

**HTML5 - CCS3**

**- Hyper Text Markup Language et Cascading Style Sheet -**

# 01 - Les bases pour créer un site

Le langage HTML a été inventé par Tim Berners-Lee en 1991. Tim Berners-Lee suit encore aujourd'hui avec attention l'évolution du Web.

Il a créé le World Wide Web Consortium (W3C) qui définit les nouvelles versions des langages liés au Web. Il a par ailleurs créé plus récemment la World Wide Web Foundation qui analyse et suit l'évolution du Web.

Il s'agit de deux langages, à la base, HTML et CSS, qui sont traduits par les navigateurs.

Le HTML (HyperText Markup Language) sert à gérer et organiser le contenu de la page.

Le CSS (Cascading Style Sheets, aussi appelées Feuilles de style), créé en 1996, sert à gérer l'apparence de la page. Le CSS3 (dernière version) a besoin, pour fonctionner, d'une page en HTML4 (1998) ou HTML5 (dernière version).

Il existe deux outils pour créer des sites web :

- Les WYSIWYG (What You See Is What You Get - Ce Que Vous Voyez Est Ce Que Vous Obtenez).

Ce sont des programmes qui se veulent très faciles d'emploi, ils permettent de créer des sites web sans apprendre de langage particulier.

Parmi les plus connus d'entre eux : Mozilla Composer, Dreamweaver, Kompozer. Leur défaut consiste dans la mauvaise qualité du code qui est automatiquement généré.

- Les éditeurs de texte.

Ce sont des programmes dédiés à l'écriture de code.

Par exemple Notepad++ :

Après l'installation de Notepad++, deux paramétrages sont importants :

- On utilise l'option [Langage] > [H] > [HTML], cela permet de colorer le code pour mieux s'y retrouver.

- On choisit aussi l'encodage, dans [Paramétrage] > [Préférences] > onglet [Nouveau document/ Dossier] > liste [Encodage] > [UTF-8 sans BOM]

Pour tester les sites web, on passe par un navigateur.

Chaque navigateur (Internet Explorer, Mozilla Firefox, Google Chrome, Safari, Opera, ...) ne fonctionne pas tout à fait de la même manière : il peut être bon de tester un code sur plusieurs navigateurs, pour éviter les surprises.

Il peut aussi y avoir des différences selon les versions successives de chaque navigateur, dans la traduction des langages du Web.

Il y a aussi les navigateurs pour mobile, ainsi Safari Mobile sur iPhone (version light de Safari), Chrome Mobile sur Android, Internet Explorer Mobile sur Windows Phone, un navigateur spécifique proche de Safari et Chrome sur Blackberry, .... Sur ce support, les mises à jour sont automatiques, ce qui enlève les soucis d'affichage connus sur les ordinateurs.

Les langages HTML et CSS permettent de créer des sites statiques.

Le PHP permet de créer des sites dynamiques.

Il faut maîtriser HTML et CSS pour se lancer dans le PHP.

Il vous faudra réserver votre nom de domaine (URL), définir une adresse à laquelle trouver votre site.

Enfin, le site créé, il suffira de le transférer sur le serveur sur lequel il sera hébergé et ainsi le rendre publique. On utilise alors le protocole FTP (File Transfer Protocol) à l'aide d'un outil comme Filezilla pour transférer tous les fichiers relatifs au site.

## En savoir plus

<http://fr.wikipedia.org/wiki/Internet>

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

<http://www.w3schools.com/w3c/>

<http://www.w3schools.com/>

## Les outils

<http://www.frenchmozilla.fr/>

<http://notepad-plus-plus.org/fr/>

## En ligne HTML

<http://www.htmlinstant.com/>

<http://htmledit.squarefree.com/>

## En ligne CSS

<http://cssdesk.com/>

<http://aurelio.net/projects/css-sandbox/>

<http://css3generator.com/>

<http://gradients.glrzad.com/>

<http://www.spritebox.net/>

## En ligne IMAGE

<http://pixlr.com/editor/>

++

<http://www.siteduzero.com/>

<http://www.framasoft.net/article1456.html>

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

## 02 - Les premiers pas de la création

Une bonne chose, avant de se lancer sur le code, c'est de s'organiser.

Créer un dossier dans lequel vous y rangerez vos sources (texte, images, etc.), un autre dans lequel vous placerez les éléments sites (pages html, feuilles de styles, images, etc.)

Pour commencer, on ouvre l'éditeur de texte Notepad++.

On peut écrire ce qu'on veut, deux lignes pour commencer :

Bonjour à tous !

Voici mon premier site web.

Ensuite, on enregistre la page au format html dans le dossier créé précédemment.

Pour le nom de fichier, on choisit ce qu'on veut, mais avec l'extension .html (par exemple index.html).

Dans l'explorateur, le fichier prend l'icône du navigateur par défaut. Si on ouvre cette page, on observe que le texte est écrit simplement sur une ligne en haut à gauche, alors qu'e nous occupons deux lignes de code.

### 2.1. Les balises et leurs attributs

Le langage HTML fonctionne avec des balises. Elles sont invisibles pour les visiteurs du site, mais elles permettent au navigateur d'opérer des instructions pour l'affichage du contenu.

Les balises sont entourées de chevrons, les signes < et >.

On utilise l'anglais et des abréviations dans les balises.

On distingue deux types de balises :

- les balises en paires (qui s'ouvrent, contiennent du texte, et se ferment : <title>Ceci est un titre</title>).
- les balises orphelines (qui ne nécessitent de début et de fin, par exemple pour une image : <image />).

Les attributs sont des options, des instructions associées aux balises.

Par exemple on précise le nom de l'image à insérer avec l'attribut src (qui signifie source) :



<!-- ici on va charger l'image photo.jpg qui se situe dans le sous-dossier image/ -->

### 2.2. La base d'une page web

Voici la base d'une page web en HTML5 :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset = "utf-8" />
    <title>Titre</title>
  </head>
  <body>
  </body>
</html>
```

Dans l'en-tête :

- On choisit plutôt l'encodage utf-8, il permet l'affichage de tous les symboles de toutes les langues.
- On donne un titre à la page, entre les balises <title></title> : le titre apparaît dans l'onglet du navigateur et dans les résultats du moteur de recherche (quand le site est référencé).

### 2.3. Indentations et commentaires

Indentations et commentaires servent au webmaster pour mieux lire le code qu'il écrit dans son éditeur de texte.

Les indentations sont des tabulations qui permettent de hiérarchiser les lignes de code. Cela permet de repérer les balises qui se trouvent encadrées par d'autres balises.

Les commentaires sont des textes qui ne sont pas traduits par le navigateur, mais qui donnent des informations au webmaster sur le code qu'il a écrit (pour s'y retrouver plus facilement par la suite, comme une page web html peut faire plusieurs centaines de lignes).

On écrit le commentaire entre <!-- et -->

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Ceci est un commentaire -->
    <meta charset = "utf-8" />
    <title>Titre</title>
  </head>
  <body>
    <!-- Ici c'est le corps de la page, je le mets en commentaire -->
  </body>
</html>
```

## 03 - Organiser le texte

Les éléments sont intégrés dans la balise <body>.

### Il est bon de faire des tests pour chaque nouveauté.

Nous allons voir ici comment rédiger des paragraphes, comment structurer sa page avec les titres, comment donner de l'importance à certains mots, comment organiser les informations sous forme de listes.

#### 3.1. Les paragraphes

Pour créer un paragraphe, on utilise la balise <p> :

```
<p>Ceci est mon premier paragraphe.</p>
```

```
<p>Ceci est mon deuxième paragraphe.</p>
```

On peut aussi faire de simples sauts de ligne à l'intérieur d'un paragraphe avec la balise orpheline <br /> (il n'y aura pas d'espace entre les deux lignes, contrairement à <p>).

Par exemple :

```
<!DOCTYPE html>
<html>
<head>
    <!-- Ceci est un commentaire -->
    <meta charset ="utf-8" />
    <title >Titre </title>
</head>
<body>
    <p>Ceci est mon premier paragraphe.
    <br />J'y mets une partie à la ligne</p>
    <p>Ceci est mon deuxième paragraphe.</p>
</body>
</html>
```

#### 3.2. Les titres

Pour les titres, on peut utiliser 6 niveaux différents, du niveau le plus important avec <h1> au niveau le moins important avec <h6> :

```
<h1>Titre de la page</h1>
<h2>Titre secondaire</h2>
<h3>Sous-titre</h3>
```

Par exemple :

```
<body>
<h1>Titre de la page</h1>
<h2>Titre secondaire 1</h2>
    <h3>Sous-titre 1.1</h3>
        <p>Ceci est mon premier paragraphe.
        <br />J'y mets une partie à la ligne</p>
    <h3>Sous-titre 1.2</h3>
        <p>Ceci est mon <em>deuxième</em> paragraphe.</p>
<h2>Titre secondaire 2</h2>
    <h3>Sous-titre 2.1</h3>
```

```
<p>Ceci est mon <strong>troisième</strong> paragraphe.</p>
<h3>Sous-titre 2.2</h3>
<p>Ceci est mon <mark>quatrième paragraphe</mark>.</p>
</body>
```

On voit déjà une mise en forme des caractères (automatique). Elle sera personnalisée en CSS.

Dans le texte de l'exemple, on a choisi de mettre en valeur certains mots, en emphase avec <em> ou en importance avec <strong>. Par défaut <em> fait apparaître le texte en italique, <strong> le fait apparaître en gras ; cela peut être changé ensuite avec le CSS. On peut aussi marquer du texte avec <mark> : il apparaît surligné en jaune par défaut.

**Attention :** <em> ne signifie pas italique, mais emphase : l'apparence de l'emphase peut être modifiée avec la feuille de style en CSS, comme pour <strong> et <mark>.

#### 3.3. Les listes

On crée une liste non ordonnée (unordered list) avec les balises <ul>, entre lesquelles on positionne chaque élément de la liste avec <li>. Chaque élément apparaît par défaut avec une puce devant (ce qui est aussi paramétrable en CSS).

Par exemple :

```
<ul>
    <li>Mathieu</li>
    <li>Ahmed</li>
    <li>François</li>
</ul>
```

On peut aussi imbriquer des listes dans des listes :

```
<ul>
    <li>Mathieu</li>
        <ul>      <li>Gateaux</li>    <li>Boisson</li>    </ul>
    <li>Ahmed</li>
        <ul>      <li>Bougies</li>    <li>Cadeau</li>    </ul>
    <li>François</li>
</ul>
```

## 04 - Créer des liens

### 4.1. Les liens absous (adresse URL complète)

Les liens absous permettent d'accéder à d'autres sites que le nôtre.

On crée un lien avec la balise `<a>`, dans laquelle on donne l'adresse URL avec l'attribut `href`. On indique aussi le texte sur lequel cliquer pour accéder à la page :

```
<a href="http://www.clg-rimbaud-aubergenville.ac-versailles.fr">Site web du collège</a>
```

Sur la page, par défaut, le lien est souligné et bleu (violet quand il a été visité).

Cela pourra être paramétré sur la feuille de style CSS.

On peut mettre ce lien dans un paragraphe :

```
<p>N'hésitez pas à visiter le <a href="http://www.clg-rimbaud-aubergenville.ac-versailles.fr">site  
web du collège</a>.</p>
```

Si jamais il y a des & dans une adresse URL, il faut les remplacer par &amp;:

### 4.2. Les liens relatifs (adresse URL incomplète)

Les liens relatifs permettent d'accéder à d'autres pages de notre site.

Par exemple on crée une deuxième page, dans le même dossier, qu'on appelle `page2.html`

Depuis l'`index`, on accède alors à la page 2 avec un lien relatif, le nom du fichier en attribut pour `href` :

```
<p>Si vous voulez accéder à la page 2, <a href="page2.html">cliquez ici</a>.</p>
```

On fait le lien inverse dans la page 2, pour naviguer de l'une à l'autre :

```
<p>Ceci est la 2e page du site, après l'index.<br />Vous pouvez revenir à l'index : <a href="index.  
html">cliquez ici</a>.</p>
```

On peut aussi avoir des sous-dossiers avec des pages (à éviter si possible, mais cela peut être pratique si on a beaucoup de pages). Prenons l'exemple d'une page 3 qui se trouve dans le sous-dossier `annexes/`. Dans ce cas le lien, depuis l'`index`, précisera le nom du dossier puis le fichier :

```
<p>Si vous voulez la page 3, <a href="annexes/page3.html">cliquez ici</a>.</p>
```

Pour revenir au dossier parent, depuis la page 3, on indique avec deux points qu'on revient dans le dossier supérieur :

```
<p>Ceci est la 3e page du site, dans le sous-dossier «annexes».<br />Vous pouvez revenir à l'index : <a  
href="..../index.html">cliquez ici</a>.</p>
```

### 4.3. Le lien vers une ancre

Le lien vers une ancre consiste à créer des liens à l'intérieur d'une seule page, par exemple quand la page est longue, avec un sommaire. On utilise l'attribut `id`.

Par exemple (ici dans `index.html`), pour un sous-titre, on ajoute l'attribut `id` :

```
<h3 id="partie-1-1">1.1. Les baleines</h3>  
<p>On écrit ce qu'on a à écrire sur le sujet</p>  
<h3 id="partie-1-2">1.2. Les dauphins</h3>  
<p>On écrit aussi ce qu'on a à écrire sur le sujet</p>
```

Pour le lien, en haut de page, on garde la structure `<a>`, mais avec le signe dièse # :

```
<p><strong>Sommaire</strong><br />  
<a href="#partie-1-1">1.1. Les baleines</a><br />  
<a href="#partie-1-2">1.2. Les dauphins</a></p>
```

Il faut que la page contienne suffisamment de texte pour qu'on observe l'effet de ces liens.

### 4.4. Le lien vers une ancre d'une autre page

On peut également aller vers la partie précise d'une autre page web de notre site.

On ajoute l'ancre avec le dièse # juste après le nom du fichier. Par exemple, depuis `page2.html`, on veut aller vers la partie sur les baleines de `index.html` :

```
<p>Accéder à nos informations sur les baleines : <a href="index.html#partie-1-1">cliquez ici</a>  
</p>
```

### 4.5. L'attribut title

En plus de l'attribut `href`, on peut ajouter l'attribut `title`, qui fait apparaître une infobulle quand on survole le lien avec la souris :

```
<p><a href="page2.html" title="Ce lien vous emmène en page 2">Vers la page 2</a></p>
```

### 4.6. Un lien pour envoyer un mail

(Déconseillé - préférez toujours un formulaire contact)

Pour les mails, on peut utiliser un lien de type `mailto` :

```
<p><a href="mailto:mon_nom@bidule.com">Envoyez-moi un mail</a></p>
```

### 4.7. Un lien pour télécharger un fichier

Comme pour une adresse html, on indique le fichier à télécharger, et le téléchargement se lance :

```
<p><a href="ma_musique.zip">Téléchargez mes morceaux !</a></p>
```

## 05 - Les images

On peut utiliser plusieurs formats d'image sur Internet :

- Le JPEG (Joint Photographic Expert Group) avec l'extension .jpg ou .jpeg.  
Ce type d'image est compressé de manière importante, beaucoup utilisé.  
On l'utilise de préférence pour les photos.
- le PNG (Portable Network Graphics ) avec l'extension .png.  
Ce type d'image accepte la transparence, et ne perd pas de qualité.  
On l'utilise de préférence, en 24 bits, pour toutes les images.
- Le GIF (Graphics Interchange Format) avec l'extension .gif.  
Ce type d'image précède le PNG, on le garde seulement pour l'avantage de pouvoir être animé.

On évite les autres formats.

Contre les problèmes, on enregistre les fichiers avec des noms en minuscules, sans espace ni accent, par exemple : mon\_image.png (le caractère « \_ » permet de remplacer un espace).

### 5.1. Insérer une image

Pour insérer une image dans une page web, on utilise la balise orpheline `<img />`.

L'image doit se trouver dans un paragraphe (`<p>`).

Il faut deux attributs, obligatoire, la source (src) et le texte alternatif (alt).

La source est le chemin de l'image, sous forme absolue ou relative. Le texte alternatif apparaît en cas de problème de téléchargement, il aide aussi pour les personnes handicapées, et pour le référencement des moteurs de recherche.

Exemple avec un chemin absolu :

```
<p>

</p>
```

Exemple avec un chemin relatif :

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

### 5.2. L'attribut title pour une image

En complément des attributs src et alt, on peut ajouter l'attribut title, qui fait apparaître une infobulle quand on survole l'image avec la souris :

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

### 5.3. Des liens sur des images

On crée des liens sur des images comme sur du texte, en encadrant l'image avec les balises `<a>` :

```
<p><a href="page2.html"></a></p>
```

### 5.4. Les figures

Pour illustrer un texte, on peut choisir les balises `<figure>`, en ajoutant une légende (`<figcaption>`). Dans ce cas la balise `<figure>` remplace l'obligation de la balise `<p>`.  
On utilise l'idée de la figure quand l'illustration est essentielle au texte.  
On peut mettre plusieurs images dans une figure.

```
<figure>


<figcaption>Le circuit du langage html</figcaption>
</figure>
```

### 5.5. Définition de zones cliquables sur une image

On crée des zones distinctes avec des liens distincts sur une image avec l'attribut usemap et les balises associées `<map>` et `<area>`.

Par exemple :

```

<map name="serveur">
<area shape="rect" coords="40,9,174,122" alt="home" title="page 2" href="page2.html">
<area shape="rect" coords="394,11,492,116" alt="servor" href="annexes/page3.html">
</map>
```

La forme de la zone peut être variée :

- pour rect (dans l'exemple) : x1, y1, x2, y2 ;
- pour circle : x, y, rayon ;
- pour poly : x1, y1, x2, y2, x3, y3, etc.

On peut mettre l'attribut title dans `<area>` pour aider l'utilisateur.)

En ligne  
<http://pixlr.com/editor/>

## 06 - Les bases CSS (1/2)

Les bases de HTML sont suffisantes pour démarrer l'apprentissage du CSS (Cascading Style Sheets ou Feuilles de style). Le CSS permet de définir la couleur du texte, la police utilisée, la taille, les bordures, le fond de la page, la mise en page...

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

- dans un fichier .css  
(la meilleure méthode, en effet une feuille de style liée peut être utilisée par plusieurs pages, voire l'ensemble des pages de votre site.)
- dans l'en-tête <head> du fichier .html
- dans le <body> du fichier .html, sous forme de styles (la moins bonne méthode)

### 6.1. Indiquer au fichier .html qu'on utilise une feuille de style .css

Pour le fichier .css, on commence par appeler la feuille de style depuis les fichiers .html.

Pour cela, on ajoute dans l'en-tête <head>, un lien (avec la balise orpheline <link />).

Le lien est de type «feuille de style» («stylesheet» dans l'attribut rel), et on donne le nom du fichier .css avec l'attribut href (dans l'exemple style.css) :

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

### 6.2. Ecrire une feuille de style .css

Sur Notepad++, on crée un nouveau fichier, que l'on nomme style.css.

Si on inscrit dans ce fichier :

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

Quand on crée un texte dans un paragraphe sur nos pages html, on remarque que ce texte apparaît en bleu dans le navigateur.

Dans ce code CSS, on trouve trois éléments :

- le nom de la balise dont on veut modifier l'apparence (dans l'exemple p pour la balise <p> des paragraphes).
- la ou les propriétés qu'on veut modifier (color dans l'exemple pour changer la couleur) ; on note une propriété par la ligne, elles sont encadrées par les accolades {}
- la valeur, indiqué pour chaque propriété après deux-points «:» (ici blue pour bleu, dans la propriété color), avec un point-virgule «;» pour terminer la ligne.

La structure est la suivante :

```
balise1 {
  propriet1:valeur;
  propriet2:valeur;
}

balise2 {
  propriet1:valeur;
}

balise3 {
  propriet1:valeur;
  propriet2:valeur;
  propriet3:valeur;
}
```

Dans l'exemple précédent, on demande que tous les paragraphes soient écrits en bleu.

On peut aussi demander que les sous-titres (<h3>) et textes d'emphase (<em>) soient écrits en rouge :

```
h3, em {
  color:red;
}
```

### 6.3. L'indentation et les commentaires

Comme dans un fichier .html, les tabulations servent à hiérarchiser les lignes de code.

On peut aussi insérer des commentaires, comme en html, avec une syntaxe différente, entre /\* et \*/ , sur plusieurs lignes ou une seule ligne.

```
/* Ceci est mon code Je le dis en commentaire */
p {
  color:blue; /* la couleur des paragraphes est bleue */
}
```

## 07 - Les bases CSS (2/2)

### 7.1. Appliquer des styles différents aux balises de même type

On peut avoir envie que les paragraphes n'aient pas tous les mêmes apparences.

Dans ce cas, on utilise les attributs class et id.

Dans le fichier .html, on va préciser la class d'un paragraphe, parmi tous.

Par exemple :

```
<h1>Titre de la page</h1>
<p class="intro">Ceci est l'introduction de ma page</p>
    <h2>Titre secondaire 1</h2>
        <h3>Sous-titre 1.1</h3>
            <p>Ceci est mon premier paragraphe.</p>
        <h3>Sous-titre 1.2</h3>
            <p>Ceci est mon <em>deuxième</em> paragraphe.</p>
    <h2>Titre secondaire 2</h2>
        <h3>Sous-titre 2.1</h3>
            <p>Ceci est mon <strong>troisième</strong> paragraphe.</p>
        <h3>Sous-titre 2.2</h3>
            <p>Ceci est mon <mark>quatrième paragraphe</mark>.</p>
```

Dans le fichier .css, on demande que seul le paragraphe qui a la class intro soit en vert :

```
.intro {
    color:green;
}
```

L'attribut id a la même fonction, mais il n'est utilisé que pour des éléments uniques, qu'on ne trouve qu'une fois (un logo, un cadre div). La syntaxe CSS comprend alors un dièse (#) au lieu du point (.).

Exemple :

- dans le fichier .html :

```

```

- dans le fichier .css :

```
#logo {
    height:120;
}
```

### 7.2. Les balises `<span>` et `<div>`

Quand on veut donner des propriétés à des éléments qui ne sont pas entourés de balises, on utilise des balises spécifiques pour le CSS.

`<span></span>` est une balise de type inline, elle s'utilise pour sélectionner une partie du texte d'un paragraphe (comme `<em>` et `<strong>`), sans retour à la ligne.

`<div></div>` est une balise de type block, elle entoure un bloc de texte (comme `<p>` et `<h1>`), avec retour à la ligne.

Exemple avec `<span>` :

- dans le fichier .html :

```
<p>Ceci est l'<span class="intro-grey">introduction</span> de ma page</p>
```

- dans le fichier .css :

```
.intro-grey {
    color:grey; /* On passe le mot choisi en gris */
    font-size:25; /* On passe le texte en taille 25, plus gros */
}
```

### 7.3. Les types de sélecteurs

Il y a plusieurs façons de créer des styles. Il est bon d'avoir bien en tête ces possibilités, en les testant :

a. On donne une propriété au contenu de la balise

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

b. une propriété à plusieurs balises

```
h3, em { color:red; }
```

c. une propriété à des balises précises

```
.class { color:red; }
```

ou

```
#id { color:red; }
```

d. une propriété à toutes les balises

```
* { /* (sélecteur universel *) */ color:red; }
```

e. une propriété à une balise contenue dans une balise

```
h3 em { color:red; /* S'applique sur le texte encadré par <em></em>, si elles sont elles-mêmes
            encadrées par <h3></h3> */ }
```

f. une propriété à une balise qui en suit une autre

```
h3 + p { color:red; /* S'applique au paragraphe <p></p> situé après </h3> */ }
```

g. une propriété à une balise qui a un attribut précis

```
a[title] { color:red; /* S'applique aux liens qui ont un attribut title */ }
```

ou

```
a[title="Cliquez ici"] { color:red; /* S'applique aux liens qui ont un attribut title
            avec le texte «Cliquez ici» */ }
```

ou

```
a[title*="ici"] { color:red; /* S'applique aux liens qui ont un attribut title qui contient le texte «ici» */ }
```

et bien d'autres encore...

En ligne

<http://www.w3.org/Style/css3-selectors-updates/WD-css3-selectors-20010126.fr.html>

<http://www.w3.org/TR/css3-selectors/>

## 08 - Formatage du texte en CSS (1/2)

### 8.1. La taille du texte

On peut indiquer avec la propriété font-size, une taille absolue (en pixels) ou une taille relative (avec des codes anglais, en pourcentage ou en em), qui s'adapte à ce que veulent les visiteurs.

- La taille absolue :

```
p {  
    font-size:16px;  
}
```

- La taille relative :

```
p {  
    font-size:small; /* le texte du paragraphe sera «petit» */  
}
```

Cela peut être minuscule (xx-small), très petit (x-small), petit (small), moyen (medium), grand (large), très grand (x-large), énorme (xx-large).

La meilleure méthode consiste à utiliser «em» : 1em donne la taille «normale» (la taille de la police), et on déplace le curseur au-dessus si on veut une taille plus grande (1.5em), au-dessous si on veut une taille plus petite (0.7em) :

```
p {  
    font-size:1.2em;  
}
```

On peut choisir de passer tous les caractères en majuscules ou en minuscules avec la propriété text-transform, et les valeurs uppercase (majuscules) ou lowercase (minuscules).

```
p {  
    text-transform: uppercase;  
}
```

### 8.2. La police

On précise quelle police afficher avec la propriété font-family

```
p {  
    font-family:Arial;  
}
```

On peut utiliser sans problème : Arial, Arial Black, Comic Sans MS, Courier New, Georgia, Impact, Times New Roman, Trebuchet MS, Verdana. Ces polices sont reconnues par les navigateurs. (\*)

On peut utiliser d'autres polices (avec @font-face), à condition de les charger parmi les fichiers du site. Dans ce cas, on télécharge une police sur <http://www.fontsquirrel.com>, site qui propose des polices que l'on a le droit d'utiliser. On choisit une police qui se télécharge en kit :

Par exemple la police Action Man : <http://www.fontsquirrel.com/fonts/Action-Man>

On télécharge le @font-face Kit. On dézippe le fichier téléchargé dans le sous-dossier polices/ du site.

Dans le fichier .css on définit la nouvelle police (le modèle à prendre est dans le zip, sous forme d'un fichier css ; on copie-colle ce qu'on souhaite).

```
@font-face {  
    font-family:'ActionManRegular';  
    src: url('polices/Action_Man-webfont.eot');  
    src: url('polices/Action_Man-webfont.eot?#iefix') format('embedded-opentype'),  
        url('polices/Action_Man-webfont.woff') format('woff'),  
        url('polices/Action_Man-webfont.ttf') format('truetype'),  
        url('polices/Action_Man-webfont.svg#ActionManRegular') format('svg');  
    font-weight: normal;  
    font-style: normal;  
}
```

Et on applique cette police à une ou plusieurs balises :

```
h3, em {  
    font-family:'ActionManRegular', Arial; /* On propose une autre police, au cas où, comme le CSS3  
n'est pas supporté par toutes les versions des navigateurs */  
    color:red;  
}
```

### 8.3. Italique, gras, souligné

Pour l'italique, c'est la propriété font-style, avec les valeurs italic, oblique ou normal.

```
h2 {  
    font-style:italic; /* Je mets tous mes titres en italique */  
}
```

Autre exemple :

```
h3, em {  
    font-family:'ActionManRegular', Arial;  
    color:red;  
    font-style:normal; /* Cela permet d'enlever ici l'italique par défaut de l'em */  
}
```

Pour le gras, on utilise la propriété font-weight, avec les valeurs bold ou normal.

```
h1 {  
    font-weight:bold; /* Je mets les grands titres en gras */  
}
```

On propose d'autres décorations avec la propriété text-decoration, avec les valeurs underline (souligné), line-through (barré), overline (ligne au-dessus), blink (clignotant), none (rien).

Exemples :

```
h3 {  
    text-decoration:underline; /* les sous-titres sont soulignés */  
}  
ou  
h2 {  
    text-decoration:underline overline; /* les titres sont à la fois soulignés et ont une ligne au-dessus */  
}  
ou  
a {  
    text-decoration:none; /* les liens ne sont pas soulignés, alors qu'ils le sont par défaut */  
    color:red; /* et ils sont noirs, et pas bleus par défaut */  
}
```

## 09 - Formatage du texte en CSS (2/2)

### 9.1. L'alignement du texte

On met en forme l'alignement des paragraphes avec la propriété `text-align`, les valeurs `left` (aligné à gauche, par défaut), `center` (centré), `right` (aligné à droite), `justify` (aligné des deux côtés).

L'alignement ne fonctionne que sur le balises de type `block` (`<p>`, `<div>`, `<h1>`, `<h2>`...), pas sur les balises de type `inline` (`<span>`, `<a>`, `<em>`...).

### 9.2. Les flottants

On utilise la propriété `float`, avec les valeurs `left` (l'image flotte à gauche) ou `right` (l'image flotte à droite).

Par exemple :

dans le fichier `.html` :

```
<p><a href="page2.html"></a>Le langage HTML a été inventé par un certain Tim Berners-Lee en 1991. Tim Berners-Lee suit encore aujourd'hui avec attention l'évolution du Web. Il a créé le World Wide Web Consortium (W3C) qui définit les nouvelles versions des langages liés au Web. Il a par ailleurs créé plus récemment la World Wide Web Foundation qui analyse et suit l'évolution du Web.<br />Il s'agit de deux langages, à la base, HTML et CSS, qui sont traduits par les navigateurs. Le HTML (HyperText Markup Language) sert à gérer et organiser le contenu de la page. Le CSS (Cascading Style Sheets, aussi appelées Feuilles de style), créé en 1996, sert à gérer l'apparence de la page. Le CSS3 (dernière version) a besoin, pour fonctionner, d'une page en HTML4 (1998) ou HTML5 (dernière version).</p>
```

dans le fichier `.css` :

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

On oblige un texte à passer sous l'image flottante avec la propriété `clear` et la valeur `both` (que ce soit `left` ou `right`).

dans le fichier `.html` :

```
<p class="dessous">Texte qui doit paraître sous l'image flottante</p>
```

dans le fichier `.css` :

```
.dessous {  
    clear:both;  
}  
  
----- UTILE  
.clear {  
    clear: both;  
line-height: 0px !important;  
font-size: 0px !important;  
height: 0 !important  
}
```

### 9.3. Formatage des listes

On peut supprimer la pastille avec la propriété `list-style-type` et la valeur `none` :

```
ul {  
    list-style-type: none;  
}
```

On peut éventuellement intégrer une petite icône à la place :

```
ul {  
    list-style-image: url('images/icones.png');  
}
```

On peut mettre les éléments de la liste l'un après l'autre avec la propriété `display` et la valeur `inline-block` :

```
li {  
    display: inline-block;  
}
```

### 9.4. `<blockquote>`, `<q>`, `<cite>` (HTML)

On peut opérer des formatages supplémentaires dans le fichier `.html`, sans CSS.

Pour une citation, on peut utiliser la balise `<blockquote>` et l'attribut `cite`, dans lequel on copie l'URL si la citation vient d'un autre site web. On utilise `<q>` pour les courtes citations.

```
<p>Cet auteur a dit:  
<q cite="http://www.machin.com/truc/page1269.html">J'écris surtout la nuit.</q></p>
```

Pour un titre, on utilise `<cite>Titre</cite>`, ce qui met le contenu en italique.

Pour une abréviation, on utilise `<abbr>`:

```
<abbr title="Organisation des Nations Unies">ONU</abbr>
```

Pour un terme qui va être défini, on utilise `<dfn>`:

```
<p>Un <dfn>data center</dfn> est un endroit où l'on stocke des serveurs numériques.</p>
```

Pour présenter les touches d'un clavier, on utilise `<kbd>`:

```
<p>Appuyer sur la touche <kbd>F1</kbd>
```

A laquelle on associe en CSS, par exemple :

```
kbd {  
    background:#f4f0d3;  
    border:1px dashed #dfcb41;  
    padding:0.1em 0.3em;  
    box-shadow:2px 2px 1px #ccc;  
}
```

Pour indiquer une mise à jour d'une partie du texte, on utilise `<ins>` ou `<del>`.

```
<p>Voici le texte d'origine, <ins datetime="2012-08-24">et on ajoute cette partie</ins>, <del datetime="2012-08-12">et on a supprimé cette partie</del>.
```

## **Lorem ipsum dolor sit amet consectetur adipiscing elit**



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa **strong**. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede [link](#) mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

## **Lorem ipsum dolor sit amet consectetur adipiscing elit**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa **strong**. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede [link](#) mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.

---

Les flottants peuvent servir à des habillages, comme sur l'image ci-dessus.



## **08/09 - (\*) Quelques précisions**

### **Serif / Sans-Serif ?**

Les termes **Serif** et **Sans-Serif** sont utilisés en typographie.

Ils désignent la présence ou l'absence d'empattements sur la forme des caractères d'une police :

#### **Quelques polices à empattement - serif**

Times ; Times New Roman ; Garamond ; Palatino ; Georgia ; Baskerville ; Caslon...

#### **Quelques polices sans empattement - sans-serif**

Arial ; Verdana ; Helvetica ; Gill Sans ; Frutiger ; Futura...

### **Collections de polices «standard» pour le Web**

*/\* Polices à empattements (serif) \*/*

font-family: «Times New Roman», Times, «Liberation Serif», FreeSerif, serif;

font-family: Georgia, «DejaVu Serif», Norasi, serif;

*/\* Polices sans empattements (sans-serif) \*/*

font-family: Arial, Helvetica, «Liberation Sans», FreeSans, sans-serif;

font-family: «Trebuchet MS», Arial, Helvetica, sans-serif;

font-family: «Lucida Sans», «Lucida Grande», «Lucida Sans Unicode», «Luxi Sans», Arial, sans-serif;

font-family: Tahoma, Geneva, Kalimati, sans-serif;

font-family: Verdana, «DejaVu Sans», «Bitstream Vera Sans», Geneva, sans-serif;

font-family: Impact, «Arial Black», sans-serif;

*/\* Polices à chasse fixe (monospace) \*/*

font-family: «Courier New», Courier, FreeMono, «Liberation Mono», monospace;

font-family: Monaco, «DejaVu Sans Mono», «Lucida Console», «Andale Mono», monospace;

**Source**

<http://fvsch.com/code/collections-polices/>

**++**

<http://www.alsacreations.com/article/lire/631-quelles-polices-pour-un-site-web.html>

<http://css.4design.tl/des-ensembles-de-polices-coherents-pour-le-web>

<http://typographisme.net/post/Les-formats-de-polices-typographiques-pour-le-Web>

<http://www.fontsquirrel.com>

<http://www.google.com/fonts/>

## 10 - La couleur et le fond

On peut utiliser les termes anglais pour choisir la couleur d'un texte ou d'un cadre, mais cela se limite à 16 couleurs. Il y a d'autres méthodes, ainsi la notation hexadécimale et la méthode RGB.  
(White, silver, gray, black, red, maroon, lime, green, yellow, olive, blue, navy, fuchsia, purple, aqua, teal)

### 10.1. La notation hexadécimale des couleurs

C'est une combinaison de chiffres (0 à 9) et de lettres (A à F), précédée d'un dièse (#). Il y a 6 caractères, qui fonctionnent par deux dans la notation (quantité de rouge, quantité de vert, quantité de bleu).

#000000 correspond à la couleur noire, #FFFFFF correspond à la couleur blanche. Ensuite, on peut se servir de GIMP pour récupérer la notation qui correspond à la couleur que l'on souhaite, ou des sites web comme <http://www.code-couleur.com/> ou <http://paletton.com> (qui propose des règles thématiques, esthétiques, sur le choix de plusieurs couleurs), <https://color.adobe.com/fr/> et <http://coolors.co/>.

On note par exemple en CSS :

```
p {  
    color:#990000;  
}
```

### 10.2. La méthode RGB

Cette autre méthode s'appuie aussi sur la quantité de rouge, de vert et de bleu (Red, Green, Blue), que l'on peut trouver dans des logiciels plus simples, sur Paint, Photofiltre ou Photoshop.

Par exemple dans Photofiltre, si on clique sur une couleur, apparaît une fenêtre avec ses caractéristiques. On voit sur la droite la quantité de rouge, de vert et de bleu (toujours de 0 à 255), dans l'exemple de l'orange on voit 255, 102 et 0. Le code sera :

```
p {  
    color:rgb(255,102,0);  
}
```

### 10.3. La couleur de fond

Pour une couleur de fond, on utilise la propriété background-color sur la balise <body> :

```
body {  
    background-color:rgb(233,238,17); /* On met un jaune en fond de page */  
    color:blue; /* On peut aussi choisir que le texte de la page soit bleu par défaut,  
    au lieu d'être noir, ce qui n'empêche pas ensuite de donner une couleur différente  
    aux paragraphes, titres, etc. */  
}
```

### 10.4. L'image de fond

On peut mettre une image de fond (comme une texture), avec background-image :

```
body {  
    background-image:url('images/texture151.jpg');  
}
```

Si elle est trop petite, l'image se répète horizontalement et verticalement (comme notre texture, créée pour qu'il n'y ait pas de raccords visibles).

On peut ajouter la propriété background-attachment, avec la valeur fixed, pour fixer le fond (scroll par défaut) :

```
body {  
    background-image:url('images/texture151.jpg');  
    background-attachment:fixed;  
}
```

On peut gérer la répétition de l'image (par défaut), avec la propriété background-repeat et les valeurs no-repeat (pas de répétition), repeat-x (répétition sur la 1<sup>ère</sup> ligne), repeat-y (répétition sur la 1<sup>ère</sup> colonne), repeat (répétition, par défaut).

Quand l'image ne se répète pas (avec background-repeat:no-repeat;), on peut préciser la position de l'image avec la propriété background-position (top ou bottom ; center, left ou right ; ou en pixels).

Par exemple :

```
body {  
    background-image:url('images/texture151.jpg');  
    background-attachment:fixed;  
    background-repeat:no-repeat;  
    background-position:top center;  
    /* ou background-position:30px 50px; /* à 30 pixels du bord gauche, à 50 pixels du haut */  
}
```

On peut combiner l'ensemble avec la propriété background :

```
body {  
    background:url('images/texture151.jpg') fixed no-repeat top center;  
}
```

(On peut mettre plusieurs images de fonds. Dans ce cas, la première image déclarée est placée au-dessus, ce qui a son importance :

```
body {  
    background:url('images/circuit_html.jpg') no-repeat top right ,  
    url('images/texture151.jpg');  
}
```

### 10.5. La transparence

Avec CSS3, on peut jouer sur la transparence avec la propriété opacity et la notation RGBA.

Pour la propriété opacity, on choisir une valeur entre 0 (transparent) et 1 (opaque).

Par exemple :

```
p {  
    opacity:0.4;  
    text-align:justify;  
    color:rgb(255,102,0);  
}
```

On peut aussi jouer sur le niveau d'opacité de la couleur de fond, avec la méthode RGBA :

```
body {  
    background-color:rgb(199,222,33); /* on mets d'abord le rgb classique pour les vieux navigateurs */  
    background-color:rgba(199,222,33,0.5); /* on ajoute le degré d'opacité à la RGB */  
}
```

# 11 - Les bordures, les ombres et les apparences dynamiques

## 11.1. Les bordures

On crée des bordures avec la propriété border :

```
h2 {  
    font-style:italic;  
    border: 3px solid red dashed;  
    /* On crée une bordure de 3 pixels, rouge, sous forme ou type de tirets */  
}
```

On peut utiliser plusieurs types de bordure :

- none : pas de bordure (par défaut) ;
- solid : un trait simple ;
- dotted : pointillés ;
- dashed : tirets ;
- double : bordure double ;
- groove : en relief ;
- ridge : autre effet relief ;
- inset : effet 3D global enfoncé ;
- outset : effet 3D global surélevé.

On peut choisir que la bordure ne soit que sur un côté avec le choix des propriétés border-top, border-bottom, border-left ou border-right.

On peut créer des bordures arrondies avec la propriété border-radius :

```
p {  
    background-color:yellow;  
    /* Pour des bordures arrondies, il faut une couleur de fond ou des bordures existantes */  
    border-radius:10px;  
}
```

Ou avec une valeur différente selon les angles :

```
p {  
    background-color:yellow;  
    border-radius:10px 5px 10px 5px;  
    /* dans l'ordre en haut à gauche, en haut à droite, en bas à droite, en bas à gauche */  
}
```

Ou encore sous forme elliptiques :

```
p {  
    background-color:yellow;  
    border-radius:20px / 10px;  
}
```

## 11.2. Les ombres

Il y a deux types d'ombre : les ombres des boîtes, les ombres du texte.

Pour les ombres des boîtes, on utilise la propriété box-shadow, avec 4 valeurs :

le décalage horizontal de l'ombre, le décalage vertical de l'ombre, l'adoucissement du dégradé, la couleur de l'ombre.

Par exemple :

```
h3 {  
    border: 1px solid red;  
    box-shadow: 5px 5px 4px red; /* l'adoucissement, facultatif, crée un flou qui adoucit l'ombre */  
}
```

Pour les ombres du texte, on utilise la propriété text-shadow :

Par exemple :

```
p {  
    color: red;  
    text-shadow: 2px 2px 4px red;  
}
```

## 11.3. Les apparences dynamiques

On peut changer l'apparence d'un élément : au survol, lors du clic, lors du focus (élément sélectionné), lorsqu'un lien a été consulté. Cela peut concerter n'importe quel élément (mais le plus souvent ce sont des liens).

Pour le survol, on utilise :hover :

```
a {  
    text-decoration: none; /* on dit que le lien n'est pas souligné */  
    color: red; /* et qu'il est rouge */  
}  
  
a:hover {  
    text-decoration: underline; /* le lien est souligné quand on le survole */  
    color: green; /* et devient vert */  
}
```

On peut modifier l'élément quand on l'active avec :active (rarement).

```
a:active {  
    text-decoration: overline;  
    color: black;  
}
```

Ou quand l'élément est sélectionné avec :focus :

```
a:focus {  
    text-decoration: line-through;  
}
```

Ou lorsque le lien a déjà été visité avec :visited :

```
a:visited {  
    text-decoration: none;  
    color: blue;  
}
```

En ligne

<http://css3generator.com/>

<http://debray.jerome.free.fr/?outils/Generateur-de-box-shadow-css3>

## 12 - Structure de la page (mise en page)

On peut structurer la page, en HTML5, dans <body>, avec des balises précises :

```
<header>Là on met le texte qui vient dans l'en-tête de la page.</header>
<footer>A l'inverse, on met ce qui vient en bas de page.</footer>
```

<nav> On note ici le sommaire du site, par exemple sous forme de liste :

```
<nav>
  <ul>
    <li><a href="index.html">Accueil</a></li>
    <li><a href="page2.html">Page 2</a></li>
    <li><a href="annexes/page3.html">Page 3</a></li>
  </ul>
</nav>
```

<section>On met là une section de contenu qui se trouve ailleurs dans la page.

Chaque section forme un bloc dans la page.</section>

<p>Chaque section peut constituer une partie de la page,  
avec par exemple <h2>Titre</h2> au début de chaque section.</p>

<aside>On met dans la balise aside un contenu qui est à part, comme un encadré.</aside>

<article>On peut se servir de article pour un article indépendant du reste sur la page.</article>

Avant on utilisait <div> pour tout cela, et on précisait tout avec des propriétés class dans le fichier .css.

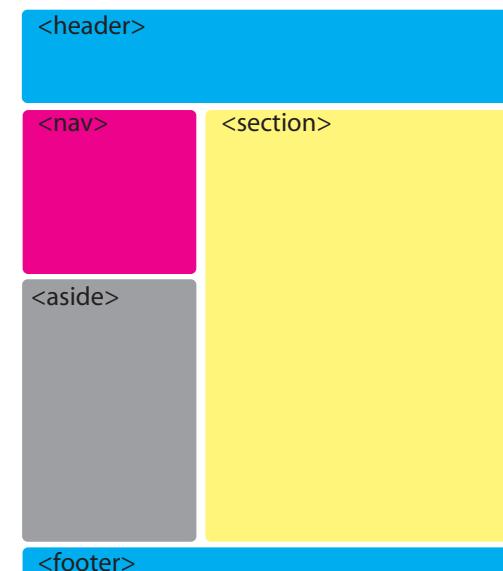
Ce nouveau système permet plus de lisibilité pour les navigateurs et les moteurs.

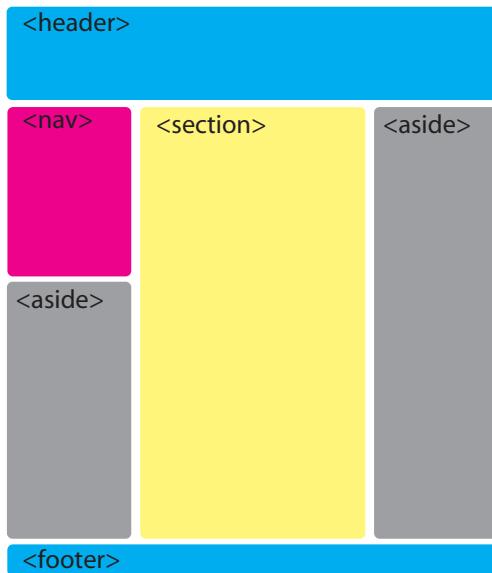
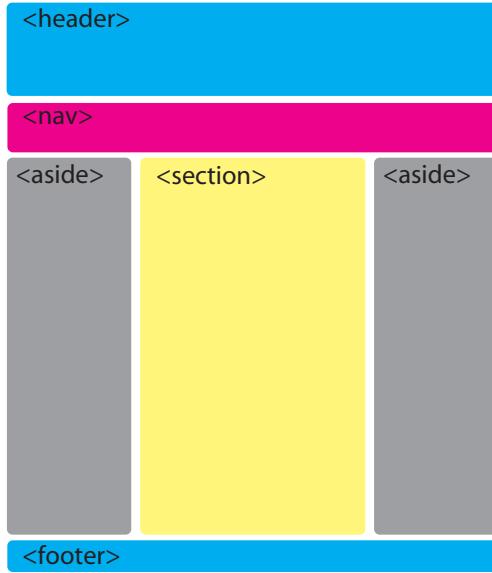
Pour que ces nouvelles balises soient reconnues par les anciennes versions d'Internet Explorer,  
il faut ajouter le script suivant dans <head> :

```
<!--[if lt IE 9]>
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
```

### En ligne

Inspiration = <http://www.alsacreations.com/static/gabarits/liste.html>  
<http://on-air.hiseo.fr/liens/ready-to-use-HTML5-CSS3-template/>  
<http://www.pvmsgarage.com/2010/04/touch-the-future-create-an-elegant-website-with-hmtl-5-and-css3/>  
<http://html5up.net/>  
<http://www.vectorskin.com/blog/structure-html5/>  
<http://www.journaldunet.com/developpeur/client-web/les-classes-conditionnelles-html.shtml>





## 13 - Les modèle des boîtes

Les balises de la fiche 12 sont des balises de type block (comme `<p>` ou `<h1>`).

La balise universelle du type block est `<div>` (contre `<span>` pour le type inline).

### 13.1. Les dimensions

On donne les dimensions au bloc avec les propriétés `width` (largeur) et `height` (hauteur), en px (pixels) ou en % (pourcentage de la page) :

```
section {  
    width: 50%; /* la section s'affiche sur la moitié de la page en largeur */  
    height: 100px; /* et sur 100 pixels en hauteur */  
}
```

On peut aussi donner une largeur minimale à une section (`min-width`), ou maximale (`max-width`), une hauteur minimale (`min-height`), ou maximale (`max-height`), ce qui permettra d'adapter l'affichage selon la résolution de l'internaute.

### 13.2. Les marges

Il y a deux types de marges (cela fonctionne pour les block et les inline) :

- les marges intérieures (entre le cadre et le texte, avec la propriété `padding`) ;
- les marges extérieures (entre le cadre et les autres éléments de la page, avec `margin`) .

Par exemple :

```
p {  
    width: 50%;  
    border: 1px black solid;  
    padding: 12px; /* on ajoute une marge intérieure au paragraphe, de 12 pixels */  
    margin: 50px; /* on ajoute une marge extérieure de 50 pixels */  
}
```

Par défaut, la même valeur s'applique aux 4 côtés.

Mais on peut préciser 4 valeurs différentes, dans l'ordre haut (top), droite (right), gauche (left), bas (bottom), ou demander seulement qu'une propriété soit modifiée parmi les quatre.

Par exemple :

```
p {  
    width: 50%;  
    border: 1px black solid;  
    padding: 6px 12px 12px 6px;  
    margin-top: 50px;  
    margin-bottom: 30px;  
}
```

### 13.3. Centrer un bloc

Pour centrer un bloc :

on lui donne une largeur avec la propriété `width`, et on demande des marges automatiques avec `margin-left:auto; margin-right:auto;`

Par exemple :

```
h2 {  
    border: 3px rgb(255,0,0) dashed;  
    width: 300px;  
    margin-left: auto;  
    margin-right: auto;  
}
```

### 13.4. Quand le texte dépasse du cadre...

Si le cadre est trop petit pour le texte, on peut gérer l'affichage du texte, avec la propriété `overflow`.

Par défaut, la valeur est `visible` (le texte continue sous le cadre).

On peut choisir la valeur `hidden` (le texte est coupé net au niveau du cadre ; on l'utilise rarement), `scroll` (le texte est coupé mais le navigateur propose des barres de défilement, en bas et à droite, pour voir tout le texte), ou `auto` (ce qui est la meilleure option, le navigateur gère lui-même ce qu'il faut faire pour qu'on puisse voir le texte).

On coupe les mots trop long (par exemple une URL) avec la propriété `word-wrap` et la valeur `break-word` :

```
p {  
    width: 50%;  
    border: 1px black solid;  
    overflow: auto;  
    word-wrap:break-word;  
}
```



En ligne

<http://fr.html.net/tutorials/css/lesson10.php>

<http://oseox.fr/css/marge.html>

[http://css.mammouthland.net/problems-css-margins-margin-padding.php](http://css.mammouthland.net/problems-css-marges-margin-padding.php)

## 13 - Les modèle des boîtes - Une alternative (box-sizing)

La boîte est un élément clef... Nous allons nous y attarder et le repenser...

Avant la boîte, il y a le contenu (content) - ce que contient la boîte - texte ou image, ...

Puis vient le padding, puis le border et enfin le margin, soit :

**Content** : le contenu, c'est à dire le texte ou l'image

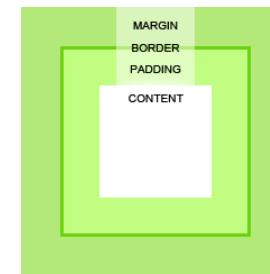
**Padding** : crée un espace autour du contenu.

Cet espace a le même background que la boîte Content.

**Border** : une bordure qui entoure le contenu et son padding.

Elle peut avoir elle-même son propre background, son épaisseur, sa texture, ses motifs.

**Margin** : la marge crée un espace autour des éléments précédents, cet espace est toujours transparent et n'est jamais affecté par les background de ces éléments.



Si je définis une boîte comme ceci :

```
.maboite {  
    width: 100px;  
    padding: 10px;  
    border: 5px black solid;  
    margin: 10px;  
}
```

Il faut l'interpréter comme un élément de :  
100px (largeur)  
+ (10px x 2) (padding)  
+ (5px x 2) (border)  
+ (10px x 2) (margin)  
**= 150px**

Il existe une alternative, Box Sizing, qui ne prend pas en compte les marges, mais intègre le border et le padding.

Dans l'exemple précédent mon contenu = 100px.

Ici, c'est bien la boîte qui fait 100px à laquelle on va déduire la valeur de la border et du padding, pour déterminer la largeur du contenu. Soit :

100px (largeur) - (10px x 2) (padding) - (5px x 2) (border) = **70px** (la largeur maximale du contenu)

### La syntaxe CSS

```
{box-sizing: content-box|border-box|initial|inherit;}
```

Brièvement, les valeurs initial et inherit : "initial" renvoie à des valeurs par défaut et "inherit" à l'élément parent qui contient l'élément.)

"content-box" nous renvoie à la page précédente - la largeur définie ne tient compte alors que du contenu (il faut donc ajouter le padding et le border (+ les marges).

"border-box" inclut le contenu + le padding + le border et exclut les marges.

Pour l'utiliser, il nous faut activer le box-sizing dans la feuille de style.

```
.maboite {  
    box-sizing: border-box;  
    width: 100px;  
    padding: 10px;  
    border: 5px black solid;  
    margin: 10px;  
}  
* {  
    box-sizing: border-box;  
}
```

Mon contenu est de 70px max à l'intérieur de la boîte et seuls les éléments dont la classe sera "maboite", seront affectés.

Box-sizing est appliqué à tous les éléments (la largeur, le padding et le border seront affectés au cas par cas, par d'autres classes)

Néanmoins, pour être sûr que la compatibilité soit assurée avec l'ensemble des plus récents navigateurs, il faudra déclarer ceci dans la feuille de style...

```
.maboite {  
    -webkit-box-sizing: border-box; /* Pour les anciennes versions des navigateurs WebKit */  
    -moz-box-sizing: border-box; /* Pour tous les navigateurs Gecko */  
    box-sizing: border-box;  
    width: 100px;  
    padding: 10px;  
    border: 5px black solid;  
    margin: 10px;  
}  
* {  
    -webkit-box-sizing: border-box; /* Pour les anciennes versions des navigateurs WebKit */  
    -moz-box-sizing: border-box; /* Pour tous les navigateurs Gecko */  
    box-sizing: border-box;  
}
```

Pourquoi l'utiliser ? Pour éviter des montagnes de calculs qui soudain deviennent inutiles.

Notamment dans une mise page fluide. Imaginons deux <div> de 50% de large chacun, que nous souhaitons l'un à côté de l'autre, chacun possède sa marge, sa bordure et son padding...

Les 50% font écho à la largeur de l'écran (laquelle ? celle de l'utilisateur ! Quel écran possède-t-il ?

Pas la moindre idée, mais il s'agit bien d'une mise en page fluide, alors, la taille définie n'est pas importante...

Les marges seront, bien sûr, définie en pixels, tout comme l'épaisseur de la bordure et le padding... en % ou pixels... et pourquoi pas l'ensemble... (?!)

Avec box-sizing rien de tout cela ! Nous définissons notre largeur et notre padding et le tour est joué.

En conclusion et pour le dire simplement : les mises en pages ne sont pas toujours simples et s'avèrent souvent des casse-tête. Box-sizing vous fera gagner du temps et vous aidera à coder plus efficacement.

Source = <http://la-cascade.ghost.io/box-sizing-pour-les-nuls/>

## 13 - Les modèle des boîtes - Une alternative (flexbox)

### Qu'est qu'une flex-box ?

Flexbox est une nouvelle propriété CSS qui nous permet de gérer simplement la mise en page d'une série d'éléments au sein d'un élément parent et plus précisément de :

- Mettre en place plusieurs éléments sur une ligne sans avoir à se soucier de la largeur de chacun d'eux.
- Modifier rapidement le sens de lecture vertical-horizontal.
- Aligner les éléments à gauche, à droite ou au centre du parent.
- Modifier l'ordre d'affichage des différents éléments.
- Déployer les éléments dans le parent sans problème pour gérer les marges ou la taille des éléments.

Flexbox n'est pour le moment utilisable que sur les versions les plus récentes de nos navigateurs, et avec des préfixes pour certains.

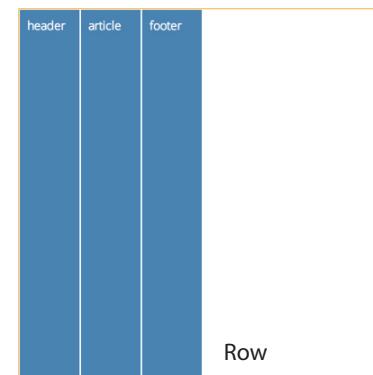
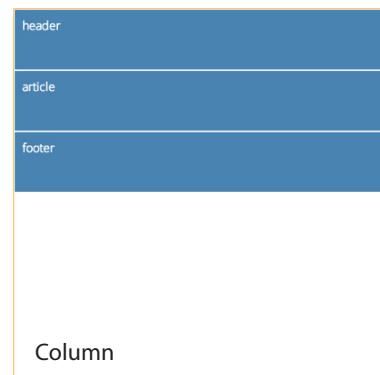
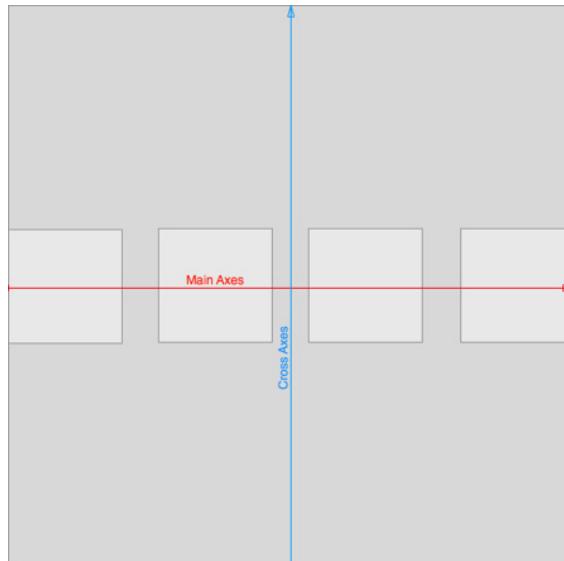
### Créons notre .flexcontainer

Tout d'abord il faut créer un conteneur pour nos éléments flex :

```
.flexcontainer {  
    display: flex;  
}
```

**À savoir :** Les flexbox utilisent un principe d'axes. Si on veux des éléments alignés verticalement, on utilise une colonne (column), pour des éléments horizontaux (par défaut) on utilise une ligne (row). Nous aurons donc 2 axes appelés "Main axis", définis par le sens de lecture, et "Cross Axis" qui est perpendiculaire au premier.

```
.flexcontainer {  
    display: flex;  
    flex-flow: column;  
/* ou */flex-flow: row;  
/* flex-flow peut aussi s'écrire flex-direction */}
```



Pour définir l'ordre d'affichage des éléments directement depuis CSS nous pouvons utiliser rowreverse ou column-reverse.

### Utilisons les Axes

Nous avons la possibilité de placer précisément les éléments dans notre flexbox, c'est ici que le principe d'axe vertical et horizontal prend son importance. Pour aligner les éléments à partir du début du main-axis, nous ajouterons la propriété justify-content: flex-start.

#### À retenir :

```
justify-content: main-axis | align-items: cross-axis  
.flexcontainer {  
    flex-flow: row;  
    align-items: flex-end;  
}
```

Grâce à flexbox, nous avons enfin la possibilité de centrer un élément au sein d'un autre aussi bien horizontalement que verticalement.

```
.flexcontainer {  
    flex-flow: column;  
    justify-content: center;  
}
```

### Aligner verticalement devient un jeu d'enfant !

Il est aussi possible de mettre des marges entre nos éléments, afin de remplir un espace donné ou simplement de séparer différents éléments. Pour cela nous utilisons à nouveau la propriété justify-content avec pour argument space-around. Les éléments de notre .flexcontainer seront séparés d'une marge identique, et d'une demi-marge sur les cotés. L'argument space-between aura le même effet sans les marges extérieures aux éléments.

Pour le cross-axis nous avons la possibilité de définir un alignement flex-end, flex-start, center et stretch. Ces 2 premiers arguments positionnent les éléments sur les extrémités de l'axe. "Center" place les éléments au centre du .flexcontainer et l'argument strech nous permet d'étirer les éléments afin qu'ils prennent la largeur ou hauteur du .flexcontainer.



## Définir une taille

C'est bien pratique toutes ces nouvelles propriétés mais j'ai souvent besoin d'éléments plus grands les uns que les autres.

Voyons comment nous pouvons donner une taille à nos différents éléments dans la flexbox.

```
#first {      #second {      #third {  
    flex: 1;      flex: 2;      flex: 1;  
}  
}  
}
```

Le premier argument correspond au flex-grow. Il n'y a pas d'unité, nous utilisons des proportions.

Si ces 3 éléments ont flex: 1; alors ils feront tous la même taille.

Si l'un des ces éléments avait pour valeur 2, il prendrait 2 fois plus de place que les autres.



Il est aussi possible de définir une taille de base grâce à flex-basis. Lorsque le navigateur calculera la taille des éléments dans le .flexcontainer, l'espace restant après l'addition des différentes tailles des éléments sera divisé proportionnellement à la propriété flex-grow.

Il existe enfin la propriété flex-shrink qui fait la même chose que flex-grow mais pour réduire les éléments dans une flexbox trop petite.

**À retenir : La propriété flex est un raccourci pour les propriétés :**

flex-grow | flex-shrink | flex-basis - soit flex: (grow) (shrink) (basis)

```
.element {  
  flex: 1 1 100px;  
}
```

## Conclusion

Les flexbox devraient nous permettre de nous passer de la plupart des systèmes de grille existants, qui sont souvent lourds, peu sémantiques et parfois encombrants.

Une logique relativement simple, une gestion naturelle du responsive, des possibilités de mise en page bien plus avancées que celles que nous connaissons.

Les flexbox sont l'outil idéal pour concevoir vos pages directement dans le navigateur tout en vous laissant une grande flexibilité.

Source = [http://www.synbioz.com/blog/mise\\_en\\_page\\_flexbox](http://www.synbioz.com/blog/mise_en_page_flexbox)

En ligne

<http://www.alsacreations.com/tuto/lire/1493-css3-flexbox-layout-module.html>  
<http://www.sketchingwithcss.com/samplechapter/cheatsheet.html>  
<http://css-tricks.com/snippets/css/a-guide-to-flexbox/>  
<http://la-cascade.ghost.io/flexbox-guide-complet/>

# 14 - Le positionnement en CSS

## 14.1. La transformation des éléments

Avec la propriété display, on peut transformer un élément inline en block, un block en inline :

```
a{  
    display:block;  
}
```

On peut aussi rendre un élément non visible :

```
.secret {  
    display:none;  
}
```

## 14.2. Le positionnement inline-block

On peut aussi passer les éléments en inline-block, avec la propriété display.

Alors les éléments se placent les uns à côté des autres, et on peut choisir leurs dimensions.

On peut alors aussi jouer sur l'alignement vertical de ces blocs.

Par défaut, ils se positionnent sur une ligne inférieure. Mais on peut utiliser la propriété vertical-align, avec les valeurs baseline (par défaut), top (en haut), middle (centré), bottom (en px ou % au-dessus de la baseline).

Par exemple :

```
nav {  
    display:inline-block;  
    width:150px;  
    border: 1px black solid;  
    vertical-align: top;  
}  
  
section {  
    display:inline-block; /* la section ne prendra pas le reste de la largeur de la page, si elle n'en a pas  
    besoin, parce que ce n'est plus un block */  
    border: 1px blue solid;  
    vertical-align: top;  
}
```

Pour les versions antérieures à IE8, la valeur inline-block est mal prise en compte.

Il faut créer une autre feuille de style (style-ie.css), et l'insérer dans le <head> :

```
<link rel="stylesheet" src="style.css" />  
<!-- [if lte IE7]-->  
<link rel="stylesheet" src="style-ie.css" />  
<! [endif]-->
```

Pour chaque élément positionné en inline-block, il faut ajouter dans cette feuille :

```
.element1 { /* le nom de l'élément concerné */  
    display: inline;  
    zoom: 1;  
}
```

## 14.3. Les positionnements absolu, fixes et relatifs

Le positionnement absolu permet de préciser l'endroit où on veut mettre l'élément.

Le positionnement fixe reprend le même principe, mais sans défilement de l'élément.

Le positionnement relatif permet de décaler l'élément par rapport à sa position normale.

Pour le positionnement absolu, on utilise la valeur absolute, et on donne les coordonnées par rapport aux bords de la page (avec top, left, right et bottom) :

```
.cadre {  
    position: absolute;  
    top: 50px; /* le cadre commence à 50 pixels du haut */  
    left: 100 px; /* le cadre commence à 100 pixels du côté gauche */  
}
```

S'il y a superposition entre deux cadres, on utilise la propriété z-index pour les hiérarchiser :

```
.cadre1 {  
    position: absolute;  
    top: 50px; /* le cadre commence à 50 pixels du haut */  
    left: 100 px; /* le cadre commence à 100 pixels du côté gauche */  
    z-index: 1;  
}  
  
.cadre2 {  
    position: absolute;  
    top: 50px; /* le cadre commence à 50 pixels du haut */  
    left: 140 px; /* le cadre commence à 140 pixels du côté gauche */  
    z-index: 2; /* le contenu du cadre2 apparaît au-dessus du contenu du cadre1 */  
}
```

Attention ! Quand un block se trouve dans un autre block, le positionnement se fait par rapport au coin supérieur gauche du premier block (et plus par rapport à la page).

On peut aussi utiliser, mais plus rarement, les valeurs fixed et relative (à tester).

En ligne

Des structures toutes prêtes sur le site :  
<http://freehtml5templates.com/>

## 15 - Les tableaux

### 15.1. Syntaxe des tableaux

Un tableau est créé dans les balises `<table></table>`.

A l'intérieur :

- on crée une ligne avec `<tr></td>` ;
- on crée une colonne avec `<td></td>`.

Par exemple :

```
<table>
  <tr>
    <td>Frantz</td>
    <td>25 ans</td>
    <td>Allemagne</td>
  </tr>
  <tr>
    <td>Alberto</td>
    <td>24 ans</td>
    <td>Espagne</td>
  </tr>
  <tr>
    <td>Michael</td>
    <td>25 ans</td>
    <td>Irlande</td>
  </tr>
</table>
```

Pour avoir les bordures, on passe par le fichier `.css` :

```
table {
  border-collapse: collapse;
}
td {
  border: 1px black solid;
  padding: 5px; /* avec une marge intérieure pour ne pas coller le texte à la bordure */
}
```

Dans le fichier `.html`, on ajoute des titres de colonnes avec `<th>` (sur une ligne `<tr>` après `<table>`) :

```
<tr>
  <th>Prénom</th> /* par défaut ce qui est entre <th> est gras et centré */
  <th>Âge</th>
  <th>Pays</th>
</tr>
```

Dans le fichier `.css`, on ajoute les propriétés pour `th` :

```
td, th {
  border: 1px black solid;
  padding: 5px; /* avec une marge intérieure pour ne pas coller le texte à la bordure */
}
```

Dans le fichier `.html`, on ajoute un titre au tableau avec `<caption>` (juste après `<table>`) :

```
<caption>Equipe de football</caption> <!-- qu'on peut aussi personnaliser en css--&gt;</pre>
```

### 15.2. Structuration de grands tableaux. Fusion

Quand on a des tableaux importants, on peut mettre une ou plusieurs lignes dans des cadres structurants : `<thead>` (pour l'entête), `<tfoot>` (pour le bas), `<tbody>` (pour le corps du tableau).

On peut fusionner des cellules, dans le fichier `.html`.

Pour fusionner deux lignes, on écrit :

```
<td colspan="2">Texte qui prendra deux cellules en ligne</td>
```

Pour fusionner deux colonnes, on écrit :

```
<tr rowspan="2">Texte partagé sur deux cellules en colonne</tr>
```

```
<table> (width=...px / style="width:...px" / class="nomdustyledéfinidanslecss")
```

<caption> (option titre du tableau) </caption>	
<thead> <th>Titre colonne 1 </th>	<th>Titre colonne 2 </th>
<tr> <td> </td>	<td> </td>
<tr> <td> (colspan=2) </td>	<td> </td>
<tr> <td> (rowspan=2) </td>	<td> </td>
</tr> <td> </td>	<td> </td>
<tfoot> <th>Titre colonne 1 </th>	<th>Titre colonne 2 </th>

```
</table>
```

## 16 - Les balises META

### C'est quoi une balise META?

Les balises META servent à placer des métadonnées (metadata) dans une page HTML. On placera ces informations dans l'élément head, et elles ne seront pas affichées sur la page.

À qui sont destinées ces informations? Eh bien, à tous les outils susceptibles de les exploiter. Et notamment :

- aux navigateurs web
- aux moteurs de recherche;
- plus largement, à tous les outils d'indexation, c'est-à-dire tous les outils qui analysent vos pages pour y identifier diverses informations.

### Les incontournables

#### Élément title (utile et obligatoire, mais pas une balise META)

De nombreux sites parlent de «META Title» pour désigner cet élément, bien qu'il ne s'agisse absolument pas d'une balise META. Il s'agit d'un élément HTML différent de meta, qui s'écrit généralement ainsi :

```
<title>Titre de la page</title>
```

Le contenu de cet élément n'apparaît pas directement dans la page, mais est utilisé de diverses manières :

- par le navigateur web, dans le titre de la fenêtre, les onglets, comme titre des marque-pages (favoris), etc.;
- par les moteurs de recherche et divers outils, comme titre d'un résultat de recherche par exemple.

Cet élément est par ailleurs obligatoire en HTML.

Enfin, notez que son importance pour le référencement des pages est primordiale.

Un élément title bien rédigé devrait, dans l'idéal :

- être lisible pour un lecteur humain,
- être informatif ou attrayant pour un lecteur humain,
- refléter le contenu de la page,
- contenir des mots-clé pertinents,
- faire moins de 70 caractères (pas une règle absolue, mais Google par exemple coupe les titres trop longs entre 65 et 70 caractères dans les pages de résultat, et les titres trop longs diluent le poids des différents mots-clé).

#### META Content-Type (très utile)

Il s'agit d'un équivalent, placé dans le code HTML, de l'en-tête HTTP Content-Type, qui permet d'indiquer le type de document et son encodage.

Utiliser cette balise META n'est pas indispensable. En effet, les informations qu'elle donne devraient logiquement être envoyées par le serveur web, au moment où il envoie la page HTML au navigateur. Mais redoubler cette information dans la page HTML elle-même

permet d'avoir une information sur l'encodage des caractères utilisé (UTF-8 ou ISO-8859-1 ou d'autres encore), ce qui s'avèrera très utile dans les cas suivants :

- Le serveur web, mal configuré, n'envoie pas les informations sur l'encodage.
- Le navigateur web va alors tenter de les trouver dans le code HTML de la page.

```
<meta charset="utf-8">
```

Il est recommandé de la placer au tout début de la section `<head>`, surtout avant `<title>`, car ce dernier peut déjà être concerné par l'encodage des caractères.

#### META Description (utile)

Cette META permet de fournir une description courte (deux à trois phrases courtes) de la page web. Cette description est notamment utilisée (dans certains cas) par les principaux moteurs de recherche, pour fournir le court texte d'aperçu des pages de résultats, ou bien par les réseaux sociaux pour définir le paragraphe de texte associé au lien partagé.

Pour rappel, voici un exemple de balise META Description :

```
<meta name="description" content="Tout sur les apparitions muettes de la Banane Masquée  
dans le cinéma colombien des années 20 et 30." />
```

On conseille généralement de ne pas utiliser la même description pour les différentes pages d'un site. Dans l'idéal, chaque page aura sa propre description habilement composée. Voici quelques conseils pour la rédaction de cette description :

- elle doit être unique, et pas dupliquée sur d'autres pages du site;
- elle doit refléter fidèlement le contenu de la page;
- elle doit être attractive pour l'utilisateur dont le regard «scanne» une page de résultats de moteur de recherche;
- elle doit reprendre les principaux mots-clé de la page et être plutôt brève (on recommande de ne pas dépasser 200 caractères, bien qu'il ne s'agisse pas d'une règle stricte).

On ne dispose pas toujours d'un contenu spécifique pour la META Description de chaque page. Dans ce cas, trois stratégies possibles :

- ne pas utiliser la balise META Description du tout;
- utiliser une META Description seulement pour certaines pages clé, ou uniquement pour la page d'accueil;
- extraire un contenu des contenus disponibles en base de données, par exemple le texte d'accroche pour une page d'article ou le début de la description du produit pour une page produit.

Notez qu'une page ne sera pas «pénalisée» par un moteur de recherche parce qu'elle n'utilise pas de META Description. Mais si vous n'en utilisez pas, faites attention à ne pas utiliser la balise du tout, et évitez les balises sans contenu :

```
<meta name="description" content="" /> <!-- À éviter -->
```

Dernière chose : fournir une META Description ne garantit pas qu'elle sera effectivement utilisée par les moteurs de recherche dans leurs pages de résultats!

Également :

META Keywords (obsolète)

META Generator (utile dans certains cas)

META Author, META Copyright (utilité faible)

etc.

Source = <http://www.alsacreations.com/article/lire/628-balises-meta.html>

## 17 - Le "responsive web design"

Cette avancée technologique peut être considérée une nouvelle philosophie de création de site.

**Le "responsive web design" est né suite à un besoin grandissant.**

En quelques années le nombre d'appareils et de résolutions servant à consulter des sites web a tout simplement explosé :

- ordinateurs, smartphones, tablettes, web TV
- format portrait, paysage, etc.

(Entre 2005 et 2008 on a identifié pas moins de 400 résolutions d'écran différentes pour tous les appareils vendus.)

Le problème est qu'on ne peut pas – ou du moins ce ne serait pas très pratique – développer autant de versions d'un site qu'il y a de résolutions différentes.

La solution à ce problème grandissant s'est donc imposée d'elle-même : le responsive design.

### Définition du responsive web design

Cette évolution des techniques de webdesign, d'intégration et de développement se base sur un objectif simple :

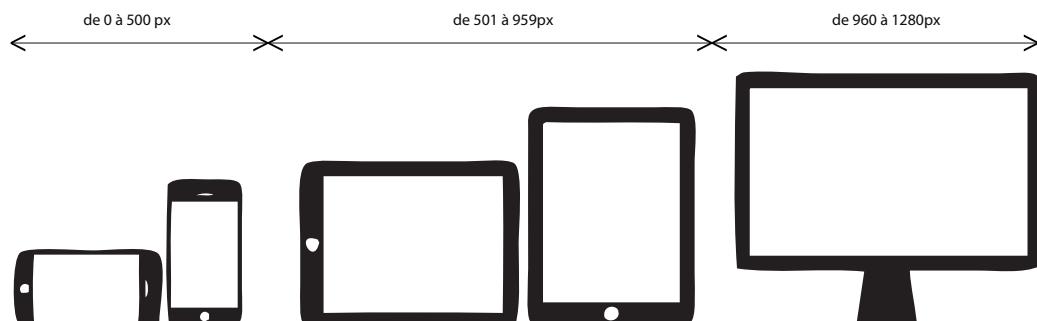
**un même site doit pouvoir s'adapter lui-même aux différentes résolutions d'écrans... et donc son contenu aussi !**

Les colonnes par exemple peuvent s'ajuster, se déplacer, voire disparaître.

Les images se redimensionnent, se replacent et il en va de même pour de nombreuses choses.

Cela ne s'arrête pas à quelques modifications pour que le tout s'adapte aisément mais il faut repenser toute l'approche de création d'un site Internet.

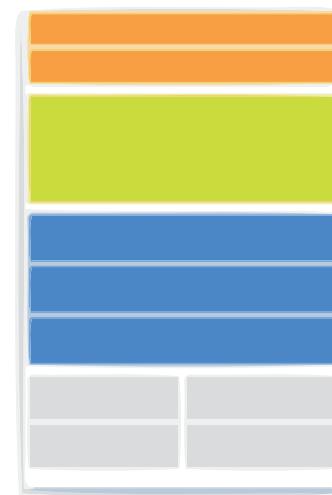
En effet, on ne navigue pas sur un smartphone comme sur un ordinateur.



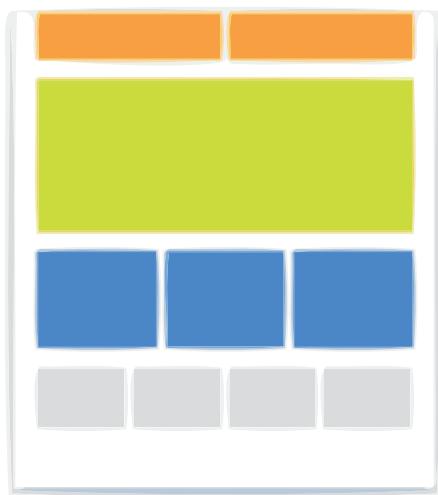
(320px)



(768px)



(1024px)



Tester un site

<http://dfcb.github.io/Responsivator/>

Voir pour l'exemple

<http://mediaqueri.es/>

### Par exemple...

#### Portrait Tablet

```
@media (min-width: 481px) and (max-width: 768px) {  
  /*ici les déclarations à modifier,...*/  
  body {  
    background-color: blue;  
  }  
}
```

#### Landscape smart phone

```
@media (min-width: 321px) and (max-width: 480px) {  
  /*ici les déclarations à modifier,...*/  
  body {  
    background-color: red;  
  }  
}
```

#### Portrait smart phone

```
@media (max-width: 320px){  
  /*ici les déclarations à modifier,...*/  
  body {  
    background-color: yellow;  
  }  
}
```

A la suite des styles déclarés dans votre feuille CSS, vous pouvez glisser ces codes et les compléter par les corrections que vous souhaitez apporter aux précédentes déclarations. Si, par exemple le fond du body était déclaré "background-color: white;", vous le verriez changer de couleur en fonction du terminal et de sa résolution.

## Annexe - Memento des balises HTML

### Balises de premier niveau

Les balises de premier niveau sont les principales balises qui structurent une page HTML. Elles sont indispensables pour réaliser le code minimal d'une page web.

Balise	Description
<html>	Balise principale
<head>	En-tête de la page
<body>	Corps de la page

### Balises d'en-tête

Ces balises sont toutes situées dans l'en-tête de la page web, c'est-à-dire entre <head> et </head> :

Balise	Description
<link />	Liaison avec une feuille de style
<meta />	Métadonnées de la page web (charset...)
<script>	Code JavaScript
<style>	Code CSS
<title>	Titre de la page

### Balises de structuration du texte

Balise	Description
<abbr>	Abréviation
<blockquote> ou <q>	Citation longue ou Citation courte
<cite>	Citation du titre d'une œuvre ou d'un évènement
<sup> ou <sub>	Exposant ou Indice
<strong>, <em>, <mark>	Mise en valeur forte, normale, visuelle
<h1>, <h2>, <h3>, <h4>, <h5>, <h6>	Titres de niveau 1 à 6
<img />	Image
<figure> et <figcaption>	Figure (image, code, etc.) et Description de la figure
<audio> et <video>	Son et Vidéo
<source>	Format source pour les balises <audio> et <video>
<a>	Lien hypertexte
 	Retour à la ligne
<p>	Paragraphe
<del> ou <ins>	Texte supprimé ou Texte inséré
<dfn>	Définition
<kbd>	Saisie clavier
<pre>	Affichage formaté (pour les codes sources)

### Balises de listes

Cette section énumère toutes les balises HTML permettant de créer des listes (listes à puces, listes numérotées, listes de définitions...)

Balise	Description
<ul>	Liste à puces, non numérotée
<ol>	Liste numérotée
<li>	Élément de la liste à puces
<dl>	Liste de définitions
<dt>	Terme à définir
<dd>	Définition du terme

### Balises de formulaire

Balise	Description
<form>	Formulaire
<fieldset>	Groupe de champs
<legend>	Titre d'un groupe de champs
<label>	Libellé d'un champ
<input />	Champ de formulaire (texte, password, bouton...)
<textarea>	Zone de saisie multiligne
<select>	Liste déroulante
<option>	Élément d'une liste déroulante
<optgroup>	Groupe d'éléments d'une liste déroulante

### Balises sectionnaires

Ces balises permettent de construire le squelette de notre site web.

Balise	Description
<header>	En-tête
<nav>	Liens principaux de navigation (menu)
<footer>	Pied de page
<section>	Section de page
<article>	Article (contenu autonome)
<aside>	Informations complémentaires

### Balises génériques

Les balises génériques sont des balises qui n'ont pas de sens sémantique.

En effet, toutes les autres balises HTML ont un sens : <p> signifie Paragraphe, <h2> signifie Sous-titre, etc. Parfois, on a besoin d'utiliser des balises génériques (aussi appelées balises universelles) car aucune des autres balises ne convient. On utilise le plus souvent des balises génériques pour construire son design. Il y a deux balises génériques : l'une est inline, l'autre est block.

Balise	Description
<span>	Balise générique de type inline
<div>	Balise générique de type block

## Annexe - Memento des balises CSS

### Propriétés de mise en forme du texte

Je résume ici la plupart des propriétés de mise en forme du texte. C'est tout ce qui touche à la présentation du texte proprement dit : le gras, l'italique, le souligné, la police, l'alignement, etc.

Propriété	Description
font-family	Nom de police
@font-face	Police personnalisée
font-size	Taille du texte
font-weight	Gras
font-style	Italique
text-decoration	Soulignement, ligne au-dessus, barré ou clignotant
font-variant	Petites capitales
text-transform	Capitales
font	Super propriété de police, qui combine font-weight, font-style, font-size, font-variant, font-family
text-align	Alignement horizontal
vertical-align	Alignement vertical (uniquement en cellules de tableau ou éléments inline-block)
line-height	Hauteur de ligne
text-indent	Alinéa
white-space	Césure
word-wrap	Césure forcée
text-shadow	Ombre de texte

### Propriétés de couleur et de fond

Propriété	Description
color	Couleur du texte
background-color	Couleur de fond
background-image	Image de fond
background-attachment	Fond fixe
background-repeat	Répétition du fond
background-position	Position du fond
background	Super propriété du fond. Combine les 5 précédentes.
opacity	Transparence

### Propriétés des boîtes

Propriété	Description
width et height	Largeur et Hauteur
min-width ou max-width	Largeur minimale ou Largeur maximale
min-height ou max-height	Hauteur minimale ou Hauteur maximale

margin	Super-propriété de marge. Combine : margin-top, margin-right, margin-bottom, margin-left.
padding	Super-propriété de marge intérieure. Combine : padding-top, padding-right, padding-bottom, padding-left.
border-width	Épaisseur de bordure
border-color	Couleur de bordure
border-style	Type de bordure
border	Super-propriété de bordure. Combine border-width, border-color, border-style. Existe aussi en version border-top, border-right, border-bottom, border-left.
border-radius	Bordure arrondie
box-shadow	Ombre de boîte
<b>Propriétés de positionnement et d'affichage</b>	
Propriété	Description
display	Type d'élément (block, inline, inline-block, none...)
visibility	Visibilité
clip	Affichage d'une partie de l'élément
overflow	Comportement en cas de dépassement
float	Flottant
clear	Arrêt d'un flottant
position	Positionnement
top, bottom, left, right	Position par rapport au haut, au bas, à la gauche, à la droite
z-index	Ordre d'affichage en cas de superposition
<b>Propriétés des listes</b>	
Propriété	Description
list-style-type	Type de liste
list-style-position	Position en retrait
list-style-image	Puce personnalisée (par une image)
list-style	Super-propriété de liste. Combine list-style-type, list-style-position, list-style-image.
<b>Propriétés des tableaux</b>	
Propriété	Description
border-collapse	Fusion des bordures
empty-cells	Affichage des cellules vides
caption-side	Position du titre du tableau

## (0A - La vidéo et l'audio

On utilise plusieurs formats audio sur le web : MP3 (le plus connu), AAC (pour Apple), OGG (format libre, connu sur Linux). Il convient de proposer les deux formats MP3 et OGG pour que tous les navigateurs puissent les lire.

On peut convertir l'audio avec Free MP3 WMA Converter :

<http://www.clubic.com/telecharger-fiche34863-free-mp3-wma-converter.html>

Pour la vidéo, il y a trois éléments :

- le format conteneur (AVI, MP4, MKV), qui contient :
- un codec audio (MP3, AAC, OGG),
- un codec vidéo (H.264, très utilisé ; Ogg Theora, libre, moins puissant ; WebM, proposé par Google, libre, concurrent sérieux de H.264).

Pour la vidéo également il convient de proposer plusieurs formats.

On peut faire les conversion avec le logiciel libre Miro Video Converter :

<http://www.mirovideoconverter.com/>

### 0A.1. Insérer de l'audio

Pour insérer un fichier audio, avec lecteur :

```
<audio src="music/morceau.mp3" controls>Le lecteur ne s'affiche pas.</audio>
```

La balise est paire afin d'afficher un message d'erreur si le lecteur ne s'affiche pas, pour les navigateurs qui ne sont pas assez récents.

Pour proposer plusieurs formats (c'est mieux !) :

```
<audio controls>
  <source src="music/morceau.mp3"></source>
  <source src="music/morceau.ogg"></source>
</audio>
```

On peut compléter la balise audio avec les attributs width (largeur), loop (jouer le morceau en boucle), autoplay (lecture au chargement de la page, à éviter), preload (préchargement du morceau à l'ouverture de la page : valeur auto par défaut, ou metadata ou none pour éviter d'user de la bande passante sur le site).

Le lecteur Dewplayer, en Flash, est une alternative intéressante (pour du MP3, avec des playlists possibles) : <http://www.alsacreations.fr/dewplayer.html>

### 0A.2. Insérer de la vidéo

Pour la vidéo, on utilise les balises <video></video>, sur le même principe :

```
<video src="video/ma_video.mp4" controls>Mettez votre navigateur à jour</video>
```

On a les mêmes attributs possibles que pour <audio>, mais avec aussi poster, qui permet de mettre une image à la place de la vidéo tant qu'elle n'est pas lancée :

```
<video src="video/ma_video.mp4" controls poster="images/une_vignette.png"></video>
```

Pour proposer plusieurs formats :

```
<video controls>
  <source src="video/ovh-data_strasbourg.mp4"></source> /* On met toujours ce format
H.264 en premier, pour les iPhone, iPod et iPad */
  <source src="video/ovh-data_strasbourg.theora.ogv"></source>
</video>
```

)

## (0B - Les métadonnées et la suite...

### 0B.1. Les métadonnées

Pour insérer des métadonnées, on utilise le langage RDFa, qui prend en compte les normes de Dublin Core (dc) et de Schema (schema).

On présente les métadonnées dans la balise <body> :

```
<body prefix="dc: http://purl.org/dc/terms/ schema: http://schema.org/">
```

L'attribut prefix permet de gérer les deux langages. On retrouve à chaque fois l'attribut property, accompagné de ce qui convient pour dc: et pour schema:

L'attribut property (référencement du contenu) :

Par exemple pour préciser la date du copyright de la page :

```
<span property="dc:dateCopyrighted schema:copyrightDate">2012</span>
```

Pour les droits (chez Dublin Core seulement sur le champ libre) :

```
<span property="dc:rights">tous droits réservés</span>
```

Pour les droits Creative Commons (par exemple) :

```
<p>Le contenu de ce site est sous licence <a property="http://creativecommons.org/ns#license" href="http://creativecommons.org/licenses/by/3.0/">Creative Commons</a>.</p>
```

Pour l'auteur de la page web :

```
<span property="dc:creator schema:creator">F. Reynaud</span>
```

L'attribut itemscope (référencement du contenu)

On peut mettre en avant une oeuvre citée avec itemscope (qui démarre l'ensemble des métadonnées sur un sujet), et itemtype (qui donne le type d'oeuvre).

```
<article itemscope itemtype="http://schema.org/Book">
<span itemprop="name">Titre du livre</span>, par <span itemprop="author">Nom de l'auteur</span>, édité chez <span itemprop="publisher">Flammarion</span> le <span itemprop="datePublished" content="2006-05-04">4 mai 2006</span>
</article>
```

Toutes les possibilités de Schema sont à l'adresse

<http://schema.org/CreativeWork>

### 0B.2. Une page validée par le W3C

Sur le site du W3C, on peut vérifier que le code que l'on a écrit n'a pas d'erreur :  
<http://validator.w3.org/>

### 0B.3. Adapter la mise en page (media queries)

Les Media Queries permettent d'adapter la mise en page du site selon la résolution d'écran de chaque navigateur, chez les usagers.

<http://www.siteduzero.com/tutoriel-3-685725-mise-en-page-adaptive-avec-les-media-queries.html>

### 0B.4. Dessiner dans les pages web (canvas)

Les canvas permettent de dessiner, jusqu'à réaliser de véritables jeux directement dans les pages web

<http://www.siteduzero.com/tutoriel-3-533489-la-balise-canvas-avec-javascript.html>

### 0B.5. Javascript

Le Javascript est un langage qui permet d'apporter de l'interactivité au site web.

### 0B.6. PHP

Le langage PHP (Personal Home Page) est un langage dynamique, utile pour la gestion des bases de données MySQL, pour une interactivité importante, pour la gestion des formulaires, des forums, des newsletters, des compteurs de visite, des jeux en ligne et réseaux sociaux.

<http://www.siteduzero.com/tutoriel-3-14668-concevez-votre-site-web-avec-php-et-mysql.html>

On associe au PHP d'autres langages dynamiques et outils : C, C++, Java, XML...

)