



CSS

TABLE DES MATIÈRES

1	- INTRODUCTION.....	4
1.1	- Historique et fonctionnalités.....	5
2	- PREMIERS PAS.....	7
2.1	- Syntaxe de base.....	8
2.2	- Fichier externe.....	10
2.3	- Les marges.....	11
2.4	- Les éléments stylés.....	13
2.5	- Les classes.....	16
2.5.1	- Objectif et syntaxe.....	16
2.5.2	- Cumul balise comme sélecteur et classe.....	19
2.5.3	- Cumul de classes.....	20
2.6	- Les identificateurs.....	21
2.6.1	- Objectif et syntaxe.....	21
2.6.2	- Exemple.....	22
2.7	- Pseudo-classes et pseudo-éléments.....	25
2.7.1	- Pseudo-classes.....	25
2.7.2	- La pseudo classe hover.....	26
2.8	- Les sélecteurs d'attributs.....	27
2.9	- Regroupements.....	28
2.9.1	- Regroupement de sélecteurs.....	28
2.9.2	- Regroupement de propriétés.....	28
2.10	- Hiérarchie.....	29
3	- LA COULEUR.....	31
3.1	- Notations pour la couleur.....	32
3.1.1	- Notation symbolique.....	32
3.1.2	- Notation Hexadécimale.....	32
3.1.3	- Notation RGB.....	32
3.1.4	- Quelques couleurs.....	33
4	- LA TYPOGRAPHIE.....	34
4.1	- La police.....	36
4.2	- Taille de la police.....	38
4.3	- Styles.....	40
4.3.1	- Gras.....	40
4.3.2	- Italiques.....	40
4.3.3	- Soulignement, surlignement.....	40
4.3.4	- La casse.....	40
4.3.5	- Exemples.....	41
4.4	- Effets.....	42
4.4.1	- La justification.....	42
4.4.2	- L'interlignage.....	42
4.4.3	- Le crénage (interlettrage).....	42
4.4.4	- L'espace inter-mots.....	42
4.4.5	- Exemples.....	43
5	- LES BORDURES & CO.....	44
5.1	- Style de bordure.....	45
5.2	- Epaisseur ou largeur.....	45
5.3	- Couleur.....	46
5.4	- Raccourci.....	46
5.5	- Exemples.....	47
5.6	- Les styles de bordures.....	49
6	- IMAGES DE FOND, DÉFILEMENT.....	51
7	- DIMENSIONNEMENT, POSITIONNEMENT.....	52
7.1	- Généralités.....	53
7.2	- Les éléments Block et Inline.....	54
7.3	- Le dimensionnement.....	55
7.3.1	- Le système HTML de boîtes.....	55

7.3.2	- Dimensionnement par défaut.....	57
7.3.3	- Éléments dimensionnés.....	59
7.3.4	- Les marges.....	61
7.4	- Le positionnement.....	65
7.4.1	- Généralités.....	65
7.4.2	- Point de départ : le flux courant.....	66
7.4.3	- Le positionnement avec les marges.....	67
7.4.4	- Positionnement absolu.....	70
7.4.5	- Positionnement relatif.....	74
7.4.6	- Positionnement flottant.....	75
7.4.6.1	- Flux courant.....	75
7.4.6.2	- Flux flottant.....	76
7.4.6.3	- Lettrine.....	76
7.4.6.4	- Encadre.....	77
7.4.7	Remarques sur le float.....	78
7.4.8	Le positionnement fixe.....	80
7.4.9	La profondeur.....	82
7.5	- Centrer les éléments.....	83
7.5.1	- Les éléments block.....	83
7.5.1.1	- Centrage horizontal.....	83
7.5.1.2	- Centrage vertical.....	85
7.5.2	- Les éléments Inline.....	86
7.5.2.1	- Centrage horizontal.....	86
7.5.2.2	- Centrage vertical.....	87
7.6	- Les <table> et CSS.....	89
8	- AFFICHER ET MASQUER.....	90
9	- ANNEXES.....	91
9.1	- Les raccourcis.....	92
9.2	- Link.....	93
9.3	- Un modèle de page avec le footer fixe.....	94
9.4	- Menus.....	96
9.4.1	- Menu vertical.....	96
9.4.2	- Menu horizontal.....	97
9.5	- Centrer	98
9.5.1	- Centrer un texte horizontalement.....	98
9.5.2	- Centrer un texte verticalement.....	99
9.5.3	- Centrer une photo horizontalement.....	100
9.6	- Un formulaire.....	102
9.6.1	- Un formulaire <inline-block>.....	102
9.6.2	- Un formulaire	103

1 - INTRODUCTION

1.1 - HISTORIQUE ET FONCTIONNALITÉS

CSS (Cascading Style Sheet) est traduit par **Feuille de Styles en Cascade**.
Les feuilles de styles datent officiellement de 1996.

Les CSS servent à mettre en forme (caractères, couleurs, marges, positionnement, dimensionnement, ...) les documents HTML.

Les CSS permettent la mise en place du principe de séparation du **contenu** et de la **forme**.

Le contenu, géré par HTML, c'est-à-dire l'information structurée logiquement accessible à tout client, même non graphique, ne doit contenir aucune balise de mise en forme (i, s, g, font, center, ...).

Les feuilles de style qui sont des ressources externes, des fichiers, présentent l'avantage de donner un style à tout un site d'où une maintenance plus aisée.

Elles permettent aussi un chargement plus rapide pour les deux motifs suivants :

- ✓ Une feuille de styles chargée à partir d'un fichier externe sera conservée dans le cache du navigateur,
- ✓ Elles diminuent la taille des documents HTML.

Définition officielle (W3C)

CSS2 : feuilles de style en cascade, niveau 2.

CSS2 est un langage de feuille de style qui permet aux auteurs et aux lecteurs de lier du style (par exemple les polices de caractères, l'espacement et un signal auditif) aux documents structurés (par exemple les documents HTML et les applications XML). En séparant la présentation du style du contenu des documents, CSS2 simplifie l'édition pour le Web et la maintenance d'un site.

CSS2 est construit sur CSS1, ainsi toute feuille de style valide en CSS1 est également valide en CSS2 à très peu d'exceptions près. CSS2 prévoit des feuilles de style liées à un média spécifique ce qui autorise les auteurs à présenter des documents sur mesure pour les navigateurs visuels, les synthétiseurs de parole, les imprimantes, les lecteurs en Braille, les appareils portatifs, etc. Cette spécification introduit aussi les notions de positionnement du contenu, de téléchargement des polices, de mise en forme des tables, de fonctions d'internationalisation, de compteurs et numérotage automatiques et quelques propriétés concernant l'interface utilisateur.

Le terme de cascade (Wikipédia)

Un autre aspect de CSS s'avère alors déterminant face aux alternatives existantes : CSS est le premier format à inclure l'idée de « cascade » (feuille de style en cascade), c'est à dire la possibilité pour le style d'un document d'être **hérité** à partir de plus d'une « feuille de style ». Cela permet d'arbitrer entre plusieurs sources concurrentes de mise en forme d'un élément, et répond donc à la nécessité de composer entre les préférences stylistiques des auteurs et des utilisateurs.

2 - PREMIERS PAS

Le style peut être défini :

- ✓ Dans une balise HTML (Règles locales, **FORTEMENT DECONSEILLE**, sauf pour "deshériter" localement),
- ✓ Dans un paragraphe situé dans une page HTML (Règles de document),
- ✓ Dans un fichier d'extension .css (Règles de site, **HAUTEMENT RECOMMANDE**).

2.1 - SYNTAXE DE BASE

Balise Style

Les balises `<style type="text/css">` et `</style>` définissent un bloc de définitions.

Sélecteur

La définition d'un style (hors balise) suit la syntaxe de base suivante :
Sélecteur, accolade ouvrante, série de paires *propriété:valeur* séparées par un point virgule, accolade fermante.

```
Sélecteur{ propriété:valeur; [propriété2:valeur2; ...] }
```

Un sélecteur peut être :

- ✓ *, html, body pour toutes les balises (sélecteur universel),
- ✓ Une balise ie un type d'éléments HTML (p, h2,... par exemple),
- ✓ Une classe,
- ✓ Un élément HTML de document identifié (id=""),
- ✓ Une pseudo-classe,
- ✓ Un sélecteur d'attribut, ...

Les commentaires

```
/* Commentaires */
```


Exemple (dans une page HTML)

```
<style type="text/css">

    /* Exemples de base */

    /* Exemples d'éléments */
    body{ background-color:beige; color:maroon; }
    h1{ color:darkgreen; }

    /* Exemple de classe */
    .surligne{ background-color:yellow; color:black; }

    /* Exemples d'identifiant */
    #pied{ background-color:silver; color:white; }

</style>
```

background-color pour la couleur du fond.
color pour la couleur du texte.

2.2 - FICHIER EXTERNE

Une feuille de style en cascade sera située dans un fichier indépendant d'extension .css (facultatif mais canonique).

Dans le fichier externe css **les balises <style></style> ne doivent pas être présentes.**

Dans le document HTML vous y faites référence, **dans le paragraphe <head>**, avec la balise <link> dont la syntaxe est la suivante :

```
<link [type="text/css"] rel="stylesheet" href="chemin/fichier.css" />
```

Exemple

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

Vous pouvez alternativement utiliser la syntaxe d'import

```
<style type="text/css">
    @import url(chemin/fichier.css) ;
</style>
```

Exemple

```
<style type="text/css">
    @import url(ma_feuille_de_style.css) ;
</style>
```

NB : notez l'absence d'apostrophes (') ou de guillemets (") autour de l'URL. Mettre des " ou des ' est facultatif.

Remarques :

<link> est une balise html. Elle permet des liens avec des fichiers externes, entre autres, des feuilles de styles. Pour les feuilles de style les valeurs des propriétés ne sont que celles qui sont présentées plus haut plus quelques autres :

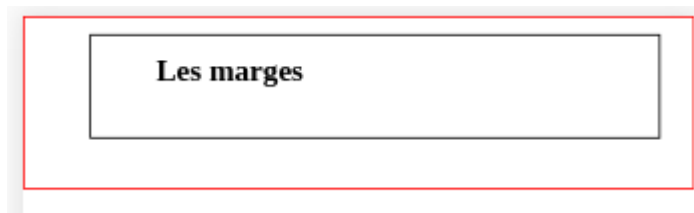
- ✓ type="text/css"
- ✓ rel="stylesheet" ou rel="alternate stylesheet"
- ✓ href="chemin/fichier.css"
- ✓ title="valeur"
- ✓ id="identifiant"

Nous verrons plus loin la propriété MEDIA.

Import est une règle CSS.

La différence entre le link et l'import est que vous pouvez inclure des feuilles de styles dans d'autres feuilles de style avec import.

2.3 - LES MARGES



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
    <link rel="stylesheet" type="text/css" href="marges.css">
    <title>marges.html</title>
  </head>

  <body>
    <div>
      <h3>
        Les marges
      </h3>
    </div>
  </body>
</html>
```

margin pour les marges externes

padding pour les marges internes

Pour ces attributs il est possible de définir les marges une à une :

```
*{margin:0; padding:0;}
div{
  border: 1px solid red;
}
h3{
  border: 1px solid black;
  margin-top: 10px;
  margin-right: 20px;
  margin-bottom: 30px;
  margin-left: 40px;
  padding-top: 10px;
  padding-right: 20px;
  padding-bottom: 30px;
  padding-left: 40px;
}
```

ou en mode raccourci :

dans le sens des aiguilles d'une montre (top, right, bottom, left)

```
margin: 10px 20px 30px 40px;
```

ou encore en mode « condensé » :

verticales (top, bottom) puis horizontales (right, left)

```
margin: 0 30px;
```

2.4 - LES ÉLÉMENTS STYLÉES

Objectif

Définir le style des éléments HTML du document ou du site.

Dans le paragraphe <style> ou dans le fichier externe vous "stylez" les éléments HTML.

Le sélecteur est donc un élément.

Les éléments que nous allons styler seront les éléments body, h1, h2, p et a.

Exemple

Sans style



Le code HTML sans style

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Les sélecteurs balises</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>

<body>

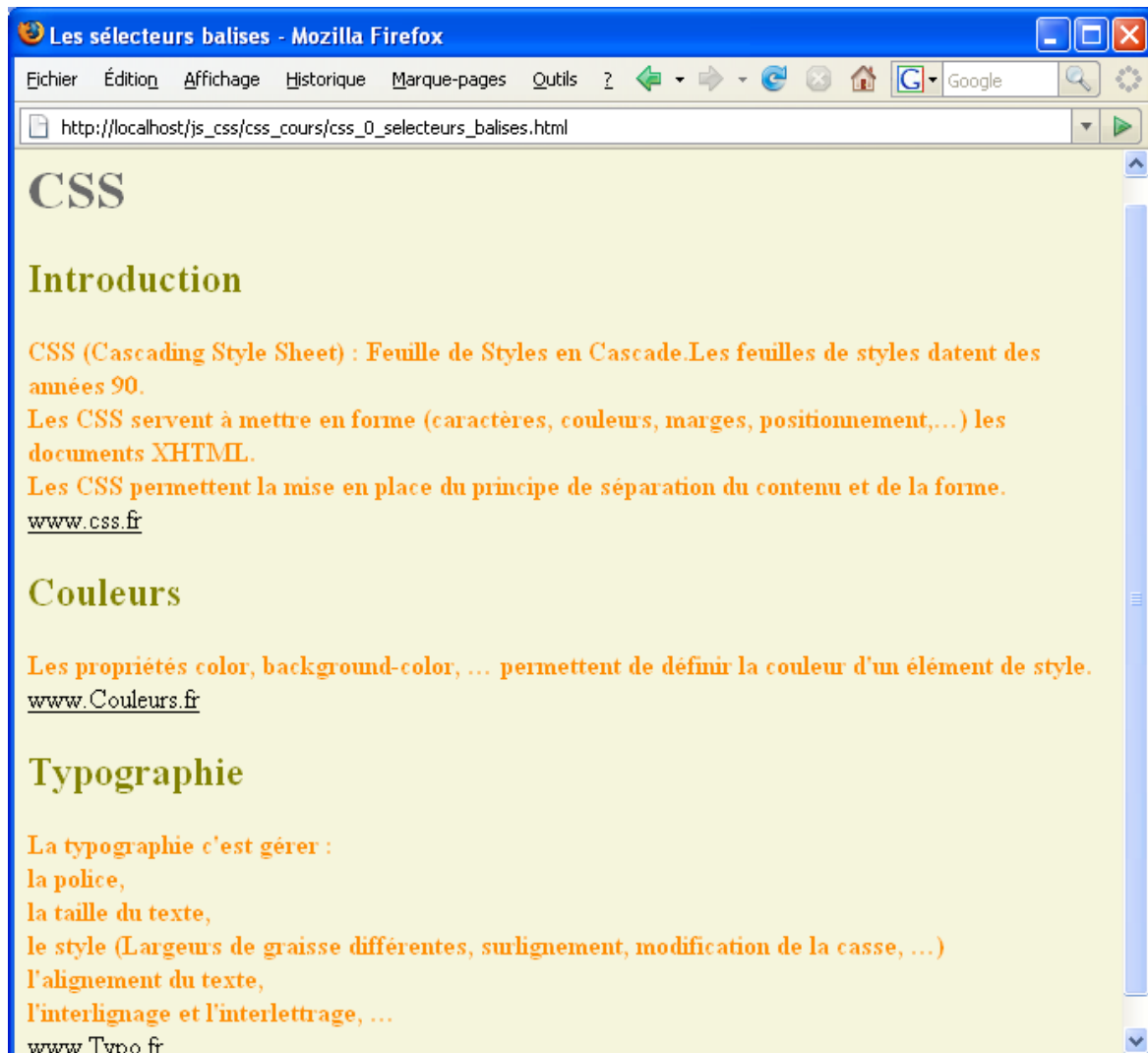
<h1>CSS</h1>

<h2>Introduction</h2>
  <p>
    CSS (Cascading Style Sheet) : Feuille de Styles en Cascade. Les
    feuilles de styles datent des années 90.<br />
    Les CSS servent à mettre en forme (caractères, couleurs,
    marges, positionnement,...) les documents XHTML.<br />
    Les CSS permettent la mise en place du principe de séparation
    du contenu et de la forme.<br />
    <a href="#">www.css.fr</a>
  </p>

<h2>Couleurs</h2>
  <p>
    Les propriétés color, background-color, ... permettent de définir
    la couleur d'un élément de style.<br />
    <a href="#">www.Couleurs.fr</a>
  </p>

<h2>Typographie</h2>
  <p>
    La typographie c'est gérer :<br />
    la police,<br />
    la taille du texte,<br />
    le style (Largeurs de graisse différentes, surlignement,
    modification de la casse, ...)<br />
    l'alignement du texte,<br />
    l'interlignage et l'interlettrage, ...<br />
    <a href="#">www.Typo.fr</a>
  </p>
</body>
</html>
```

Avec le style



Le code CSS

```
<style type="text/css">
  body{background-color:beige; color:white;}
  h1{color:dimgray;}
  h2{color:olive;}
  p{color:darkorange; font-weight:bold;}
  a{color:black; font-weight:normal;}
</style>
```

2.5 - LES CLASSES

2.5.1 - Objectif et syntaxe

Objectif

Les exemples précédents montrent des sélecteurs qui sont des éléments HTML (body, h1,...). Il est possible de créer des sélecteurs qui ne sont pas des éléments.

Ce cas permet de créer des classes qui seront disponibles pour un ou plusieurs types d'éléments.

Une classe est un nom valide précédé d'un point.

Ce cas est utile lorsque vous avez plusieurs éléments de types différents qui sont de même style ou comportent des éléments de style en commun (par exemple les titres de différents niveaux de même couleur ou bien des liens situés dans des listes).

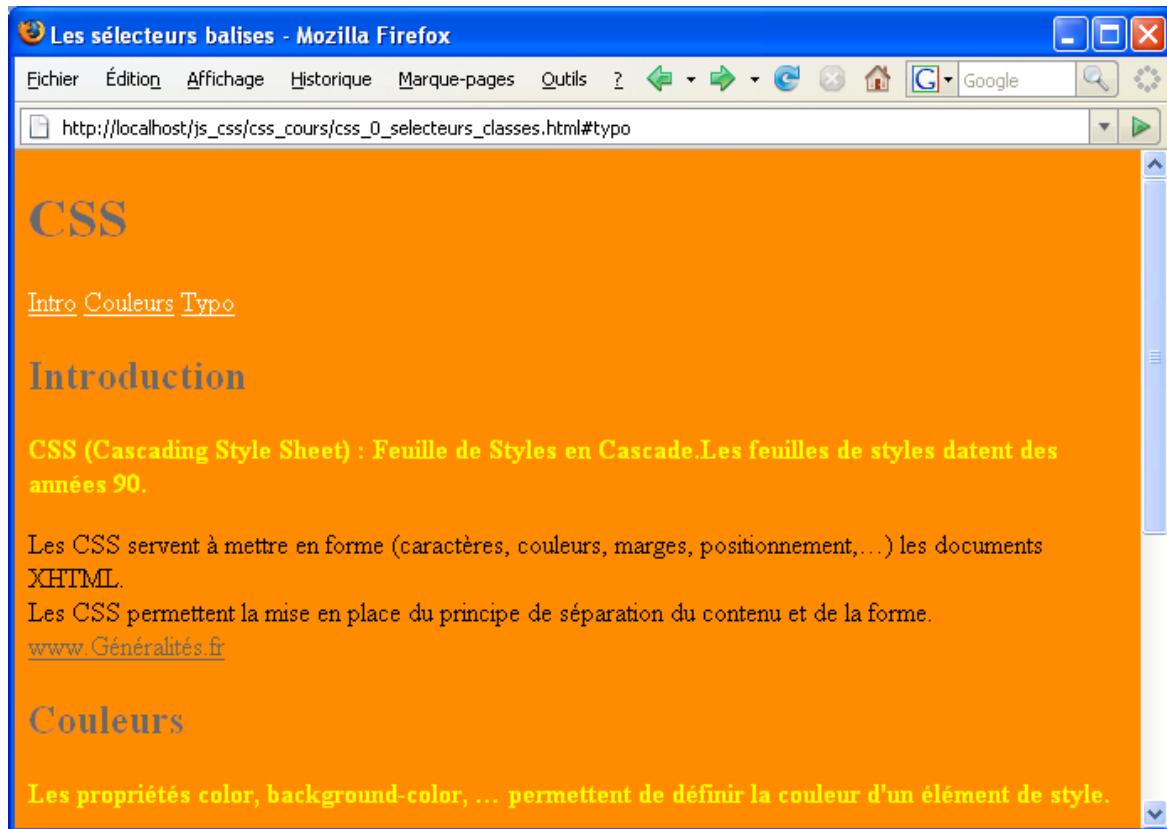
Ceci permet une réutilisation.

Syntaxe

```
.nom_de_classe{ propriété:valeur[; propriété2:valeur2;] }
```


Exemple

Reprenons l'exemple précédent légèrement modifié : nous ajoutons des ancres internes (dans les sous-titres) et des liens (en blanc) vers ces ancres et chaque sous-titre contient plusieurs paragraphes; le premier en jaune et les autres en noir. Pour cela nous créons deux classes : `liens_internes` et `paragraphe_1`.



CSS

```
<style type="text/css">
  body{background-color:darkorange; color:white;}
  h1{color:dimgray;}
  h2{color:silver;}
  p{color:black;}
  a{color:dimgray;}

  .paragraphe_1{color:yellow; font-weight:bold;}
  .liens_internes{color:white;}
</style>
```

Code HTML

```

<body>

<h1>CSS</h1>
<a href="#intro" class="liens_internes">Intro</a>
<a href="#couleurs" class="liens_internes">Couleurs</a>
<a href="#typo" class="liens_internes">Typo</a>

<h2><a name="intro">Introduction</a></h2>
  <p class="paragraphe_1">
    CSS (Cascading Style Sheet) : Feuille de Styles en Cascade. Les
    feuilles de styles datent des années 90.
  </p>
  <p>
    Les CSS servent à mettre en forme (caractères, couleurs,
    marges, positionnement,...) les documents XHTML.<br />
    Les CSS permettent la mise en place du principe de séparation
    du contenu et de la forme.<br />
    <a href="#">www.Généralités.fr</a>
  </p>

<h2><a name="couleurs">Couleurs</a></h2>
  <p class="paragraphe_1">
    Les propriétés color, background-color, ... permettent de définir
    la couleur d'un élément de style.
  </p>
  <p>
    La propriété color définit la couleur du texte<br />
    <a href="#">www.Couleurs.fr</a>
  </p>

<h2><a name="typo">Typographie</a></h2>
  <p class="paragraphe_1">
    La typographie c'est gérer :
  </p>
  <p>
    la police,<br />
    la taille du texte,<br />
    le style (Largeurs de graisse différentes, surlignement,
    modification de la casse, ...)<br />
    l'alignement du texte,<br />
    l'interlignage et l'interlettrage, ...<br />
    <a href="#">www.Typo.fr</a>
  </p>

</body>

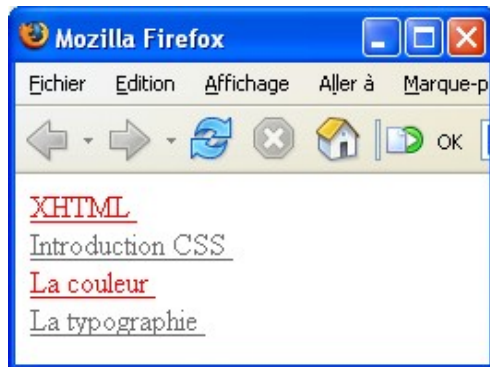
```

2.5.2 - Cumul balise comme sélecteur et classe

Objectif

L'objectif est d'avoir le même élément HTML stylé différemment en fonction de sa "position" (l'élément `<a>` dans l'exemple).

Le premier lien est rouge, le deuxième gris, le troisième rouge, etc.



Démarche

Styliser le sélecteur `a` (élément).

Styliser une classe : le sélecteur `.lien_pair`.

Code

Dans le `<head>` :

```
<style type="text/css">
  a{color:red;}
  .lien_pair{color:dimgray;}
</style>
```

et dans le `<body>` :

```
<body>
<a href="css_xhtml.html">XHTML</a><br />

<a class="lien_pair" href="css_intro.html">Introduction CSS</a><br />

<a href="css_couleur.html">La couleur</a><br />

<a class="lien_pair" href="css_typographie.html">La typographie</a>
</body>
```

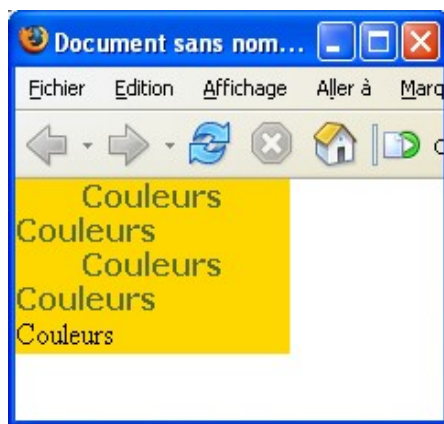
2.5.3 - Cumul de classes

Il est possible de cumuler plusieurs classes.

```
.classe1{ ... }
.classe2{ ... }
```

```
<balise class="classe1 classe2" ...>
```

Exemple



```
<style type="text/css">
  *{margin:0; padding:0;}
  ul{list-style-type:none; width:150px; background-color:gold;}

  .olive_foncee{color:darkolivegreen; font-weight:bold; font-
family:Verdana;}
  .centrage{text-align:center;}
</style>
```

```
<ul>
  <li class="olive_foncee centrage">Couleurs</li>
  <li class="olive_foncee">Couleurs</li>
  <li class="olive_foncee centrage">Couleurs</li>
  <li class="olive_foncee">Couleurs</li>
  <li>Couleurs</li>
</ul>
```

2.6 - LES IDENTIFICATEURS

2.6.1 - Objectif et syntaxe

Un identificateur (id) permet d'identifier un élément HTML dans un document c'est-à-dire de le rendre unique.

Une classe est une généralisation alors qu'une identification est une spécialisation.

Un sélecteur CSS peut s'y référer en le préfixant avec #.

```
#id{ attribut: valeur; }
```

```
<div id="encart_publicitaire"> ... </div>
```

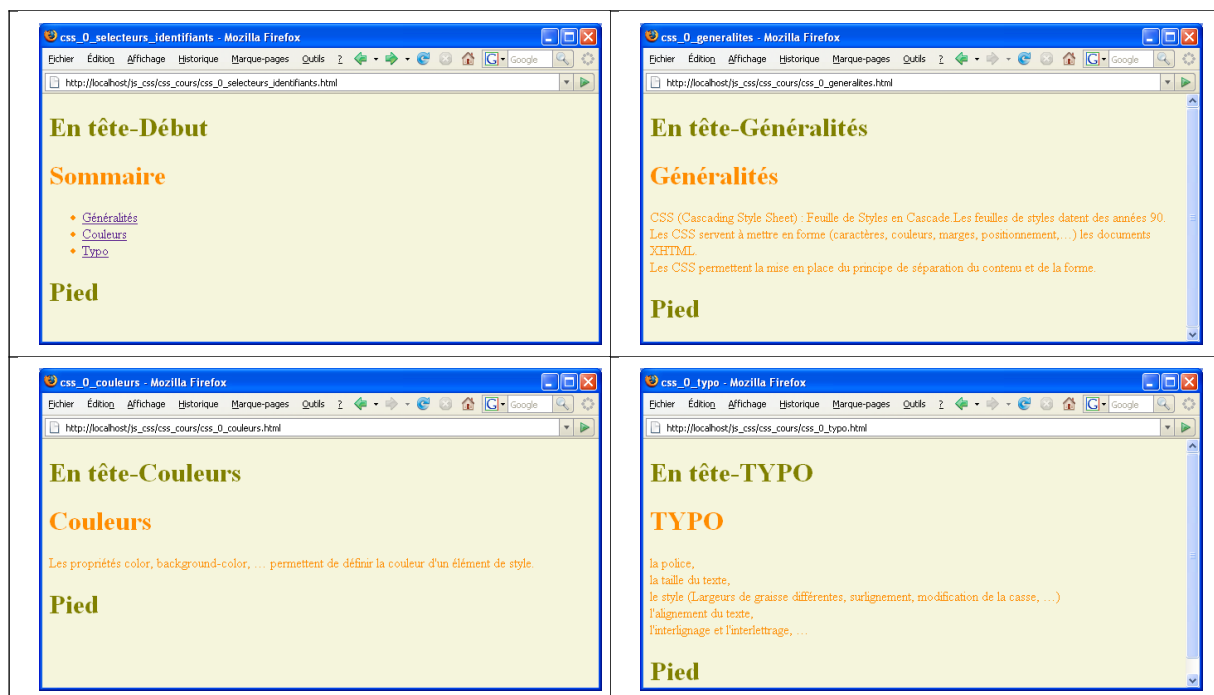
```
#encart_publicitaire{ background-color:yellow; }
```

2.6.2 - Exemple

Quatre documents avec des identifiants et des liens css vers :

- ✓ Le sommaire
- ✓ Généralités
- ✓ Couleurs
- ✓ Typo

Il faut changer les règles dans `css_0.css` pour voir les effets des liens et des #.



Le fichier CSS : css_0.css

```
/* CSS Document */
body{background-color:beige;}

#en_tete{color:black;}
#corps{color: darkorange;}
#pied{color:olive;}
```

Les documents HTML : ils ont tous le même <head> (sauf le titre)

css_0_sommaire.html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>css_0_sommaire.html</title>
<link href="css_0.css" rel="stylesheet" type="text/css" />
</head>

<body>

<div id="en_tete"><h1>En tête-Début</h1></div>
<div id="corps">
  <h1>Sommaire</h1>
  <ul>
    <li><a href="css_0_generalites.html">Généralités</a></li>
    <li><a href="css_0_couleurs.html">Couleurs</a></li>
    <li><a href="css_0_typo.html">Typo</a></li>
  </ul>
</div>
<div id="pied"><h1>Pied</h1></div>

</body>
</html>
```

css_0_generalites.html

```
<body>
<div id="en_tete"><h1>En tête-Généralités</h1></div>
<div id="corps">
  <h1>Généralités</h1>
  <p>
    CSS (Cascading Style Sheet) : Feuille de Styles en Cascade. Les
    feuilles de styles datent des années 90.<br />
    Les CSS servent à mettre en forme (caractères, couleurs,
    marges, positionnement,...) les documents XHTML.<br />
    Les CSS permettent la mise en place du principe de séparation
    du contenu et de la forme.<br />
  </p>
</div>
<div id="pied"><h1>Pied</h1></div>
</body>
```

css_0_couleurs.html

```
<body>
<div id="en_tete"><h1>En tête-Couleurs</h1></div>
<div id="corps">
  <h1>Couleurs</h1>
  <p>
    Les propriétés color, background-color, ... permettent de définir
    la couleur d'un élément de style.<br />
  </p>
</div>
<div id="pied"><h1>Pied</h1></div>
</body>
```

css_0_typo.html

```
<body>
<div id="en_tete">
  <h1>En tête-TYPO</h1>
</div>
<div id="corps">
  <h1>TYPO</h1>
  <p>
    la police,<br />
    la taille du texte,<br />
    le style (Largeurs de graisse différentes, surlignement,
    modification de la casse, ...)<br />
    l'alignement du texte,<br />
    l'interlignage et l'interlettrage, ...<br />
  </p>
</div>
<div id="pied"><h1>Pied</h1></div>
</body>
```


2.7 - PSEUDO-CLASSES ET PSEUDO-ÉLÉMENTS

2.7.1 - Pseudo-classes

Les pseudo-classes sont, principalement, des éléments correspondant à des actions.
Les pseudo-classes CSS sont les suivantes :

- ✓ :first-child (Premier enfant d'un élément; c'est une classe; cf paragraphe suivant).
- ✓ :link (Lien)
- ✓ :visited (Lien déjà visité)
- ✓ **:hover (Lien survolé)**
- ✓ :active (Lien actif donc cliqué)
- ✓ :focus (Élément prenant le focus)
- ✓ :lang (Langue)

Par exemple la pseudo-classe **a:hover** permet de gérer le survol d'un lien avec la souris.
(Pour la balise <a> les autres possibilités sont a:link, a:visited, a:active).

NB : jusqu'à la version 6 de IE cette action n'était valide que pour la balise <a> contrairement aux recommandations du W3C.
~~A partir de IE 7; il est possible de l'appliquer à d'autres balises(div, p, ...) comme cela est le cas sous Firefox.~~

```
a:hover { text-decoration:none; }
```

```
<a href="#">La couleur</a>  
<a href="#">Les polices</a>  
<a href="#">Les bordures</a>
```

A pour effet d'afficher sans décoration tout lien survolé.

Il est possible de combiner classe et pseudo-classe.

```
a.lien_noir:hover{ text-decoration:none; color:black; }
```

```
<a href="#">La couleur</a>  
<a class="lien_noir" href="#">Les polices</a>  
<a href="#">Les bordures</a>
```

A pour effet d'afficher sans décoration tout lien de la classe *lien_noir* survolé et de l'afficher en noir.

Tous les autres liens seront affichés en fonction des attributs de la pseudo-classe a:hover si elle est définie.

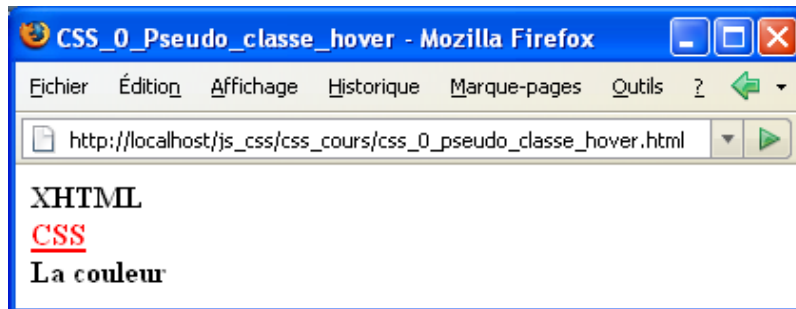
Note : il existe aussi a:active, a:visited, ...

2.7.2 - La pseudo classe hover

Objectif

Modifier la mise en forme du lien lorsque que la souris le survole; de noir il passe à rouge souligné.

Ecran



Code

```
<html>
<head>
<title>CSS_0_Pseudo_classe_hover</title>
<style type="text/css">
    a{color:black; text-decoration:none; font-weight:bold;}
    a:hover{color:red; text-decoration:underline;}
</style>
</head>

<body>
  <a href="#">XHTML</a><br />
  <a href="#">CSS</a><br />
  <a href="#">La couleur</a><br />
</body>
</html>
```

2.8 - LES SÉLECTEURS D'ATTRIBUTS

- **Objectif et syntaxe**

Styler des éléments en fonction de la valeur d'un attribut d'élément.

Balise[attribut="valeur"] { propriété:valeur[;propriété2:valeur2;]}

- **Ecran**

Les zones de saisie sont sur fond noir; le mot de passe sur fond rouge.



- **Codes**

```
<style type="text/css">
    input[type="text"]{ background-color:black; color:white;
position:absolute; left:100px; }
    input[type="password"]{ background-color:red; color:white;
position:absolute; left:100px; }
</style>
```

```
<body>
<form action="" method="get">
    <label>Nom</label><input name="nom" type="text" value="Haddock" /><br/>
    <label>Prénom</label><input name="prenom" type="text"
value="Archibald" /><br/>
    <label>E-mail</label><input name="email" type="text"
value="ha@free.fr" /><br/>
    <label>Mot de passe</label><input name="mdp" type="password" /><br/>
</form>
</body>
```

Exercice: complétez le formulaire.

2.9 - REGROUPEMENTS

2.9.1 - Regroupement de sélecteurs

Si plusieurs sélecteurs ont les mêmes caractéristiques il est possible de les regrouper.

```
h1{ margin-left:0; }  
p{ margin-left:0; }  
.texte{ margin-left:0; }
```

devient

```
h1, p, .texte{ margin-left:0; }
```

Si tous les sélecteurs ont quelques propriétés communes il est possible d'utiliser le sélecteur universel *.

```
*{ margin:0; padding:0; }
```

2.9.2 - Regroupement de propriétés

Il est possible d'utiliser des "raccourcis" pour qualifier une classe ou un sélecteur eu égard à certaines propriétés génériques (font, border, background, ...)

```
selecteur{ font: italic bold 120% Tahoma; }
```

Correspond à

```
selecteur  
{  
font-style: italic;  
font-weight: bold;  
font-size: 120%;  
font-family: tahoma;  
}
```

/ raccourci border: width style color */*

2.10 - Hiérarchie

• Présentation

CSS permet d'utiliser des identifications hiérarchiques en utilisant l'arborescence des blocs d'un document.

Les syntaxes sont les suivantes :

Balise.classe{ propriétés }

```
a.vert{color:green;} /* Liens de la classe .vert */
```

Balise_parent balise_enfant{ propriétés }

```
div a {color:purple;} /* Liens dans une Div */
```

```
div ul li a {color:red;} /* Liens d'item d'une liste d'une Div */
```

#identifiant balise{ propriétés }

```
#menu li a {color:gold;} /* Lien d'un élément ul identifié par id="menu" */
```

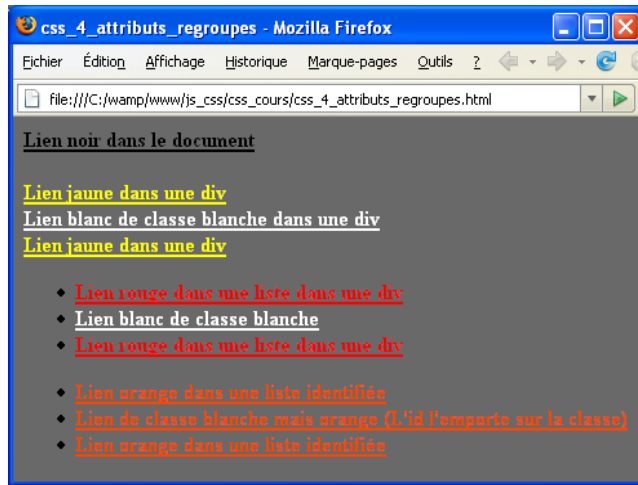
Mixtes

```
/* liens de la classe .lien_pair dans les listes */  
ul li a.lien_pair{color: dimgray;}
```

```
/* liens de classe .lien_pair dans un item de liste identifié par menu */  
#menu li a.lien_pair {color:green;}
```

Elles seront utilisées sporadiquement tout au long de ce support.
Cf aussi le paragraphe sur l'héritage.

• Exemple



```
<head>
<style type="text/css">
  body{background-color:dimgray;}
  a{color:black; font-weight:bold;} /* Liens */
  div a{color:yellow;} /* Liens dans une Div */
  a.blanche{color:white;} /* Liens de la classe blanche */
  div ul li a{color:red;} /* Liens dans un item de liste d'une Div */
  #menu_h ul a{color:orangered;}
</style>
</head>
```

```
<body>
<a href="#">Lien noir dans le document</a><br /><br />
<div>
  <a href="#">Lien jaune dans une div</a><br />
  <a class="blanche" href="#">Lien blanc de classe blanche dans une
div</a><br />
  <a href="#">Lien jaune dans une div</a><br />
</div>
<div>
  <ul>
    <li><a href="#">Lien rouge dans une liste dans une div</a></li>
    <li><a class="blanche" href="#">Lien blanc de classe blanche</
a></li>
    <li><a href="#">Lien rouge dans une liste dans une div</a></li>
  </ul>
</div>
<div>
  <ul id="menu_h">
    <li><a href="#">Lien orange dans une liste identifiée</a></li>
    <li><a class="blanche" href="#">Lien de classe blanche mais
orange (L'id l'emporte sur la classe)</a></li>
    <li><a href="#">Lien orange dans une liste identifiée</a></li>
  </ul>
</div>
</body>
```

3 - LA COULEUR

3.1 - NOTATIONS POUR LA COULEUR

Les propriétés **color** (couleur du texte), **background-color** (couleur du fond), **border-color** (couleur d'une bordure) permettent de définir la couleur d'un élément de style.

3.1.1 - Notation symbolique

Certaines couleurs sont nommées : white, black, red, green, blue, ... C'est la notation la plus lisible.

```
body{ background-color:gray; }
```

3.1.2 - Notation Hexadécimale

Elles sont précédées du #. Elles comportent 6 chiffres hexadécimaux (de 0 à F). C'est la plus courante.

Le noir #000000 et le blanc #FFFFFF.

```
body{ background-color:#FF00AA ; }
```

Note : il existe une notation courte sur 3 chiffres (#FFF, #000, ...).

3.1.3 - Notation RGB

RGB(Red Green Blue ou Rouge Vert Bleu) en valeur absolue de 0 à 255 ou en pourcentage de 0 à 100 pour chaque couleur.

```
body{ background-color:rgb(0, 0, 255); }
```

```
#contenu{ color:rgb(50%,10%,50%); } /* violet */
```


3.1.4 - Quelques couleurs

Couleur	Mot-clé	RGB	Hexa	Hexa court
Noir	Black	0,0,0	#000000	#000
Rouge	Red	255,0,0	#FF0000	#F00
Gris	Gray	128,128,128	#808080	
Olive	Olive	128,128,0	#808000	
Vert	Green	0,255,0	#00FF00	#0F0
Marron	Maroon	128,0,0	#800000	
Bleu	Blue	0,0,255	#0000FF	#00F
Jaune	Yellow	255,255,0	#FFFF00	#FF0
Blanc	White	255,255,255	#FFFFFF	#FFF

- **Exemple**

Le drapeau tricolore

```
<style type="text/css">
  /* Le texte en rouge et le fond en bleu */
  body{ background-color:blue; color:red; }
  /* Les liens en blanc */
  a{ color:white; }
</style>
```



4 - LA TYPOGRAPHIE

La typographie gère :

- ✓ la police ou fonte,
- ✓ la taille du texte,
- ✓ le style (Largeurs de graisse différentes, surlignement, modification de la casse, ...)
- ✓ l'alignement du texte,
- ✓ l'interlignage et l'interlettrage, ...

NB : les polices *à chasse fixe ou monospace*, qui sont dites aussi non proportionnelles; sont celles où chaque caractère occupe la même largeur.

Par exemple la `Courier New` est fixe

0123456789-ABCDEFGHIJKLMNOPQRSTUVWXYZ-abcdefghijklmnopqrstuvwxyz

alors que la `Comic sans MS` ou l'`Arial` sont proportionnelles.

0123456789-ABCDEFGHIJKLMNOPQRSTUVWXYZ-abcdefghijklmnopqrstuvwxyz

NB : les polices exotiques.

Pour être certain d'afficher des polices exotiques il existe plusieurs solutions :

- ✓ Insérer une image à la place du texte,
- ✓ Insérer un PDF à la place du texte.

Mais ces deux techniques comportent des coûts de chargement et de maintenance exorbitants et dérogent au principe de séparation du contenu et de la présentation.

4.2 - TAILLE DE LA POLICE

La taille est précisée de façon absolue ou relative.
Elle est définie par la propriété **font-size**.

- **Les unités de taille fixe**

pt	Point anglais (1/72 de pouce soit 0,352778 mm) Alors que le point européen (le didot) vaut 0,376 mm
pc	Pica (12 points soit 4,233mm)
cm	Centimètre
mm	Millimètre
in	Pouce (25,4mm)
px	pixel

- **Les tailles relatives**

%	Pourcentage
em	Largeur de M dans la police de référence
ex	Hauteur de x dans la police de référence

- **Les mots-clés**

Ces tailles sont fixées par le navigateur.

xx-small	
x-small	
Small	Les bons choix
Medium	Les bons choix
Large	
x-large	
xx-large	

- Exemples



```
<style type="text/css">
.taillemini{ font-size: xx-small ; }
.taillemaxi{ font-size: xx-large ; }
.taille10{font-size: 10pt ; }
.taille20{font-size: 20px ; }
.poidsplume{ font-weight: lighter ; }
.poids lourd{ font-weight: bolder ; }
</style>
```

```
<body>
<span class="taillemini">taillemini</span>
<span class="taillemaxi">taillemaxi</span>
<span class="taille10">taille10</span>
<span class="taille20">taille20</span>
<span class="poidsplume">poidsplume</span>
<span class="poids lourd">poids lourd</span>
</body>
```

4.3 - STYLES

4.3.1 - Graise

Elle est définie par la propriété **font-weight**.

Les valeurs sont : **normal**, **bold**, **lighter**, **bolder**, 100, 200, ..., 900.

4.3.2 - Italiques

Elle est définie par la propriété **font-style**.

Les valeurs sont : **normal**, **italic**.

4.3.3 - Soulignement, surlignement,...

Ils sont gérés par la propriété **text-decoration**.

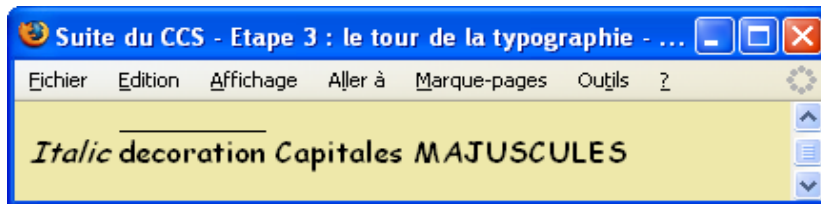
Les valeurs sont : **underline** (souligné), **overline** (surligné), **line-through** (barré), **blink** (clignotant), **none** (aucune décoration).

4.3.4 - La casse

Elle est gérée par la propriété **text-transform**.

Les valeurs sont **capitalize** (première lettre en majuscule), **lowercase** (minuscules), **uppercase** (majuscules), **none** (sans modification).

4.3.5 - Exemples



```
<style type="text/css">
.styleitalique{ color:black ; font-style:italic ; }

.decoration{ color:black ; text-decoration:underline ; }

.capitales { color:black ; text-transform:capitalize ; }
.majuscules { color:black ; text-transform:uppercase ; }
</style>
```

```
<body>
<span class="styleitalique">Italic</span>
<span class="decoration">decoration</span>

<span class="capitales">capitales</span>
<span class="majuscules">majuscules</span>
</body>
```

4.4 - EFFETS

4.4.1 - La justification

Elle est gérée par la propriété **text-align**.
Les valeurs sont left, right, center, justify ou **normal**.

4.4.2 - L'interlignage

Il est géré par la propriété **line-height**.
Les valeurs sont exprimées en unités de taille (pt, pc, cm, mm, in ou px, em, ex) en absolu ou en négatif.
Le texte est toujours **centré verticalement** lorsque la propriété line-height est utilisée.

4.4.3 - Le crénage (interlettrage)

Il est géré par la propriété **letter-spacing**.
Les valeurs sont exprimées en unités de taille (pt, pc, cm, mm, in ou px, em, ex) en absolu ou en négatif.

4.4.4 - L'espace inter-mots

Il est géré par la propriété **word-spacing**.
Les valeurs sont exprimées en unités de taille (pt, pc, cm, mm, in ou px, em, ex) en absolu ou en négatif.

4.4.5 - Exemples



```
<style type="text/css">
.interligne{ color: darkorange ; line-height: 3em ; }
.interlettre{ color: darkorange ; letter-spacing: 3mm ; line-height:
1em ;}
.intermot{ color: darkorange ; word-spacing: 3em ;}
.centre{ color: sienna ; text-align: center ; }
.adroite{ color: sienna ; text-align: right ; }
.justifie{ color: sienna ; text-align: justify ; }
</style>
```

```
<body>
<br /><span class="interligne">Interlignage</span>
<br /><span class="interligne">Interlignage</span>

<br /><span class="interlettre">Interlettrage</span>
<br /><span class="intermot">Inter wordage</span>

<div class="centre">centré</div>
<div class="adroite">droite</div>
<div class="justifie">Ce texte, s'il est assez long sera justifié
!!!</div>
</body>
```

5 - LES BORDURES & CO

Les bordures sont caractérisées par leur style, leur épaisseur et leur couleur.

5.1 - STYLE DE BORDURE

Il est géré par la propriété **border-style**.

Les valeurs possibles sont : **solid** (trait plein), double (deux traits), dotted (pointillés), dashed (tirets), groove (effet 3D gravé), ridge (effet 3D sortant), hidden, inset (effet entrant), outset (effet sortant).

Cf plus loin pour une visualisation des styles.

Il est possible de définir **plusieurs styles de bordure pour un élément**.

Soit côté par côté avec les propriétés border-top-style, border-right-style, border-bottom-style et border-left-style.

Soit en définissant 2, 3 ou 4 valeurs pour border-style (Ce système n'est pas reconnu par Firefox).

2 valeurs affecteront les lignes horizontales puis verticales.

3 valeurs affecteront le top, les côtés verticaux et le bottom.

4 valeurs affecteront les 4 côtés dans le sens des aiguilles d'une montre en commençant par top.

5.2 - EPAISSEUR OU LARGEUR

Elle est gérée par la propriété **border-width**.

Les valeurs possibles sont thin, medium et thick ou une valeur numérique (cf le chapitre 4 sur les unités de taille).

De la même façon que pour le style il est possible de définir plusieurs épaisseurs pour un même élément.

Soit avec les propriétés border-top-width, border-right-width, border-bottom-width et border-left-width.

Soit en définissant 2, 3 ou 4 valeurs pour border-width.

5.3 - COULEUR

Elle est gérée par la propriété **border-color**.
Les couleurs sont celles étudiées au chapitre 3.

De la même façon que pour le style il est possible de définir plusieurs couleurs pour une même bordure.

Soit avec les propriétés `border-top-color`, `border-right-color`, `border-bottom-color` et `border-left-color`.

Soit en définissant 2, 3 ou 4 valeurs pour `border-color`.

5.4 - RACCOURCI

W3C : The border properties specify the width, style, color.

`border: largeur style couleur;`

```
h1{ border: thin solid black; }
```

5.5 - EXEMPLES

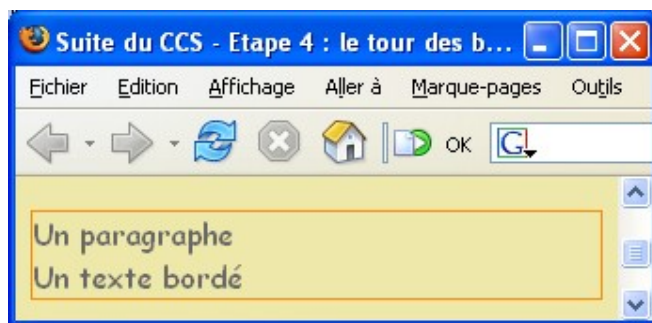
- **Simple**

```
.bords{ border-width:thin; border-style:solid; border-color:darkorange; }
```

ou

```
.borde{ border:thin solid darkorange; }
```

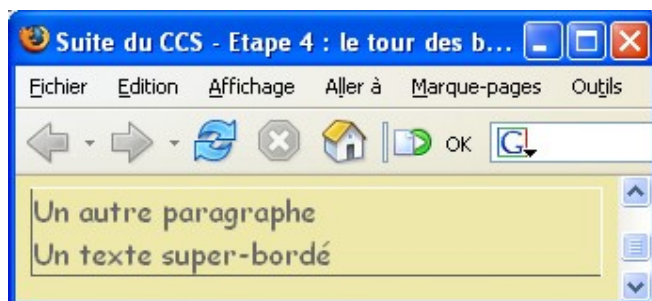
```
<p class="bords">  
    Un paragraphe<br />  
    Un texte bordé  
</p>
```



- **Elaboré**

```
.superbords{ border-width:thin; border-style:solid; border-top-color:white; border-right-color:white; }
```

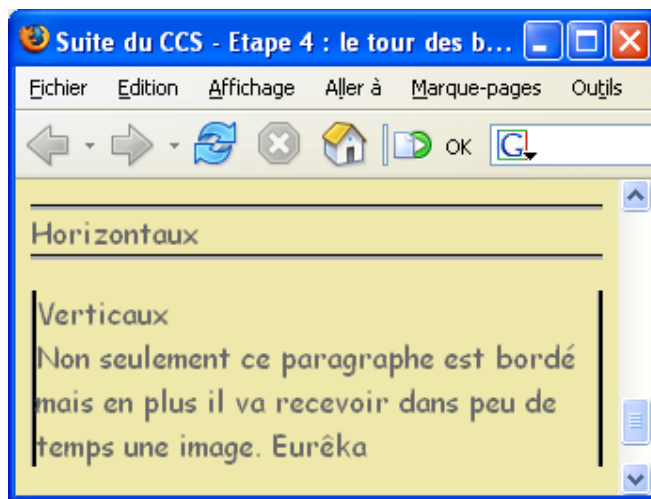
```
<p class="superbords">  
    Un autre paragraphe<br />  
    Un texte super-bordé  
</p>
```



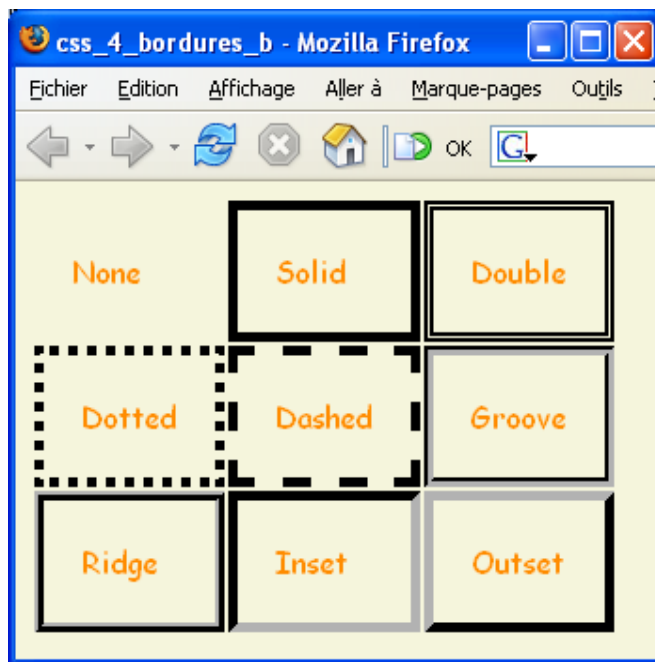
- **Partiels**

```
.horizontal{ border-width:medium; border-style:groove none; border-color:black; }  
  
.vertical{ border-width:medium; border-style:none ridge; border-color:black; }
```

```
<p class="horizontal">  
    horizontal  
</p>  
<p class="vertical">  
    vertical  
</p>
```



5.6 - LES STYLES DE BORDURES



```
<html>
  <head>
    <title>css_4_bordures_b</title>
    <style type="text/css">
      body{ background-color:beige; color:darkorange; font-
family:'Comic sans MS';}
      td{padding:20px; border-width:thick; border-color:black}
      #aucun{border-style:none;}
      #plein{border-style:solid;}
      #double{border-style:double;}
      #pointille{border-style:dotted;}
      #tiret{border-style:dashed;}
      #creux{border-style:groove;}
      #relief{border-style:ridge;}
      #incruste{border-style:inset;}
      #extrude{border-style:outset;}
    </style>
  </head>
  <body>
    <table>
      <tr>
        <td id="aucun">None</td>
        <td id="plein">Solid</td>
        <td id="double">Double</td>
      </tr>
      <tr>
        <td id="pointille">Dotted</td>
        <td id="tiret">Dashed</td>
        <td id="creux">Groove</td>
      </tr>
      <tr>
        <td id="relief">Ridge</td>
        <td id="incruste">Inset</td>
        <td id="extrude">Outset</td>
      </tr>
    </table>
  </body>
</html>
```

```
        <td id="relief">Ridge</td>
        <td id="incruste">Inset</td>
        <td id="extrude">Outset</td>
    </tr>
</table>
</body>
</html>
```

6 - IMAGES DE FOND, DÉFILEMENT

Cf CSS.odt

7 - DIMENSIONNEMENT, POSITIONNEMENT

7.1 - GÉNÉRALITÉS

- **Dimensionnement**

Le dimensionnement des éléments est réalisé au moyen des propriétés *width* et *height*.

- **Positionnement**

Les éléments HTML sont positionnés dans le flux courant.

Le positionnement de base dépend de deux paramètres :

Le type d'élément (bloc – block - ou ligne – inline -)

Les éléments blocs sont situés sur des lignes différentes, ils se succèdent verticalement, alors que les éléments "*inline*" sont situés sur la même ligne, ils se succèdent horizontalement.

Cf `css2_complements.doc` pour la liste des éléments et leur classification.

Le système de boîte HTML normalisé par le W3C.

Les boîtes, pour les éléments de type bloc, à l'exception de la balise `<div>`, ajoutent par défaut des marges externes (*margin*), des marges internes (*padding*) et une bordure (*border*) aux dimensions du contenu.

Pour les éléments de type "*inline*" ces propriétés sont égales à 0.

CSS va permettre de sortir les éléments du flux courant.

Ajoutons qu'avec CSS2 la notion de profondeur a été introduite avec z-Index.

7.2 - LES ÉLÉMENTS BLOCK ET INLINE

- **Quelques éléments block**

`<div>`, `<h1>` ... `<h6>`, `<p>`, `` `` `<dl>`, ``, `<table>`, `<blockquote>`, `<pre>`, `<form>`, `<hr />`.

- **Quelques éléments Inline**

``, `<a>`, ``, `` (obsolète), ``, `<i>`(obsolète), ``, `
` `<input />`, `<abbr>`, `<acronym>`, `<label>`, `<input ... />`

- **Comportement**

Il est possible de modifier le comportement de positionnement d'un élément avec la propriété `display` (Block/Inline).

Par exemple des `` en Inline pour réaliser un menu horizontal ou des `<a>` en block pour réaliser un effet bouton.

Un élément block (sauf pour les paragraphes `<p>`) peut contenir d'autres éléments blocks.

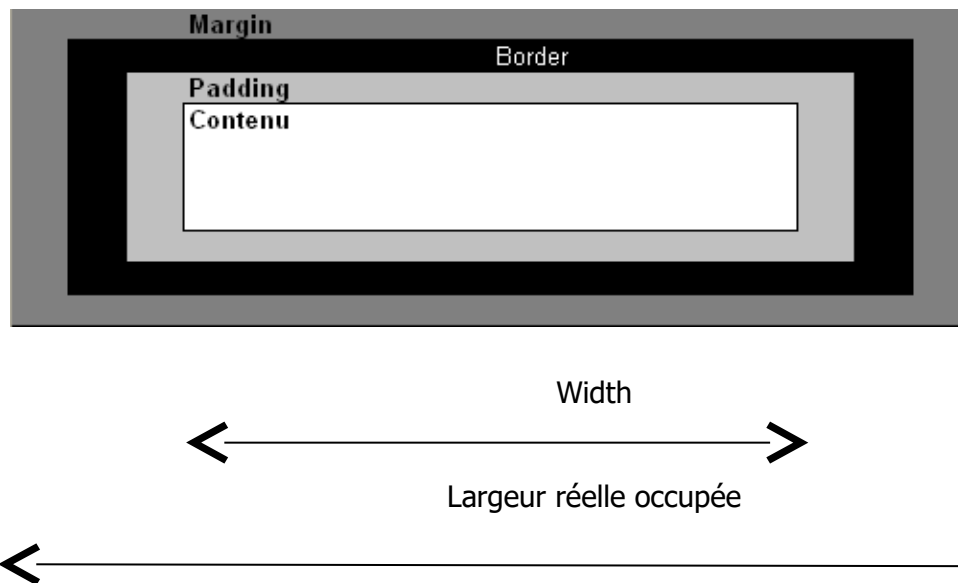
Un élément en ligne ne peut contenir que d'autres éléments en ligne sauf exceptions.

7.3 - LE DIMENSIONNEMENT

7.3.1 - Le système HTML de boîtes

La largeur d'un élément est définie par width.
La hauteur d'un élément est définie par height.

La boîte du W3C.



Selon le W3C la dimension d'une boîte est en largeur : la somme des marges externes, du bord, des marges internes et du contenu. Idem pour la hauteur.

Si par exemple un élément se voit affecter une largeur de 200 pixels, une hauteur de 100 pixels, une marge extérieure et une marge intérieure de 20 pixels sur tous les côtés, la largeur de l'élément mesure 280 pixels: $20+20+200+20+20$.

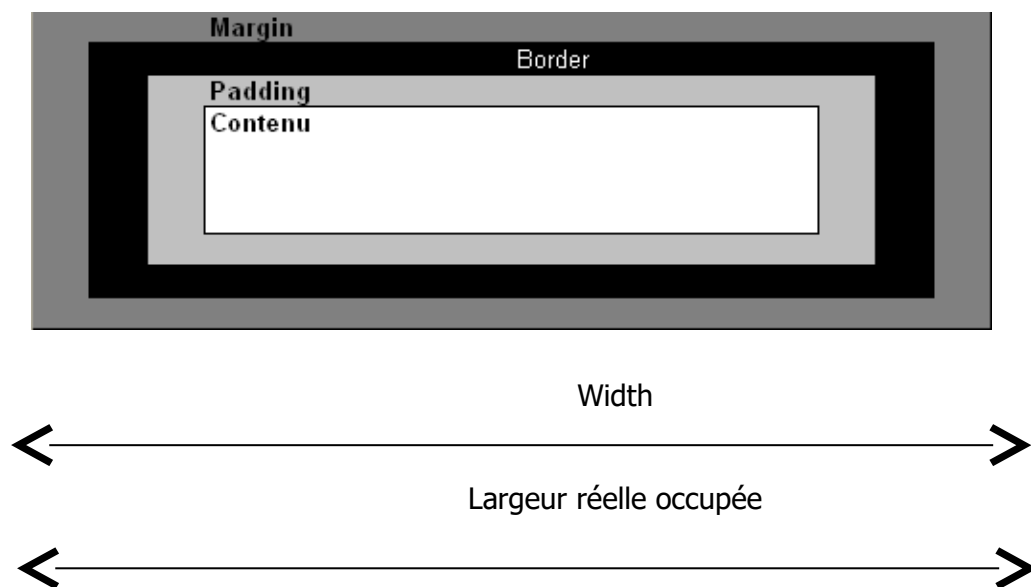
Donc pour le W3C avec la propriété width nous définissons le contenu.

Pour IE (Jusqu'à la version 6 incluse) la largeur que nous définissons correspond à la dimension du contenu ainsi que des marges et bords. Donc une boîte IE est plus petite qu'une boîte W3C.

Si par exemple un élément se voit affecter une largeur de 200 pixels, une hauteur de 100 pixels, une marge extérieure et une marge intérieure de 20 pixels sur tous les côtés, la largeur de l'élément mesure 200 pixels: et le contenu mesure: $-20-20+200-20-20$ donc 120 pixels.

Ainsi le rendu est très différent selon les navigateurs.

La boîte d IE (Versions ≤ 6).



7.3.2 - Dimensionnement par défaut

Les propriétés qui permettent le dimensionnement sont **width** (pour la largeur) et **height** (pour la hauteur).

Par défaut les éléments ne sont pas dimensionnés. Ils occupent tout l'espace attribué en standard.

Les éléments de type *block* occupent la largeur du document.

Les éléments *inline* occupent la place correspondant au contenu.

Les éléments *block* peuvent être dimensionnés.

Les éléments *inline* se divisent en deux catégories :

- ✓ les éléments "**remplacés**" (, <input>, <textarea>, <select>, <object>) qui sont **dimensionnables**.
- ✓ les éléments "**non-remplacés**" (, ,) qui ne sont pas **dimensionnables**.

Les marges externes (margin) et les marges internes – dites aussi remplissage - (padding) vont jouer un rôle dans le dimensionnement mais aussi dans le positionnement.

- **Exemple**

Affichage d'éléments blocs et d'éléments inline sans dimensionnement.

Les éléments block prennent toute la largeur.

Les éléments inline sont dimensionnés en fonction du contenu.



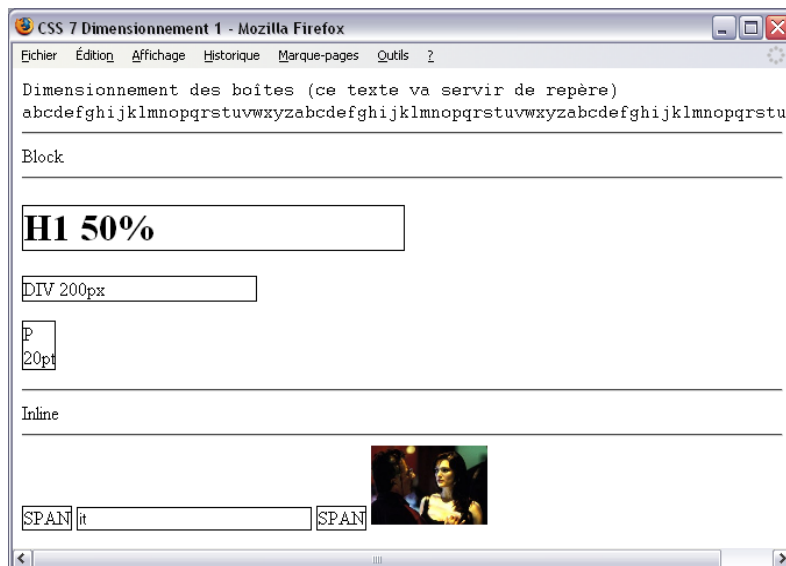
```
<head>
<title>Dimensionnement standard</title>
<style type="text/css">
  .borde{border:solid 1px black;}
</style>
</head>
<body>
<span class="measure">
Dimensionnement des boîtes (ce texte va servir de repère) <br />
abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz
</span>
<hr />Block<hr />
<h1 class="borde">H1 200px</h1>
<div class="borde">DIV 200px</div>
<p class="borde">P 20pt</p>
<hr />Inline<hr />
<span class="borde">SPAN</span>
<input class="borde" id="it" type="text" value="it" size="20" />
<span id="espagne" class="borde">SPAN</span>

</body>
```

7.3.3 - Éléments dimensionnés

- **Premier exemple**

Reprenons le même document HTML et appliquons des propriétés de dimensionnement. Seuls les `` ne sont pas modifiables.



```
<head>
<title>Dimensionnement standard dimensionne</title>
<style type="text/css">
    .borde{border: 1px solid black;}

    .mesure{font-family:courier; font-size:12pt;}
    h1{width:50%;}
    div{width:200px;}
    p{width:20pt;}
    #espagne{width:50em;}
    #it{width:200px;}
    #photo{width:100px;}
</style>
</head>

<body>

<span class="mesure">
Dimensionnement des boîtes (ce texte va servir de repère au
dimensionnement des éléments).<br />
abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz
</span>

<hr />Block<hr />
<h1 class="borde">H1 50%</h1>
<div class="borde">DIV 200px</div>
<p class="borde">P 20pt</p>
```

```
<hr />Inline<hr />
<span class="borde">SPAN</span>
<input class="borde" id="it" type="text" value="it" size="20" />
<span id="espagne" class="borde">SPAN</span>

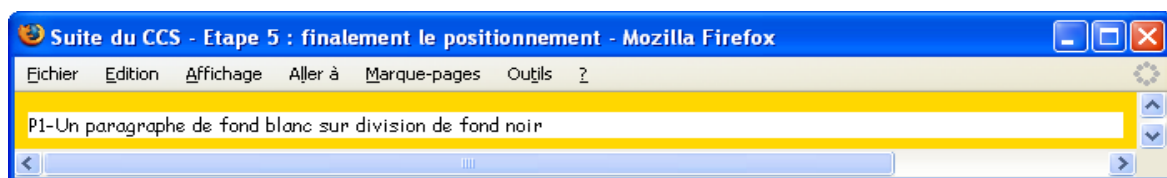
</body>
```

• Autre exemple

Si un paragraphe <p> de fond blanc est inséré dans une division <div> de fond noir il remplira la division. Le fond noir ne sera pas visible.

```
<style type="text/css">
#div1{ background: black; }
#p1{ background: white; }
</style>
```

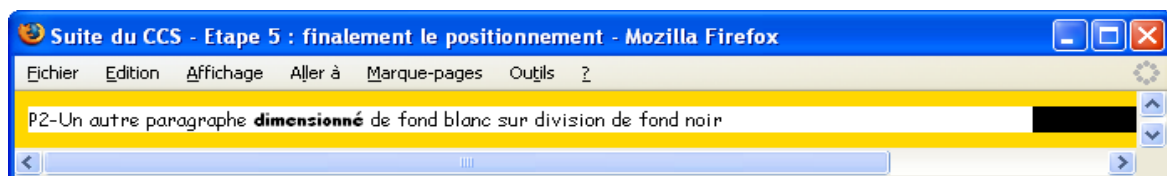
```
<div id="div1"><p id="p1">P1-Un paragraphe de fond blanc sur division de
fond noir</p></div>
```



La division et le paragraphe sont dimensionnés. Le fond noir apparaît.

```
<style type="text/css">
#div2{ background:black; width: 800px; }
#p2{ background:white; width: 600px; }
</style>
```

```
<div id="div2"><p id="p2">P2-Un autre paragraphe
<strong>dimensionné</strong> de fond blanc sur division de fond noir</p></div>
```



7.3.4 - Les marges

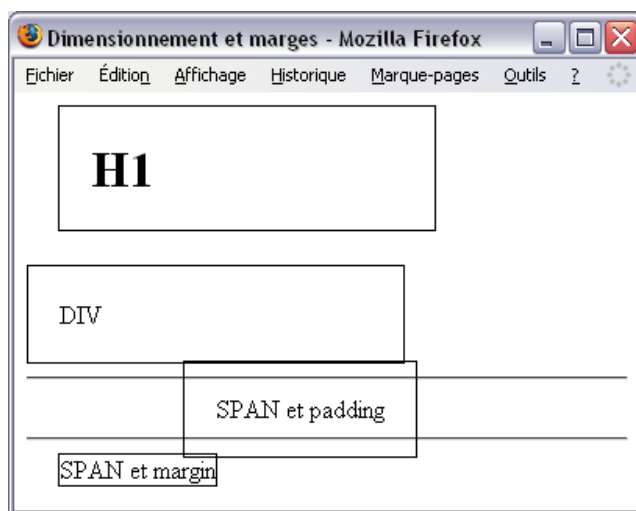
Les marges internes (*padding*) et les marges externes (*margin*) redimensionnent et déplacent l'élément, en tous les cas le contenu. D'ailleurs pour centrer un élément on joue sur les marges right et left à auto (cf plus loin).

- **Premier exemple**

Reprenons encore le premier document (en enlevant certains éléments) en ajoutant des propriétés de style modifiant les marges internes (padding) de certains éléments.

Le dernier possède une marge externe qui le déplace sans le redimensionner. Ce sera l'objet du paragraphe suivant.

Les éléments sont bien redimensionnés.



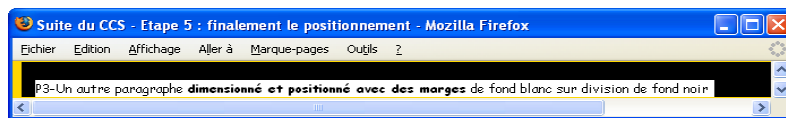
```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Dimensionnement et marges</title>
<style type="text/css">
    .borde{border:1px solid black}
    h1{width:200px; padding:20px; margin-left:20px;}
    div{width:200px; padding:20px;}
    #espagne{padding:20px; margin:100px;}
    #portugal{margin:20px;}
</style>
</head>
<body>
    <h1 class="borde">H1</h1>
    <div class="borde">
        DIV
    </div>
    <hr/>
    <span class="borde" id="espagne">SPAN et padding</span>
    <hr/>
    <span class="borde" id="portugal">SPAN et margin</span>
</body>
```

- **Autre exemple**

La propriété `margin` appliquée au `<p>` blanc le déplace et la propriété `padding` appliquée à la `<div>` noire l'agrandit et déplace le paragraphe blanc.

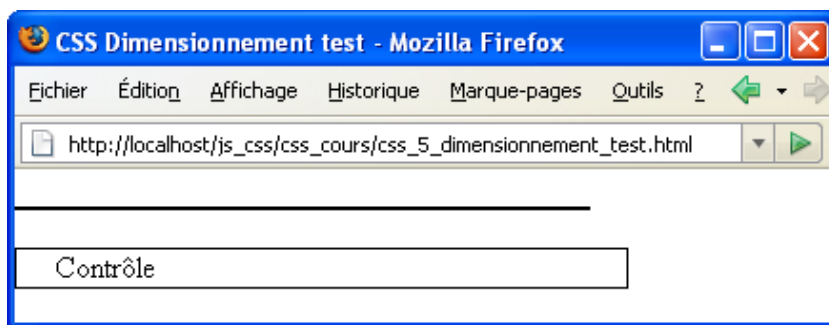
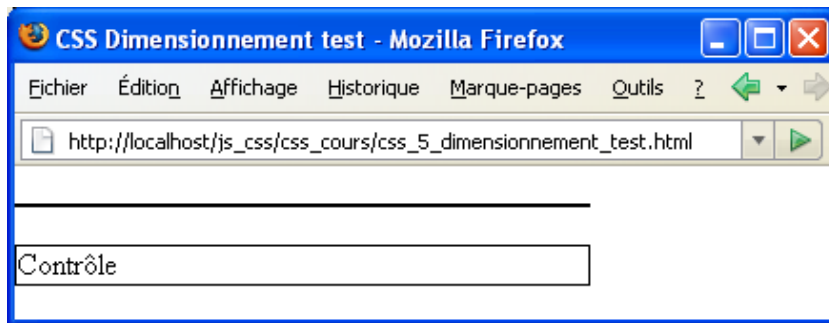
```
<style type="text/css">
#div3{ background:black; width:800px; padding-top:1ex; }
#p3{ background:white; width:600px; margin-left:1em; }
</style>
```

```
<div id="div3">
<p id="p3">P3-Un autre paragraphe dimensionné et positionné avec
des marges</strong> de fond blanc sur division de fond noir
</p>
</div>
```



- **Dimensionnement et padding**

L'agrandissement de l'élément par le padding.



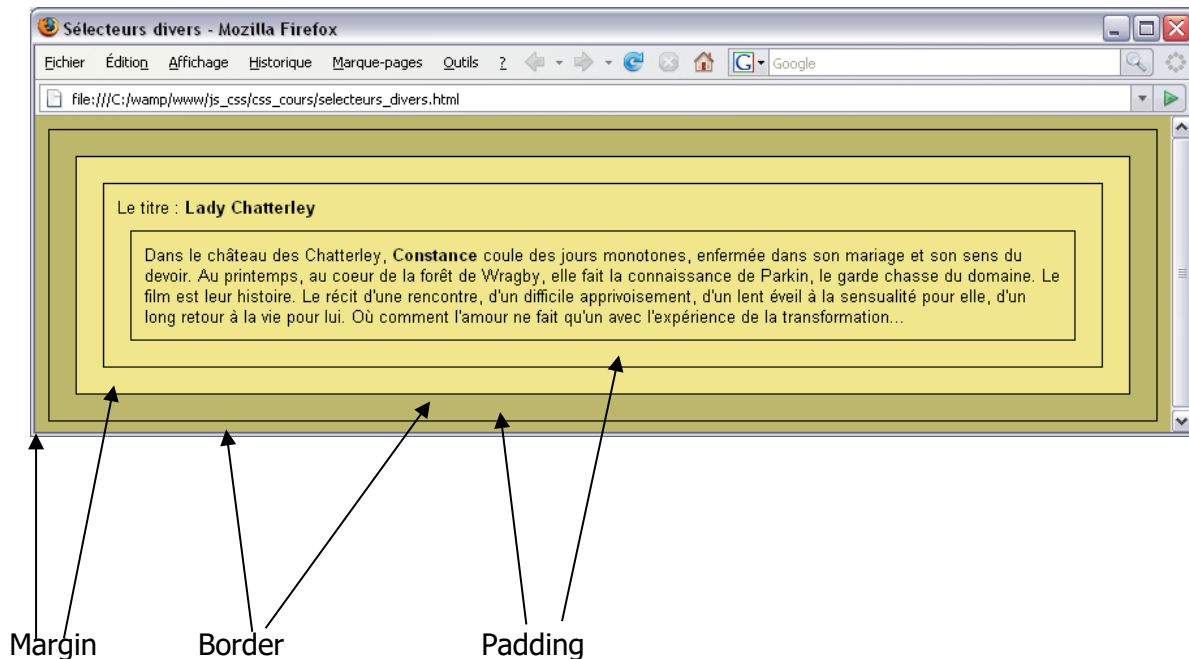
```
<style type="text/css">
  *{ margin: 0; padding:0; }
  span#empan{ width: 300px; border: 1px solid black; display:block; }
  div#controle{ width: 300px; border: 1px solid black; }
</style>
```

```
<body>
  <br /><span id="empan"></span><br />
  <div id="controle">Contrôle</div>
</body>
```

Dans le deuxième cas ceci change : le padding-left de la <div>

```
div#controle{ width: 300px ; border: 1px solid black ; padding-
left:10px; }
```

- **Padding et margin**



```
<head>
<title>Sélecteurs divers</title>
<style type="text/css">
    *{margin:10px; padding:10px; border:1px solid black;}
    html{background-color:darkkhaki;}
    body{background-color:khaki; font-family:arial; font-size:small;}
    strong {padding:0; margin:0; border:0;}
</style>
</head>

<body>
    <div>
        Le titre : <strong>Lady Chatterley</strong>
        <p>
            Dans le château des Chatterley,
            <strong>Constance</strong> coule des jours monotones, enfermée dans son
            mariage et son sens du devoir. Au printemps, au coeur de la forêt de
            Wragby, elle fait la connaissance de Parkin, le garde chasse du domaine
        </p>
    </div>
</body>
```


7.4 - LE POSITIONNEMENT

7.4.1 - Généralités

Un document HTML est composé de boîtes. Ces boîtes ont des rapports de parenté ou hiérarchiques entre elles : ancêtres, parents, enfants, frères.

Un ancêtre est un élément qui en contient un autre ou plusieurs. Les éléments qui ont le même parent sont frères.

Les boîtes sont disposées dans le flux courant. Le positionnement va modifier la disposition naturelle du flux courant.

Dans le flux courant les éléments *block* se suivent sur des lignes différentes (comme si l'on ajoutait un retour chariot) alors que les éléments *inline* se suivent sur la même ligne tant qu'il y a de la place; ensuite ils passent à la ligne suivante. (cf le premier exemple sur le dimensionnement).

Principalement le positionnement peut être **absolu**, **relatif** ou **flottant**. Nous verrons aussi le **fixe**.

Les positionnements absolu et fixe font sortir l'élément du flux courant.

Le positionnement relatif n'effectue qu'un décalage.

Le positionnement flottant ne sort pas non plus du flux courant. Le premier élément flottant oui, les autres non.

Comme nous l'avons vu les marges jouent aussi sur le positionnement.

7.4.2 - Point de départ : le flux courant

Les éléments sont positionnés dans le flux courant en fonction de leur nature (Block/Inline).



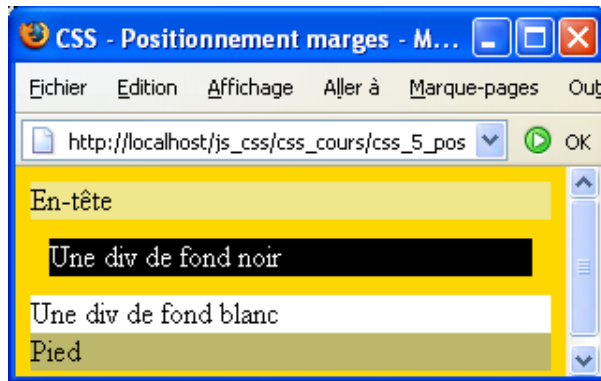
```
<style type="text/css">
  body{ background-color: gold ;}
  #en_tete{ background: khaki; color: black; }
  #noir{ background: black; color: white; }
  #blanc{ background: white; color: black; }
  #pied_de_page{ background: darkkhaki; color: black; }
</style>

<body>
  <div id="en_tete">En-tête</div>
  <div id="noir">Une div de fond noir</div>
  <div id="blanc">Une div de fond blanc </div>
  <div id="pied_de_page">Pied</div>
</body>
```

7.4.3 - Le positionnement avec les marges

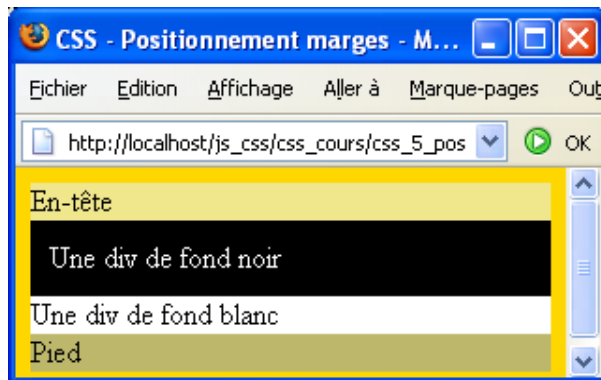
Nous avons déjà vu que les marges jouent sur le positionnement.

- **Marges externes : margin**



```
<style type="text/css">
  body{ background-color:gold ;}
  #en_tete{ background:khaki; color:black; }
  #noir{ background:black; color:white; margin:10px; }
  #blanc{ background:white; color:black; }
  #pied_de_page{ background:darkkhaki; color:black ; }
</style>
```

- **Marges internes : padding**



- **Les deux : margin et padding (déplacement et dimensionnement)**

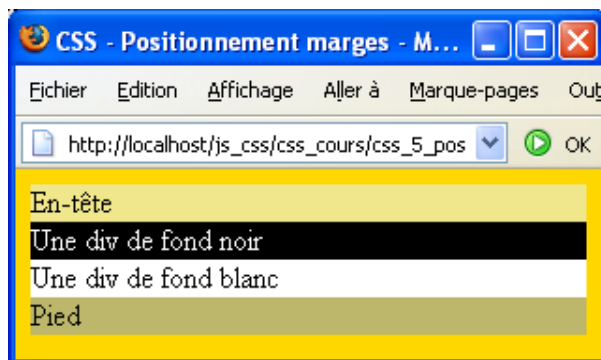


- **Imbrications de <div>**

```
<div id="en_tete">En-tête</div>
<div id="noir">Une div de fond noir
  <div id="blanc">Une div de fond blanc </div>
</div>
```

Sans marges

```
<style type="text/css">
#noir{ background:black; color:white; }
</style>
```

**Avec marges**

```
<style type="text/css">
#noir{ background:black; color:white; margin:10px; padding:10px; }
</style>
```



7.4.4 - Positionnement absolu

Il permet de positionner de façon absolue un élément éventuellement au **pixel près**.

Le positionnement absolu est réalisé avec la valeur *absolute* de la propriété *position*.
Les propriétés **top**, **left** mais aussi **right** et **bottom** permettent de positionner l'élément.

A partir du moment où les éléments sont en position *absolute*, leurs dimensions changent.
Les <div> se comportent comme des éléments *Inline*.



```
<body>
  <div id="en_tete">En-tête</div>
  <div id="noir">Une div de fond noir</div>
  <div id="blanc">Une div de fond blanc </div>
  <div id="pied_de_page">Pied</div>
</body>
```

```
<style type="text/css">
  body{ background-color:gold; }
  #en_tete{ background:khaki; color:black; position:absolute;
top:0px ; left:7px; }
  #noir{ background:black; color:white; position:absolute; top:33px ;
left:7px; }
  #blanc{ background:white; color:black; position:absolute; top:66px ;
left:7px; }
  #pied_de_page{ background:darkkhaki; color:black ;
position:absolute; left:77px; top:100px ;}
</style>
```

Mais le positionnement pour les éléments imbriqués **dépend du positionnement du parent** ou du premier ancêtre positionné (qui peut-être le `<body>` si aucun élément n'est positionné).

Les décalages sont réalisés au moyen des propriétés *top*, *right*, *bottom* et *left*.

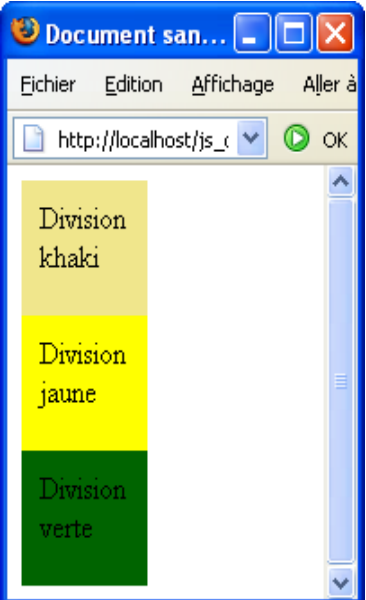
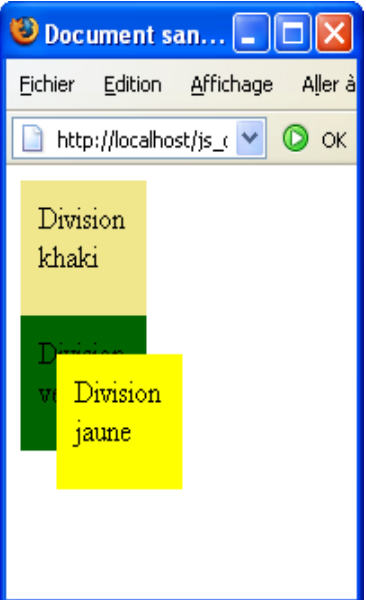
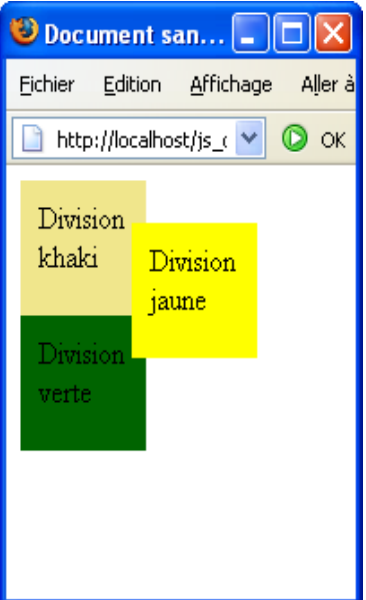
D'où, avec les mêmes styles, un affichage différent en fonction de la structure du document (le décalage devient relatif parce que les `<div>` sont des parents).



```
<body>
  <div id="en_tete">En-tête</div>
  <div id="noir">Une div de fond noir
    <div id="blanc">Une div de fond blanc </div>
  </div>
  <div id="pied_de_page">Pied</div>
</body>
```

La `<div>` blanche est décalée par rapport à la `<div>` noire (puisque qu'elle est imbriquée). Elle est décalée relativement de 66px. Et non plus de 66px par rapport au `<body>` son ex-parent. Son parent est devenu la `<div>` noire.

- **Visibilité de la sortie du flux courant**

Flux courant	Sortie du flux courant (Position:absolute, margin:20px)	Sortie du flux courant (Absolute, Top et Left)
		

Les <div> sont les unes au-dessous des autres.

```
<style type="text/css">
  #kakhi{background-color:khaki; padding:10px; width:50px;
height:50px;}
  #jaune{background-color:yellow; padding:10px; width:50px;
height:50px;}
  #vert{background-color: green; padding:10px; width:50px;
height:50px;}
</style>
```

La division jaune sort du flux, elle change de position du fait du rajout des **marges**. La division verte vient suivre la division kakhi puisque la division jaune n'est plus dans le flux.

```
<style type="text/css">
  #kakhi{background-color:khaki; padding:10px; width:50px;
height:50px;}
  #jaune{background-color:yellow; padding:10px; width:50px;
height:50px; position:absolute; margin:20px;}
  #vert{background-color: green; padding:10px; width:50px;
height:50px;}
</style>
```


Comme la 2 mais cette fois-ci la jaune est positionné de façon absolue.

```
<style type="text/css">
  #kakhi{background-color:khaki; padding:10px; width:50px;
height:50px}
  #jaune{background-color:yellow; padding:10px; width:50px;
height:50px; position:absolute; top:30px; left:70px;}
  #vert{background-color:green; padding:10px; width:50px; height:50px}
</style>
```

7.4.5 - Positionnement relatif

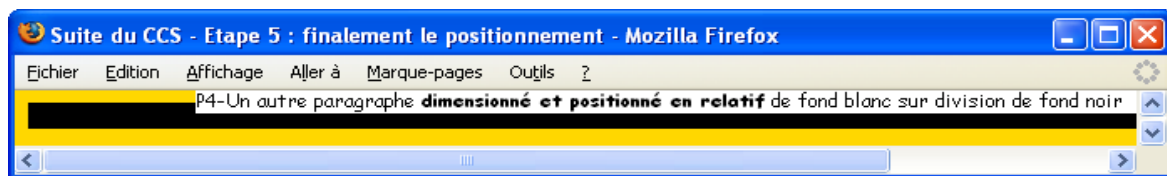
Le positionnement relatif décale l'élément par rapport à sa position normale dans le flux courant.

Pour préciser un positionnement relatif on utilise la propriété *position* avec la valeur *relative*. Plusieurs types de décalage sont possibles (*bottom*, *right*, *top*, *left*).

Top vers le bas, *left* vers la droite, *bottom* décalera vers le haut et *right* vers la gauche.

```
<style type="text/css">
#div4{ background:black ; width:800px; }
#p4{ background:white; width:600px; position:relative; bottom:10px;
left:100px; }
</style>
```

```
<div id="div4">
<p id="p4">P4-Un autre paragraphe <strong>dimensionné et positionné en
relatif</strong> de fond blanc sur division de fond noir
</p>
</div>
```



Définir `div, p{ margin: 0 ; padding:0 ; }` rend les comportements homogènes. Cf le système de boîte de IE.

En reprenant l'exemple du paragraphe précédent et en adoptant un positionnement relatif. La `<div>` ne sort pas du flux (la verte reste à sa place primitive) mais la jaune est décalée par rapport à la khaki.

	<pre><style type="text/css"> #kakhi{background-color:khaki; padding:10px; width:50px; height:50px; } #jaune{background-color:yellow; padding:10px; width:50px; height:50px; position:relative; top:30px; left:70px; } #vert{background-color:darkgreen; padding:10px; width:50px; height:50px; } </style></pre>
--	---

7.4.6 - Positionnement flottant

Une boîte flottante de type bloc est retirée du flux normal, et placée le plus à droite (float:right) ou le plus à gauche (float:left) possible dans son conteneur. Le contenu suivant la boîte flottante s'écoule le long de celle-ci, dans l'espace laissé libre à gauche (float:right) ou à droite (float:left) puis en dessous.

7.4.6.1 - Flux courant



```
<div>
<p><em>Jacques Dutronc</em></p>
<p>
    
</p>
<p>
    L'homme au cigare et aux lunettes noires naît le 28 avril 1943 à
    Paris. Son père, ingénieur, est un grand amateur de musique et très jeune,
    Jacques apprend le piano avant de se consacrer plutôt à la guitare.
    Adolescent peu ...
</p>
</div>
```

7.4.6.2 - Flux flottant

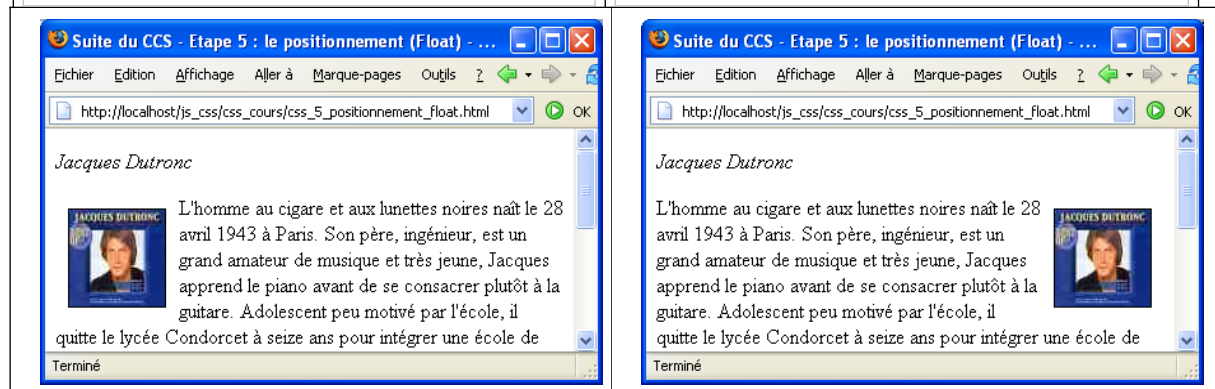
pour l'image d'identifiant #flottant

```

```

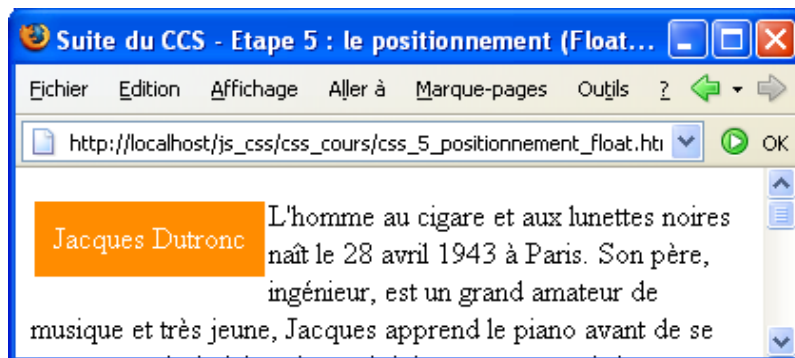
```
#flottant{float:left; margin:10px;
}
```

```
#flottant{float:right;
margin:10px; }
```



7.4.6.3 - Lettrine

L'effet lettrine est obtenu avec un texte flottant.

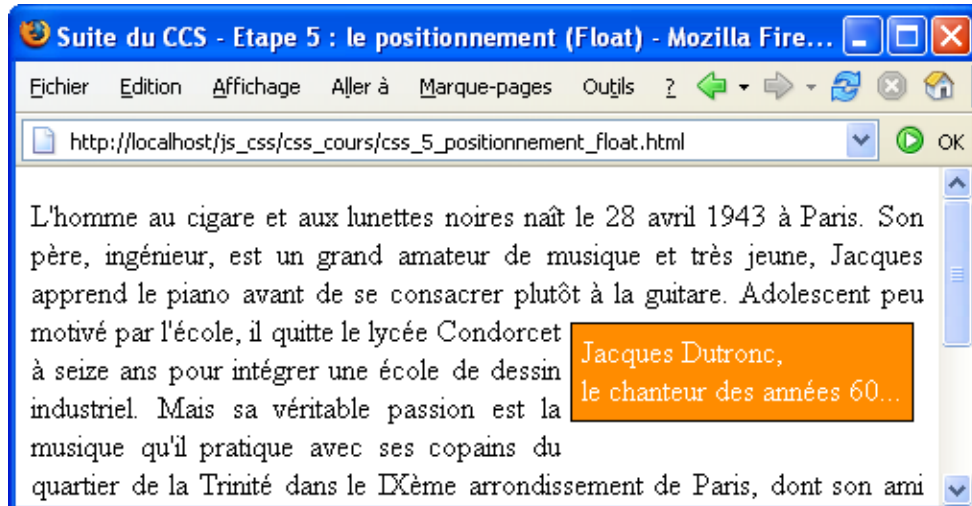


```
<style type="text/css">
.lettrine{ float:left; background-color: darkorange; color: white; font:
24px; padding:10px; margin:2px}
</style>
```

```
<p class="lettrine">Jacques Dutronc</p>
<p>L'homme au cigare et aux lunettes noires naît le 28 avril 1943 à Paris.
Son père, ingénieur, est un grand amateur de musique et très jeune,
Jacques apprend le piano avant de se consacrer plutôt à la guitare.
Adolescent peu motivé par l'école, il quitte le lycée Condorcet à seize
ans pour intégrer une école de dessin industriel. Mais sa véritable
passion est la musique qu'il pratique avec ses copains du quartier de la
Trinité...</p>
```

7.4.6.4 - Encadre

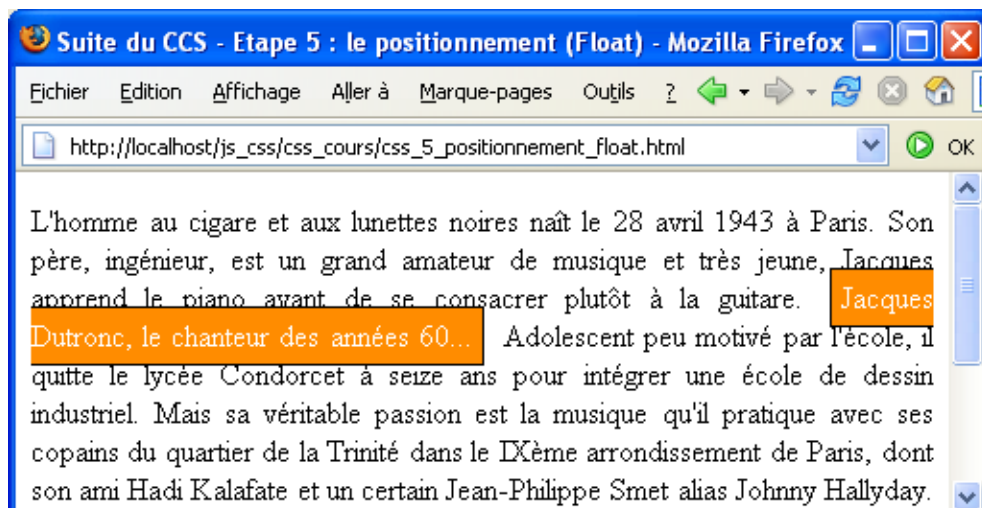
Un encadré est obtenu aussi avec un flottant.



```
<style type="text/css">
.encadre{ float:right; background-color: darkorange; color: white; font:
24px; padding:5px; margin:5px; border: 1px solid black; }
p{ text-align:justify;}
</style>
```

```
<p>L'homme au cigare et aux lunettes noires naît le 28 avril 1943 à Paris.
Son père, ingénieur, est un grand amateur de musique et très jeune, Jacques
apprend le piano avant de se consacrer plutôt à la guitare.
<span class="encadre">Jacques Dutronc,<br /> le chanteur des années
60...</span>
Adolescent peu motivé par l'école, il quitte le lycée Condorcet à seize
ans pour intégrer une école de dessin industriel.
</p>
```

NB : sans flottant nous aurions eu ceci :



7.4.7 Remarques sur le float

Admettons cette structure de page : l'en-tête de page, une zone centrale contenant un menu et un contenu, un pied de page.

- **Point de départ**

```
<body>
  <div id="en_tete">Le flottement</div>
  <div id="centre">
    <div id="centre_gauche">
      <ul id="menu_gauche">
        <li><a href="#">Item</a></li>
        <li><a href="#">Item</a></li>
        <li><a href="#">Item</a></li>
        <li><a href="#">Item</a></li>
      </ul>
    </div>
    <div id="centre_droit">Contenu</div>
  </div>
  <div id="pied">Copyright PB & co</div>
</body>
```

qui rend ceci



Nb : les <div> sont bordées :

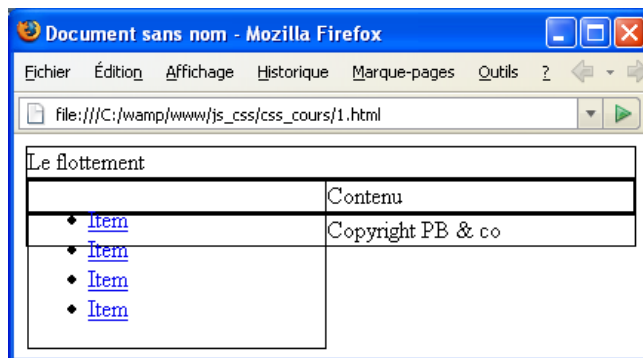
```
<style type="text/css">
  div{border:1px solid black;}
</style>
```

- **Appliquons le flottement pour la division "centre_gauche".**

```
<style type="text/css">
  div{border:1px solid black;}
  #centre_gauche{float:left; width:200px;}
</style>
```

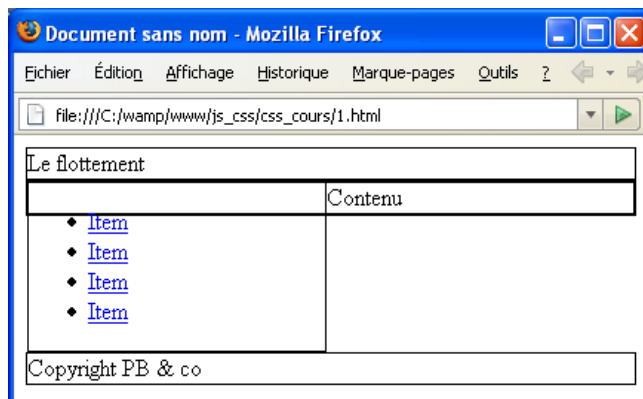
Cela donne ceci; on remarque que la division "centre_droit" est positionnée aux mêmes coordonnées que la division flottante comme si cette dernière sortait du flux, et que la division "pied" se place sous la division "centre_droit" et que leur contenu respectifs sont à la droite de la division flottante.

Ceci est correct, par rapport à ce qui est attendu concernant le contenu, mais ne l'est pas pour le "pied".



- **Déplaçons le "pied"**

En fait le flottement se transmet aux "frères". Il faut donc annuler l'effet de flottement avec la propriété **clear**. Les valeurs sont **left**, **right**, **both**.



```
<style type="text/css">
  div{border:1px solid black;}
  #centre_gauche{float:left; width:200px;}
  #pied{clear:left;}
</style>
```

Le même rendu, plutôt déplacement, est obtenu avec un dimensionnement en hauteur de la division "centre" ou de la division "centre_droit". L'un n'empêche pas l'autre d'ailleurs.

```
#centre{ width:100%; height:100px}
```

7.4.8 Le positionnement fixe

Le positionnement fixe permet de fixer la position d'un élément quelque soit le défilement dans la page au moyen des barres de défilement. Alors que les positionnements précédents étaient liés au document, celui-ci est lié à la fenêtre (**KO avec IE**).

- **Ecrans**

Dans cet exemple le paragraphe qui contient le texte "Le masque" ne change pas de position même si l'on se déplace avec l'ascenseur ou le tapis roulant.



- **Script**

```
<head>
  <title>CSS - Positionnement fixe</title>

  <style type="text/css">
    body{ background-color: gold; color: black; font-family: 'Comic
sans MS'; font-size: 12px; }

    #en_tete{ color: black; font: italic bold 120%/140% Verdana ;
position:fixed; top: 10px; left: 10px; }
    #pied_de_page{ color: black ; font: italic bold 120%/140%
Verdana; position:fixed; top: 100px ; left: 10px; }

    #image1{position:absolute; top:20px; left:200px; }
  </style>
</head>
```

```
<body>

  <p id="en_tete">Le masque</p>

  <p id="pied_de_page">Pied</p>
</body>
```

7.4.9 La profondeur

Par défaut un élément s'affiche par dessus un autre élément.

La propriété *z-index* permet de modifier le comportement standard.

Les valeurs possibles vont de 0 à 99 ... et plus. Une valeur plus élevée signifie un plan en avant.

```
<style type="text/css">
#div6{ background:black; width:800px; height:60px; position:absolute;
top:10px; left:10px; }
#p6_0{ background:white; width:600px; height:20px; position:absolute;
top:10px; left:10px; z-index:0; }
#p6_1{ background:yellow; width:600px; height:20px; position:absolute;
top:20px; left:20px; z-index:1; }
</style>
```

```
<div id="div6">
  <p id="p6_1">
    Un paragraphe <strong>de premier plan</strong> en noir et blanc
  </p>
  <p id="p6_0">
    Un paragraphe <strong>de second plan</strong> en noir et blanc
  </p>
</div>
```



Si l'on inverse les ordres on obtient ceci.



7.5 - CENTRER LES ÉLÉMENTS

7.5.1 - Les éléments block

7.5.1.1 - Centrage horizontal

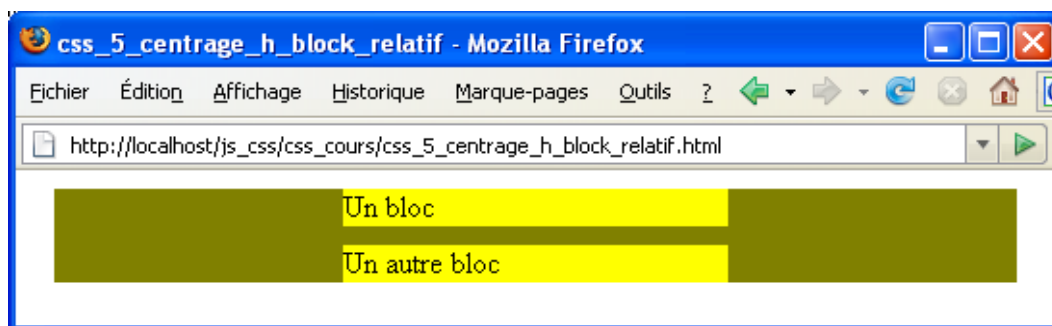
Le centrage horizontal peut être réalisé :

- ✓ soit de façon "relative"
- ✓ soit en utilisant des positionnements "absolus"

- **Relatif**

En utilisant les marges left et right avec la valeur auto.

Exemple : Paragraphes centrés.



```
<head>
<title>css_5_centrage_block_relatif</title>

<style type="text/css">
    div#conteneur{ background:yellow; width:500px; }
    div#conteneur p{ background:lightgreen; width:200px ; margin:10px
auto; }
</style>
</head>

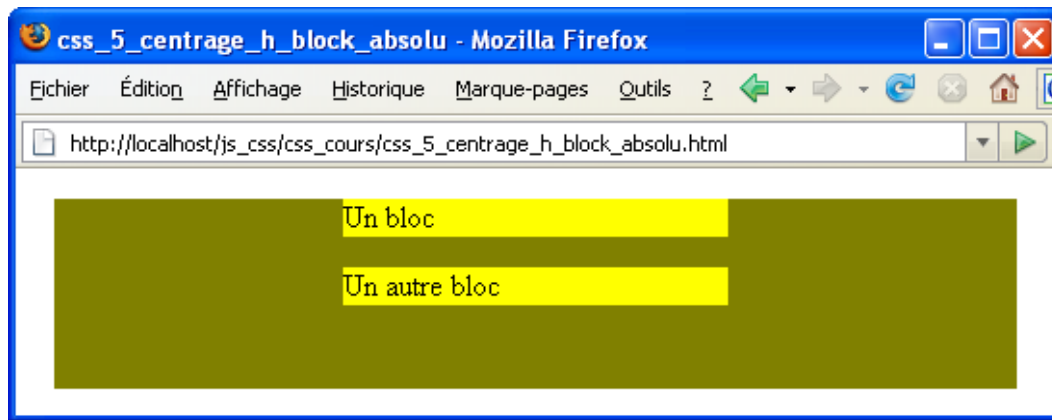
<body>
    <div id="conteneur">
        <p>Un bloc</p>
        <p>Un autre bloc</p>
    </div>
</body>
```

Pour centrer un body (façon blog) :

```
*{margin:0; padding:0; font-family:Arial;}
html{background-color:silver;}
body {width:810px; margin:20px auto;}
```

- **Absolu**

Exemple : Paragraphes centrés.



Les deux conteneurs sont positionnés en relatif.

Le deuxième est décalé de 50% dans la largeur, puis sa marge est négative (la moitié de sa largeur).

```
<title>css_5_centrage_h_block_absolu</title>

<style type="text/css">
    div#conteneur{ position:relative;background:olive; width:500px ;
height:100px; margin:10px auto; }

    div#conteneur p{ position:relative; background:yellow; width:200px ;
left:50%; margin-left:-100px; }
</style>
</head>

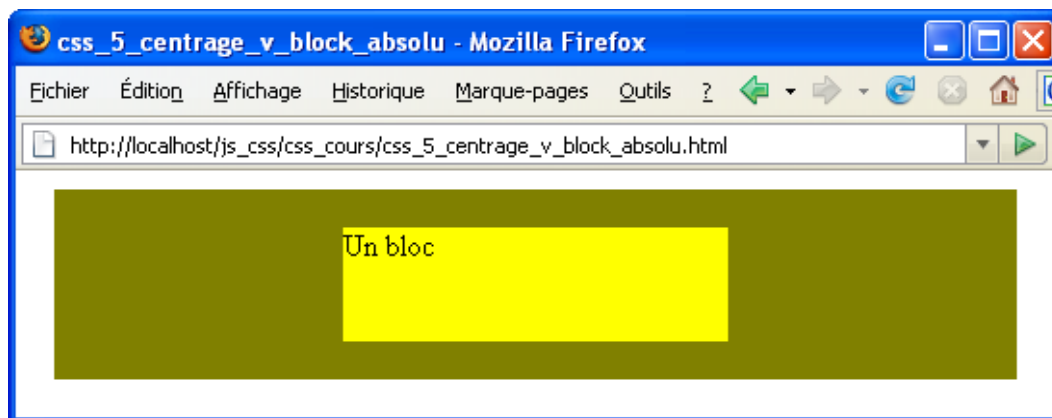
<body>
    <div id="conteneur">
        <p>Un bloc</p>
        <p>Un autre bloc</p>
    </div>
</body>
```

NB : s'il n'y a qu'un seul élément sa position peut être absolue, mais s'il y en a deux il faut qu'elle soit relative.

7.5.1.2 - Centrage vertical

Il n'existe pas de possibilité de centrage relatif vertical des conteneurs. Donc nous utilisons la même technique que pour le centrage horizontal absolu.

Exemple : Paragraphe centré.



Le premier conteneur est positionné en relatif, le deuxième en absolu.
Le deuxième est décalé de 50% dans la hauteur, puis sa marge est négative (la moitié de sa hauteur).

```
<title>css_5_centrage_v_block_absolu</title>

<style type="text/css">
  div#conteneur{
    position: relative;
    background: olive;
    width: 500px;
    height: 100px;
    margin: 10px auto;
  }
  div#conteneur p{
    position: absolute;
    background: yellow;
    width: 200px;
    height: 60px;
    left: 50%;
    margin-left: -100px;
    top: 50%; margin-top: -30px;
  }
</style>
</head>

<body>
  <div id="conteneur">
    <p>Un bloc</p>
  </div>
</body>
```

7.5.2 - Les éléments Inline

7.5.2.1 - Centrage horizontal

Un texte centré dans une <div> avec la propriété **text-align**.



```
<title>css_5_centrage_h_inline_relatif</title>

<style type="text/css">
  div#conteneur{
    background:yellow;
    width:300px;
    height:100px;
    margin:10px auto;
  }
  .texte{ text-align:center; }
</style>
</head>

<body>
  <div id="conteneur" class="texte">
    Un texte inline
  </div>
</body>
```

La même chose avec une image à la place du texte.



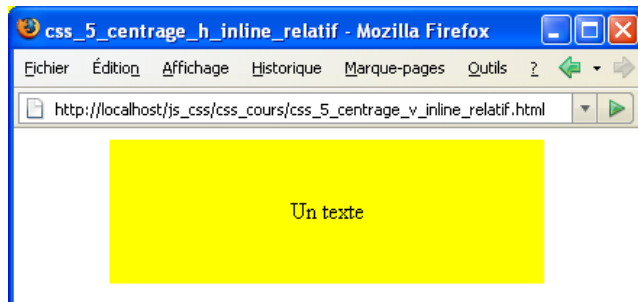
7.5.2.2 - Centrage vertical

Notez que la propriété **vertical-align(*)** ne porte pas sur tous les éléments (cellule de tableau – table-cell - , éléments de formulaire, span, ...).

Dans les autres cas il faut utiliser **line-height** qui centre verticalement.

L'élément inline doit avoir la même hauteur que le conteneur.

De la vraie « bidouille ». Cela ne fonctionne plus si le texte est sur 2 lignes ...



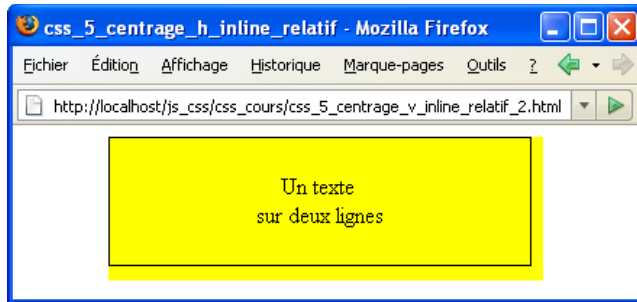
```
<title>css_5_centrage_h_inline_relatif</title>

<style type="text/css">
  div#conteneur{
    background: yellow;
    width: 300px;
    height: 100px;
    margin: auto;
  }
  .texte{
    text-align: center;
    line-height: 100px;
  }
</style>
</head>

<body>
  <div id="conteneur" class="texte">
    Un texte
  </div>
</body>
```

NB: avec table-cell. Mais cela ne fonctionne pas sous IE 7.

```
<style type="text/css">
    div#conteneur{ background:yellow; width:300px ; height:100px; margin:auto }
    .texte{ text-align:center; display:table-cell; vertical-align:middle; height:90px; width:290px; border:1px black solid;}
</style>
<div id="conteneur" >
    <!-- -->
    <label class="texte">Un texte <br /> sur deux lignes</label>
</div>
```



(*) : Les valeurs de vertical-align sont : top, super, middle, baseline, bottom, ...

7.6 - LES <TABLE> ET CSS

Cf CSS.odt.

8 - AFFICHER ET MASQUER

Les propriétés **Visibility** et **Display** permettent d'afficher ou de masquer un élément.

Mais c'est la pseudo classe **:hover** qui permet d'afficher ou de masquer **dynamiquement** des éléments en pur CSS.

La norme W3C spécifie que tout élément d'un document peut implémenter la pseudo-classe **:hover**. Or IE (jusqu'à la version 6) ne l'implémente que pour la balise `<a>`. C'est donc avec cette balise que nous verrons les effets Afficher/Masquer.

C'est plus exactement avec la propriété *display* que l'on peut produire ces effets dynamiques. Les valeurs sont *none*, *inline* ou *block*; elles permettent respectivement de masquer, d'afficher en ligne ou d'afficher un bloc.

La propriété **display**(none, block, inline) diffère de la propriété **Visibility**(hidden, visible).

Display:none et Visibility:hidden masquent l'élément.

Display:inline ou display:block et Visibility:visible affichent l'élément.

Avec Display:none l'élément n'occupe aucune place dans le document. Il sort du flux.

Avec visibility:hidden une place est réservée.

Cf CSS.odt pour l'application avec des menus commentés (info-bulles).

9 - ANNEXES

9.1 - LES RACCOURCIS

Border

`#id{ border: épaisseur style couleur; }`

```
#DivPopup{ border:1px solid silver; }
```

9.2 - LINK

```
<link type="text/css" rel="stylesheet" href="chemin/fichier.css" />
```

9.3 - UN MODÈLE DE PAGE AVEC LE FOOTER FIXE

La photo !!!

Header
Nav
Article
Footer

Lorsque l'on manipule l'ascenseur le Header et le Footer ne bougent pas.

Header
Nav
Article

Footer

```
<body>
  <header>
    Header
  </header>
  <nav>
    Nav
  </nav>
  <article>
    Article
  </article>
  <footer>
    Footer
  </footer>
</body>
```

```
*{margin: 0; padding: 0;}
html{
  height: 150%;
}
body{
  height: 150%;
  margin-top: 1%;
  margin-left: 1%;
}
header{
  background-color: black;
  color: white;
  height: 5%;
  width: 100%;
  position: fixed;
  top: 0;
}
nav{
  background-color: gray;
  color: white;
  height: 5%;
  padding: 1%;
  position: relative;
  top: 1%;
}
article {
  height: 70%;
  position: relative;
  top: 1%;
}
footer{
  background-color: black;
  color: white;
  height: 5%;
  width: 100%;
  position: fixed;
  bottom: 0;
}
```

9.4 - MENUS

9.4.1 - Menu vertical

[Se connecter](#)
[Produits](#)
[Se déconnecter](#)

Rien de spécial

```
<nav id="menuVertical">
  <ul>
    <li><a href="#">Se connecter</a></li>
    <li><a href="#">Produits</a></li>
    <li><a href="#">Se déconnecter</a></li>
  </ul>
</nav>
```

```
#menuVertical {
  margin: 0;
}
#menuVertical ul {
  list-style: none;
}
#menuVertical ul li a {
  color: white;
}
```


9.4.2 - Menu horizontal

[Se connecter](#) | [Produits](#) | [Se déconnecter](#)

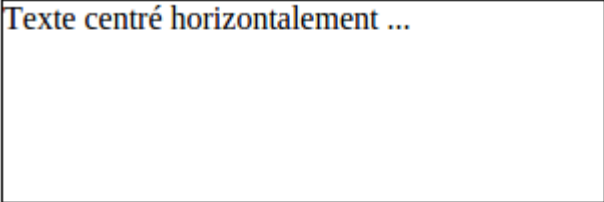
```
<nav id="menuHorizontal">
  <ul>
    <li><a href="#">Se connecter</a></li>
    <li> | </li>
    <li><a href="#">Produits</a></li>
    <li> | </li>
    <li><a href="#">Se déconnecter</a></li>
  </ul>
</nav>
```

```
#menuHorizontal {
  margin: 0;
}
#menuHorizontal ul {
  list-style: none;
}
#menuHorizontal ul li {
  display: inline-block;
}
#menuHorizontal ul li a {
  color: white;
}
```

9.5 - CENTRER ...

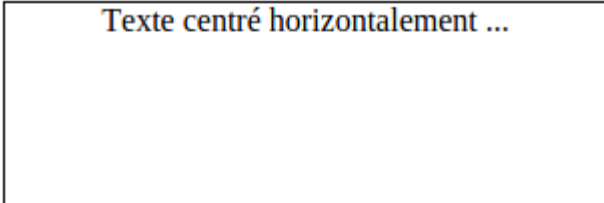
9.5.1 - Centrer un texte horizontalement

Pas centré !



Texte centré horizontalement ...

Centré ...



Texte centré horizontalement ...

```
<div id="texteCentreHorizontalement">  
  Texte centré horizontalement ...  
</div>
```

```
#texteCentreHorizontalement{  
  border: 1px solid black;  
  width: 300px;  
  height: 100px;  
  text-align: center;  
}
```

9.5.2 - Centrer un texte verticalement

La propriété **vertical-align(*)** ne porte pas sur tous les éléments (cellule de tableau – table-cell -, éléments de formulaire, span, ...).

Dans les autres cas il faut utiliser **line-height** qui centre verticalement. L'élément inline doit avoir la même hauteur que le conteneur.



```
<div id="divCentreVerticalement">  
  Texte centré verticalement ...  
</div>
```

```
#divCentreVerticalement{  
  border: 1px solid black;  
  width: 300px;  
  height: 100px;  
  line-height: 100px;  
}
```

9.5.3 - Centrer une photo horizontalement



```
<div id="div300">
  
</div>
```

```
#div300 {
  border: 1px solid black;
  width: 300px;
  display: block;
}

#imageCentreHorizontalement {
  display: block;
  margin-left: auto;
  margin-right: auto;
}
```

9.6 - UN FORMULAIRE

9.6.1 - Un formulaire <inline-block>

Prénom :

Nom :

```
<form>
  <label class="etiquette">Prénom : </label>
  <input type="text" name="prenom" />
  <br>
  <label class="etiquette">Nom : </label>
  <input type="text" name="nom" />
  <br>
  <input type="submit" />
</form>
```

```
.etiquette{display: inline-block; width: 100px;}
input{margin: 5px; padding: 3px;}
```

9.6.2 - Un formulaire

TODO

Prénom :

Nom :

```
<form>
  <label class="etiquette">Prénom : </label>
  <input type="text" name="prenom" />
  <br>
  <label class="etiquette">Nom : </label>
  <input type="text" name="nom" />
  <br>
  <input type="submit" />
</form>
```