Il est absolument nécessaire de maîtriser ces différentes notions, et surtout de se rappeler que les règles appliquées se font EN FONCTION du conteneur parent !

Il est donc absolument indispensable, lors de la création d'une page d'un site, de réfléchir au préalable à l'agencement des différents blocs, qui constitueront la page.