

CSS

Table des matières

1. La petite histoire du CSS.....	2
2. Déclaration du CSS.....	2
2.1. Fichier .css.....	2
2.2. En-tête <head> du fichier HTML.....	3
2.3. Directement dans les balises.....	3
3. Appliquer un style.....	3
3.1. Commentaires dans du CSS.....	4
3.2. Style : class et id.....	4
3.3. Les balises universelles.....	5
3.4. Les sélecteurs avancés.....	5
4. Formattage du texte.....	6
5. Habillage.....	7
6. Couleur du texte.....	8
7. Couleur de fond.....	9
8. Images de fond.....	9
9. Les Bordures.....	11
10. Apparences dynamiques.....	11
11. Le modèle des boîtes.....	12
Dépassement de texte.....	14
12. Le positionnement.....	14
12.1. Les positionnements absolu, fixe et relatif.....	14
12.2. Le positionnement inline-block.....	15
13. Squelette de base HTML – CSS.....	18



1. La petite histoire du CSS

CSS (**Cascading Style Sheets**), permet de choisir la couleur du texte, la police utilisée, la taille du texte, les bordures, le fond... et de faire la mise en page du site (menu à gauche, en-tête calé en haut, etc).

Aux débuts du Web, CSS n'existait pas, il n'y avait initialement que le langage HTML. Cependant, les pages HTML commençaient à devenir assez complexes. Il y avait de plus en plus de balises et c'était un joyeux mélange entre le fond et la forme, qui rendait la mise à jour des pages web de plus en plus complexe. C'est pour cela que l'on a créé le langage CSS.

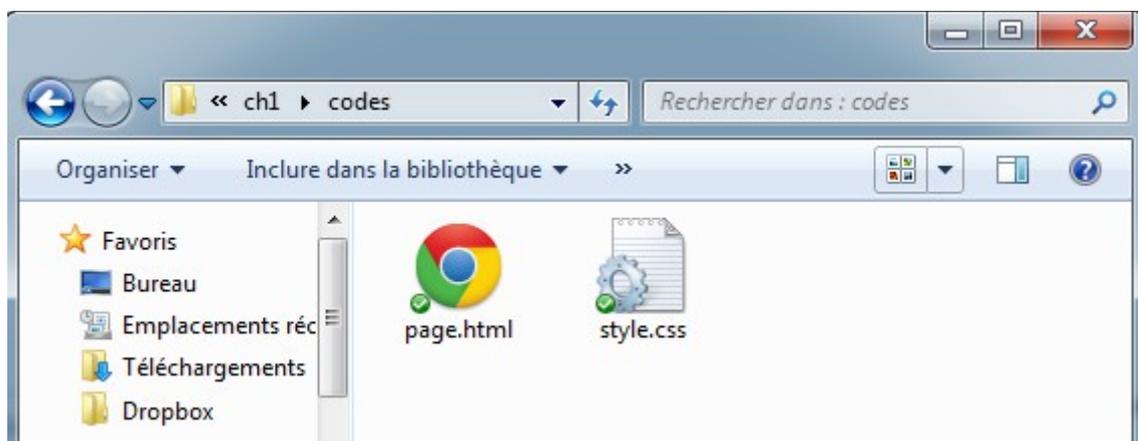
2. Déclaration du CSS

On peut écrire du code en langage CSS à trois endroits différents :

- dans un fichier `.css` (*méthode la plus recommandée*) ;
- dans l'en-tête `<head>` du fichier HTML ;
- directement dans les balises du fichier HTML *via* un attribut `style` (*méthode la moins recommandée*).

2.1. Fichier `.css`

On écrit le plus souvent le code CSS dans un fichier spécial ayant l'extension `.css`. Cela nous évite de tout mélanger dans un même fichier.



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="style.css" />
    <title>Premiers tests du CSS</title>
  </head>
  <body>
    <h1>Mon site</h1>
    <p>Bonjour et bienvenue !</p>
  </body>
```

```
</html>
```

La ligne 5, `<link rel="stylesheet" href="style.css" />` indique que ce fichier HTML est associé à un fichier appelé `style.css` et chargé de la mise en forme.

Tout le **site** changera d'apparence.

2.2. En-tête `<head>` du fichier HTML

Cela consiste à insérer le code CSS directement dans une balise `<style>` à l'intérieur de l'en-tête `<head>`.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <style>
      p
      {
        color: blue;
      }
    </style>
    <title>Premiers tests du CSS</title>
  </head>
  <body>
    <h1>Mon site</h1>
    <p>Bonjour et bienvenue !</p>
  </body>
</html>
```

Toute la **page** changera d'apparence.

2.3. Directement dans les balises

Ajouter un attribut `style` à n'importe quelle balise et insérez le code CSS directement dans cet attribut :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Premiers tests du CSS</title>
  </head>
  <body>
    <h1>Mon site</h1>
    <p style="color: blue;">Bonjour et bienvenue !</p>
  </body>
</html>
```

3. Appliquer un style

Dans un code CSS, on trouve trois éléments différents :

- **Des noms de balises** : on écrit les noms des balises dont on veut modifier l'apparence.

- **Des propriétés CSS** : les « effets de style » de la page sont rangés dans des propriétés. Il y a par exemple la propriété `color` qui permet d'indiquer la couleur du texte.
- **Les valeurs** : pour chaque propriété CSS, on doit indiquer une valeur. Par exemple, pour la propriété `color`, il faut indiquer le nom de la couleur.

Prenons le code CSS suivant :

```
h1
{
  color: blue;
}
em
{
  color: blue;
}
```

Il signifie que les titres `<h1>` et les textes importants `` doivent s'afficher en bleu.

Si les deux balises doivent avoir la même présentation, il suffit de combiner la déclaration en séparant les noms des balises par une virgule :

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

3.1. Commentaires dans du CSS

Les commentaires ne seront pas affichés, ils servent simplement à indiquer des informations.

Taper `/*`, suivi de votre commentaire, puis `*/` pour terminer votre commentaire.

3.2. Style : class et id

Pour que certains paragraphes seulement soient écrits d'une manière différente on utilise des attributs spéciaux *qui fonctionnent sur toutes les balises* :

- l'attribut `class` ;
- l'attribut `id`.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="style.css" />
    <title>Premiers tests du CSS</title>
  </head>
  <body>
    <h1>Mon super site</h1>
    <p class="introduction">Bonjour et bienvenue !</p>
  </body>
</html>
```

Dans le CSS, indiquez le nom de votre classe en commençant par un point, comme ci-dessous :

```
.introduction
{
  color: blue;
}
```

L'attribut **id** fonctionne exactement de la même manière que **class** mais il ne peut être utilisé *qu'une fois* dans le code.

Dans le fichier CSS, il faudra faire précéder le nom de l'id par un dièse (#) :

```
#introduction
{
  color: blue;
}
```

3.3. Les balises universelles

Il existe deux balises dites **universelles** :

- ` ` : c'est une balise de type **inline**, c'est-à-dire une balise que l'on place au sein d'un paragraphe de texte, pour sélectionner certains mots uniquement. Les balises `` et `` sont de la même famille.
- `<div> </div>` : c'est une balise de type **block**, qui entoure un bloc de texte. Les balises `<p>`, `<h1>`, etc. sont de la même famille. Ces balises ont quelque chose en commun : elles créent un nouveau « bloc » dans la page et provoquent donc obligatoirement un retour à la ligne.

Exemple :

```
<p>Bonjour et <span class="salutations">bienvenue</span> !</p>
```

```
.salutations
{
  color: blue;
}
```

3.4. Les sélecteurs avancés

Pour sélectionner les éléments de la page à modifier, on utilise ce qu'on appelle des **sélecteurs**.

*** : sélecteur universel**

```
*
{
}
```

Sélectionne toutes les balises sans exception. On l'appelle le sélecteur universel.

AB : une balise contenue dans une autre

```
h3 em
{
}
```

Sélectionne toutes les balises `` situées à l'intérieur d'une balise `<h3>`. Notez qu'il n'y a pas de

virgule entre les deux noms de balises.

A + B : une balise qui en suit une autre

```
h3 + p
{
}
```

Sélectionne la première balise <p> située après un titre <h3>.

A[attribut] : une balise qui possède un attribut

```
a[title]
{
}
```

Sélectionne tous les liens <a> qui possèdent un attribut title.

Exemple :

A[attribut="Valeur"] : une balise, un attribut et une valeur exacte

```
a[title="Cliquez ici"]
{
}
```

Idem, mais l'attribut doit en plus avoir exactement pour valeur « Cliquez ici ».

Exemple :

A[attribut*="Valeur"] : une balise, un attribut et une valeur

```
a[title*="ici"]
{
}
```

Idem, l'attribut doit cette fois contenir dans sa valeur le mot « ici » (peu importe sa position).

Exemple :

4. Formattage du texte

Pour modifier la **taille** du texte, on utilise la propriété CSS font-size.

Exemple :

```
p
{
  font-size: 14px; /* Paragraphes de 14 pixels */
}
```

Pour modifier la **police** utilisée, on utilise la propriété CSS font-family.

```
p
{
  font-family: Impact, "Arial Black", Arial, Verdana, sans-serif;
}
```

Le navigateur essaiera d'abord d'utiliser la police Impact ou, si elle n'y est pas, Arial Black, ou sinon Arial, ou sinon Verdana, ou si rien n'a marché, mets une police standard (sans-serif).

Mettre en **italique** : font-style: italic;

Mettre en **gras** : font-weight: bold;

Soulignement et **autres décorations** : propriété CSS text-decoration qui peut prendre différentes valeurs :

- underline : souligné.
- line-through : barré.
- overline : ligne au-dessus.
- blink : clignotant.
- none : normal (par défaut).

L'alignement : propriété CSS text-align et on indique l'alignement désiré :

- 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 ».

5. Habillage



Lorem ipsum dolor sit
consectetur adipiscing elit.
vitae lorem imperdiet lacus
molestie. Cum sociis natoque p
et magnis dis parturient montes, nascetur
mus. Donec eu purus. Phasellus metus

Le CSS permet de faire flotter un élément autour d'un texte grâce à la propriété CSS **float**.

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

```
<p>
```

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



Ce texte entoure l'image.
bla bla bla bla bla bla bla

Pour ne plus faire flotter l'élément, il faut utiliser la propriété **clear**, qui peut prendre ces trois valeurs :

- 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

Texte écrit en-dessous

après un float: left; ou après un float: right;.

```
<p></p>
```

```
<p>Texte écrit à côté de l'image.</p>
```

```
<p class="dessous">Texte écrit sous l'image.</p>
```

```
.imageflottante
```

```
{  
  float: left;  
}
```

```
.dessous
```

```
{  
  clear: both;  
}
```

6. Couleur du texte

Il y a différentes façons d'indiquer la couleur :

1. Indiquer le nom de la couleur

Propriété **color**, suivie d'un des seize noms de couleurs utilisables en CSS (white, black, red, yellow, blue, ...).

Exemple : color: maroon

2. Indiquer le code couleur

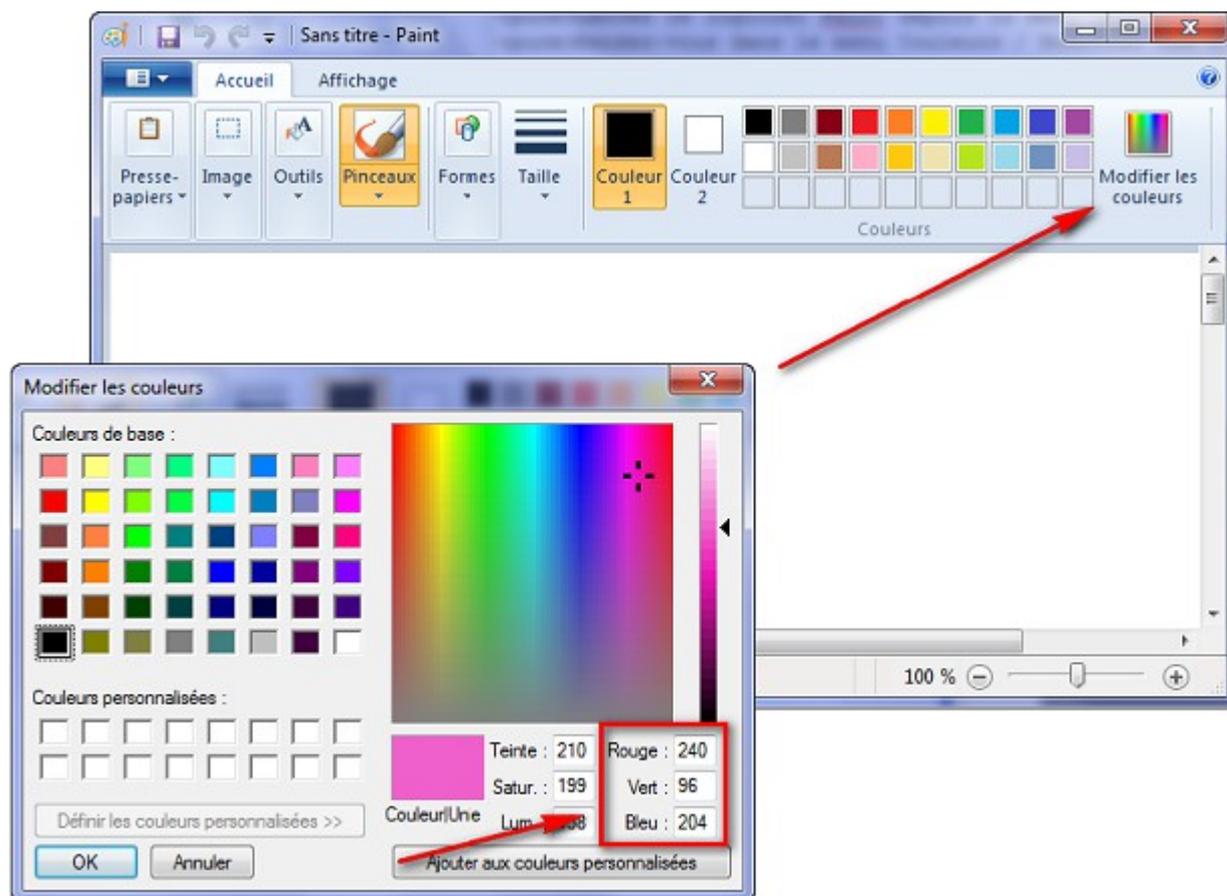
Propriété **color**, suivie des codes couleur **RGB**¹.

Exemple : color: rgba(240, 96, 204, 0);

Certains logiciels de retouche d'image permettent de trouver la couleur désirée (ex : paint).

Remarque : le dernier paramètre indique la transparence et a une valeur comprise entre 0 et 1 (ainsi, pour une valeur de 0.6, le texte sera opaque à 60%).

1 Red Green Blue (Rouge Vert Bleu)



7. Couleur de fond

Pour indiquer une couleur de fond, utiliser la propriété CSS `background-color` et elle s'utilise de la même manière que la propriété `color`. Pour indiquer la couleur de fond de la page web, il faut travailler sur la balise `<body>`.

Exemple :

```
body
{
  background-color: black; /* Le fond de la page sera noir */
  color: white; /* Le texte de la page sera blanc */
}
```

8. Images de fond

La propriété permettant d'indiquer une image de fond est `background-image`.

Exemple : `background-image: url("neige.png");`

La propriété CSS `background-attachment` permet de « fixer » le fond. Deux valeurs sont disponibles :

- `fixed` : l'image de fond reste fixe ;

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

La propriété CSS **background-repeat** permet de « répéter » le fond. Quatre valeurs sont disponibles :

- no-repeat : le fond ne sera pas répété. L'image sera donc unique sur la page.
- repeat-x : le fond sera répété uniquement sur la première ligne, horizontalement.
- repeat-y : le fond sera répété uniquement sur la première colonne, verticalement.
- repeat : le fond sera répété en mosaïque (par défaut).

La propriété CSS **background-position** permet d'indiquer où doit se trouver l'image de fond . Deux valeurs sont disponibles :

- en pixels pour indiquer la position du fond par rapport au coin supérieur gauche de la page
- utiliser les valeurs : top, bottom, left, center, right

Exemple :

```
body
{
  background-image: url("soleil.png");
  background-attachment: fixed; /* Le fond restera fixe */
  background-repeat: no-repeat; /* Le fond ne sera pas répété */
  background-position: top right; /* Le fond sera placé en haut à droite */
}
```

Depuis CSS3, il est possible de donner plusieurs images de fond à un élément. Pour cela, il suffit de séparer les déclarations par une virgule, comme ceci :

```
body
{
  background: url("soleil.png") fixed no-repeat top right, url("neige.png") fixed;
}
```

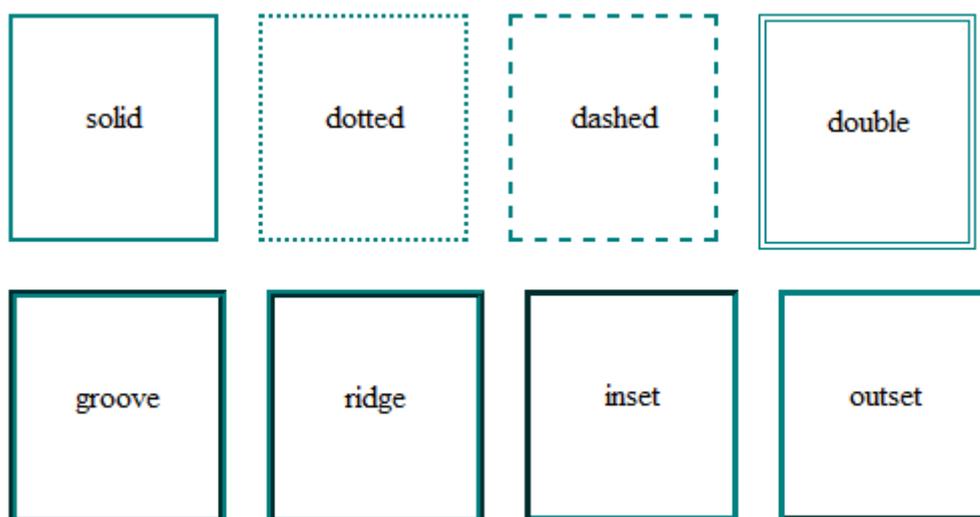


La première image de cette liste sera placée par-dessus les autres. Attention donc, l'ordre de déclaration des images a son importance : si vous inversez le soleil et la neige dans le code CSS précédent, vous ne verrez plus le soleil !

9. Les Bordures

Pour modifier l'apparence des bordures il faut utiliser la propriété **border** qui combine plusieurs valeurs.

1. la largeur en pixels
2. la couleur (cf §couleur du texte)
3. le type de bordure avec différentes valeurs disponibles :
 - none : pas de bordure (par défaut) ;
 - solid : un trait simple ;
 - dotted : pointillés ;
 - dashed : tirets ;
 - etc...



Exemple : `border: 3px blue dashed;`

Pour ne modifier qu'**une bordure**, utiliser les propriétés :

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

Pour obtenir des **bordures arrondies**, utiliser la propriété `border-radius` en indiquant la taille de l'arrondi en pixels.

Exemple : `border-radius: 10px;`

10. Apparences dynamiques

En CSS, on peut modifier l'apparence de certaines sections dynamiquement, après le chargement de la page, lorsque certains évènements se produisent. On utilise pour cela les pseudo-formats.

- **:hover** permet de changer l'apparence au survol (par exemple : a:hover pour modifier l'apparence des liens lorsque la souris pointe dessus).
- **:active** applique un style particulier au moment du clic. En pratique, il n'est utilisé que sur les liens.
- **:focus** applique un style lorsque l'élément est sélectionné.
- **:visited** applique un style à un lien vers une page qui a déjà été vue.

Exemple :

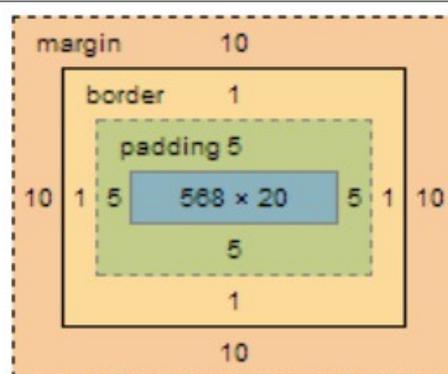
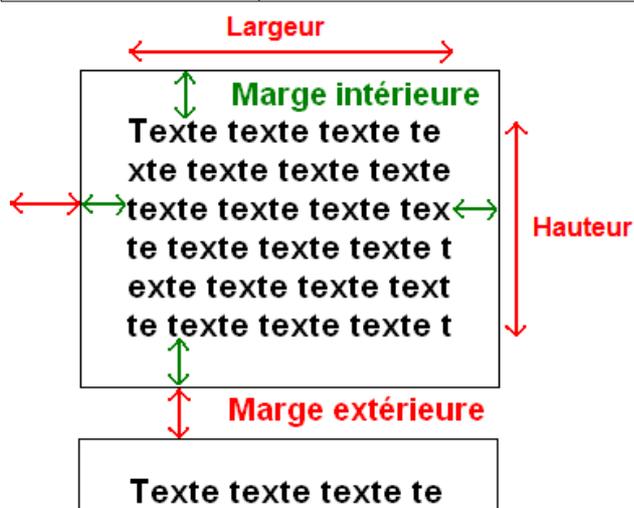
```
p:hover /* Apparence au survol des paragraphes */
{
  text-decoration: underline;
  color: green;
}
```

11. Le modèle des boîtes

Dans une mise en page réalisée en CSS, tous les éléments sont considérés comme des boîtes. Chacune de ces boîtes est constituée d'un contenu, d'un espacement intérieur, d'une bordure, et d'une marge externe.

Voici les propriétés CSS qui permettent de déterminer les dimensions, la couleur, le style de chacun de ses constituants :

Propriété CSS	Ce qui est concerné :
width et height	largeur et hauteur du contenu (texte, image, etc.)
padding	espacement intérieur, entre le contenu et la bordure
border	bordure (ou encadrement)
margin	marge externe, espace (transparent) entourant le tout



```
#header {
  border: black solid 1px;
  padding: 5px;
  margin: 10px;
}
```

- width : c'est la largeur du bloc exprimé en pixels (px) ou en pourcentage (%).
- height : c'est la hauteur du bloc exprimé en pixels (px) ou en pourcentage (%).
- padding : indique la taille de la marge intérieure en pixels (px).
- margin : indique la taille de la marge extérieure en pixels (px).

1. Un bloc peut avoir des dimensions minimales et maximales :

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

2. Les marges extérieures peuvent avoir des valeurs différentes :

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

3. Idem pour les marges intérieures !

Exemple :

```
p
{
width: 350px;
border: 1px solid black;
text-align: justify;
padding: 12px;
margin: 50px; /* Marge extérieure de 50px */
}
```

Remarque : utiliser la propriété **margin: auto** pour centrer des blocs. Pour cela, il faut obligatoirement donner une largeur au bloc (avec la propriété width).

Exemple :

```
p
{
width: 350px; /* On a indiqué une largeur (obligatoire) */
margin: auto; /* On peut donc demander à ce que le bloc soit centré avec auto */
}
```

Dépassement de texte

Si vous voulez que le texte ne dépasse pas des limites du bloc, il va falloir utiliser la propriété `overflow`. Voici les valeurs qu'elle peut accepter :

- `visible` (par défaut) : si le texte dépasse les limites de taille, il reste visible et sort volontairement du bloc.
- `hidden` : si le texte dépasse les limites, il sera tout simplement coupé. On ne pourra pas voir tout le texte.
- `scroll` : le texte sera coupé s'il dépasse les limites. Sauf que cette fois, le navigateur mettra en place des barres de défilement pour qu'on puisse lire l'ensemble du texte.
- `auto` : le navigateur décide de mettre ou non des barres de défilement.

Remarque : la propriété `word-wrap: break-word` permet de forcer la césure des très longs mots (généralement des adresses un peu longues).

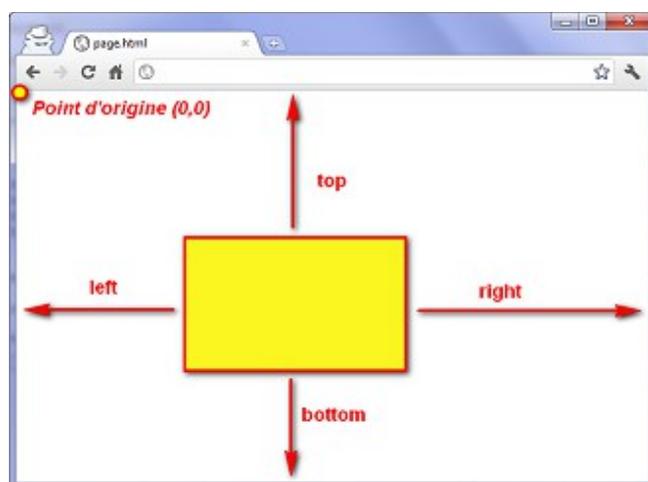
12. Le positionnement

12.1. Les positionnements absolu, fixe et relatif

La propriété CSS `position` permet de positionner avec précision des éléments sur la page. Pour cela, on lui donne une de ces valeurs :

- `absolute` : positionnement absolu ; il permet de placer un élément n'importe où sur la page (en haut à gauche, en bas à droite, tout au centre, etc.).
- `fixed` : positionnement fixe ; identique au positionnement absolu mais, cette fois, l'élément reste toujours visible, même si on descend plus bas dans la page.
- `relative` : positionnement relatif ; ce positionnement permet d'effectuer des « ajustements » : l'élément est décalé par rapport à sa position initiale.

Si un bloc est positionné en absolu, il faut indiquer au navigateur où le positionner sur la page à l'aide des quatre propriétés CSS :



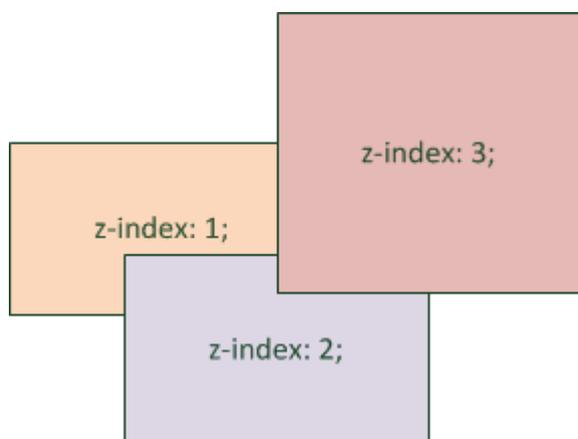
- `left` : position par rapport à la gauche de la page ;
- `right` : position par rapport à la droite de la page ;
- `top` : position par rapport au haut de la page ;
- `bottom` : position par rapport au bas de la page.

Exemple :

```

element
{
  position: absolute;
  right: 0px;
  bottom: 0px;
}
    
```

Remarque : les éléments positionnés en absolu sont placés par-dessus le reste des éléments de la page ! 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 le montre la figure ci-contre.

12.2. Le positionnement inline-block

En CSS la propriété **display** permet de transformer n'importe quel élément de la page d'un type vers un autre et les faire apparaître sous forme de blocs. À ce moment-là, les éléments vont se positionner les uns en-dessous des autres et il devient possible de modifier leurs dimensions !

Voici quelques-unes des principales valeurs que peut prendre la propriété display :

Valeur	Exemples	Description
inline	<a>, , ...	Éléments d'une ligne. Se placent les uns à côté des autres.
block	<p>, <div>, <section>...	Éléments en forme de blocs. Se placent les uns en-dessous des autres et peuvent être redimensionnés.
inline-block	<select>, <input>	Éléments positionnés les uns à côté des autres (comme les inlines) mais qui peuvent être redimensionnés (comme les blocs).
none	<head>	Éléments non affichés.

Les éléments en inline-block nous permet d'utiliser la propriété vertical-align. Cette propriété permet de modifier l'alignement vertical des éléments. 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) ;

- top : aligne en haut ;
- middle : centre verticalement ;
- bottom : aligne en bas ;
- (valeur en px ou %) : aligne à une certaine distance de la ligne de base (baseline).

NB : les éléments inline-block se positionnent sur une même ligne de base (appelée baseline), en bas.

Exemple : nous voulons réaliser la page suivante.



Donnons le code HTML correspondant...

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="style.css" />
    <title>Zozor - Le Site Web</title>
  </head>
  <body>
    <header>
```

```

    <h1>Zozor</h1>
    <h2>Carnets de voyage</h2>
</header>

<nav>
  <ul>
    <li><a href="accueil.html">Accueil</a></li>
    <li><a href="blog.html">Blog</a></li>
    <li><a href="cv.html">CV</a></li>
  </ul>
</nav>

<section>
  <aside>
    <h1>À propos de l'auteur</h1>
    <p>C'est moi, Zozor ! Je suis né un 23 novembre 2005.</p>
  </aside>
  <article>
    <h1>Je suis un grand voyageur</h1>
    <p>Bla bla bla bla (texte de l'article)</p>
  </article>
</section>

<footer>
  <p>Copyright Zozor - Tous droits réservés
  <a href="mailto:zorro@monsite.org">Me contacter !</a></p>
</footer>
</body>
</html>

```

...et le code CSS associé.

```

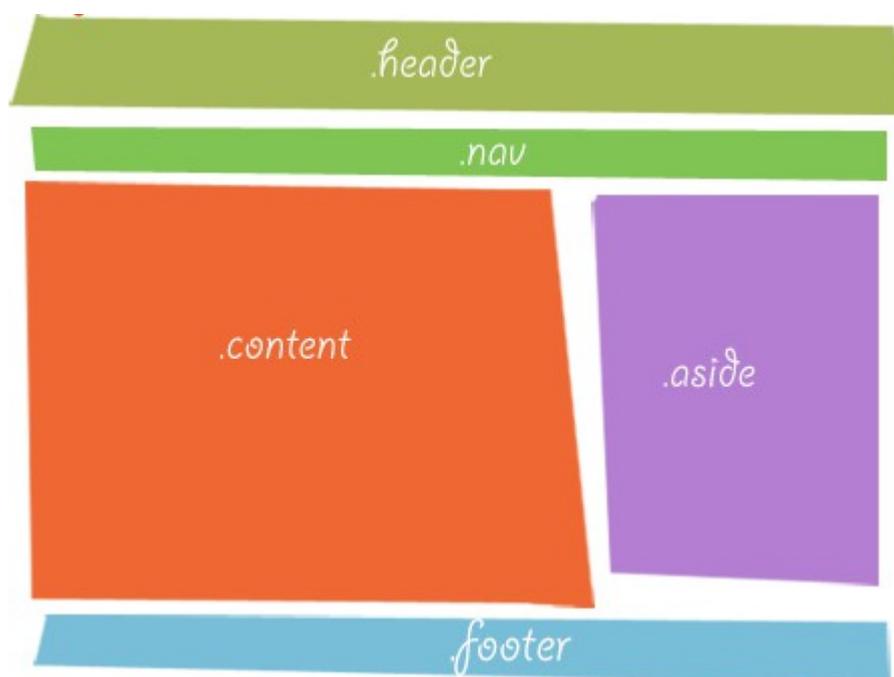
nav
{
  display: inline-block;
  width: 150px;
  border: 1px solid black;
  vertical-align: top;
}

section
{
  display: inline-block;
  border: 1px solid blue;
  vertical-align: top;
}

```

13. Squelette de base HTML – CSS

Le squelette d'une page web possède souvent une structure de base à cinq blocs principaux.



Cette structure de base à cinq blocs principaux convient dans la majorité des cas, car elle permet de fabriquer une grande variété de mises en page.

HTML	CSS
<code><!DOCTYPE html></code>	<code>/* Mes styles */</code>
<code><html></code>	<code>.header {....}</code>
<code><head></code>	
<code><title>Titre de la page</title></code>	<code>.nav {....}</code>
<code><link rel="stylesheet" href="style.css"/></code>	
<code></head></code>	<code>.content {....}</code>
	<code>.aside {....}</code>
<code><body></code>	<code>.footer {....}</code>
<code><div class="header">En-tête</div></code>	
<code><div class="nav">Navigation</div></code>	
<code><div class="content">Contenu</div></code>	
<code><div class="aside">Contexte</div></code>	
<code><div class="footer">Pied de page</div></code>	
<code></body></code>	
<code></html></code>	

Il existe sur le web des collections de modèles de mise en page, à télécharger gratuitement.

Exemple : « [Layout Gala](#) »