

Menu en absolu. Largeur fixe (15em)

## CSS = Cascading Style Sheets

- écrits dans un fichier .css

Exemple : style1.css

- un même fichier peut être partagé par plusieurs pages  $\Rightarrow$  Uniformité de présentation)
- ajouter dans l'en-tête (head) de la page HTML qui utilise le CSS :

```
1 <link rel="stylesheet"
2   type="text/css" href="style1.css">
```

- la css définit des règles pour appliquer des *styles* (couleur, police, etc.) à des parties d'une page HTML.

## Structure d'une feuille de style

```
1 body {color:rgb(183,0,183); margin: 1.5cm;}
2 h1, h2 {
3   color: aqua;
4   text-align: center;
5 }
6 p {
7   background-color: yellow;
8 }
```

- Suite de règles ;
- Une règle =

```
Sélecteur {
  propriété : valeur ;
  ...
  propriété : valeur ;
}
```

## Les sélecteurs

Définissent à quels éléments un style est appliqué.

```
1 body {color:rgb(183,0,183); margin: 1.5cm;}
2 h1, h2 {
3   color: aqua;
4   text-align: center;
5 }
6 p {
7   background-color: yellow;
8 }
```

- les éléments du body sont en violet avec une marge de 1,5cm
- les paragraphes ont *en plus* une couleur de fond jaune
- les titres h1 et h2 sont *en plus* centrés et leur texte est en bleu clair

Certaines propriétés comme les fontes ou la couleur sont héritées par les éléments descendant.

## On peut sélectionner...

### une classe

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

### une balise d'une certaine classe

```
p.important {color: blue;}
```

tous les `em` dans un paragraphe seront en gris

`p.important {color:green;}` Tous les éléments de classe `important` à l'intérieur d'un paragraphe seront verts.

## On peut sélectionner ...

`.resume em {color:yellow;}` Tous les `em` descendants d'un élément de classe `resume` seront jaunes.

### un sélecteur enfant d'un autre sélecteur

`resume>em {color:yellow;}`

### un sélecteur frère immédiat d'un autre sélecteur

`h5+p {color:pink;}`

## Attribut id et feuille de style

```
1 | p#resume {color :green;}
2 | #toto {color :blue;}
```

?

et dans la page HTML :

```
1 | <h1 id="toto">Celui-ci est bleu </h1>
2 | <p id="resume"> celui-là est vert </p>
```

?

Rappel : deux éléments dans la même page ne peuvent avoir le même id.

## Sélecteur d'attributs général

Pour sélectionner un élément sur la valeur d'un attribut quelconque :

[ ATTR=VALEUR ]

Exemple:

```
1 | * [class="important"] {color :red;}
2 | p [class="remarque"] {color :red;}
3 | * [lang="fr"] {color :red;}
```

?

- `"*"` : n'importe quelle balise
- fonctionne sur n'importe quel attribut
- pas de valeur précisée  $\Rightarrow$  sélectionne n'importe quelle valeur précisée

## Remarque

`.toto` et `*[class="toto"]` sont similaires mais...

<p class="titi toto"> un paragraphe;...</p>

est reconnu par :

```
1 | .toto {color: red;}
2 | *[class="titi toto"] {color: red;}
```

?

mais pas par :

```
1 | * [class="toto"] {color: red;}
```

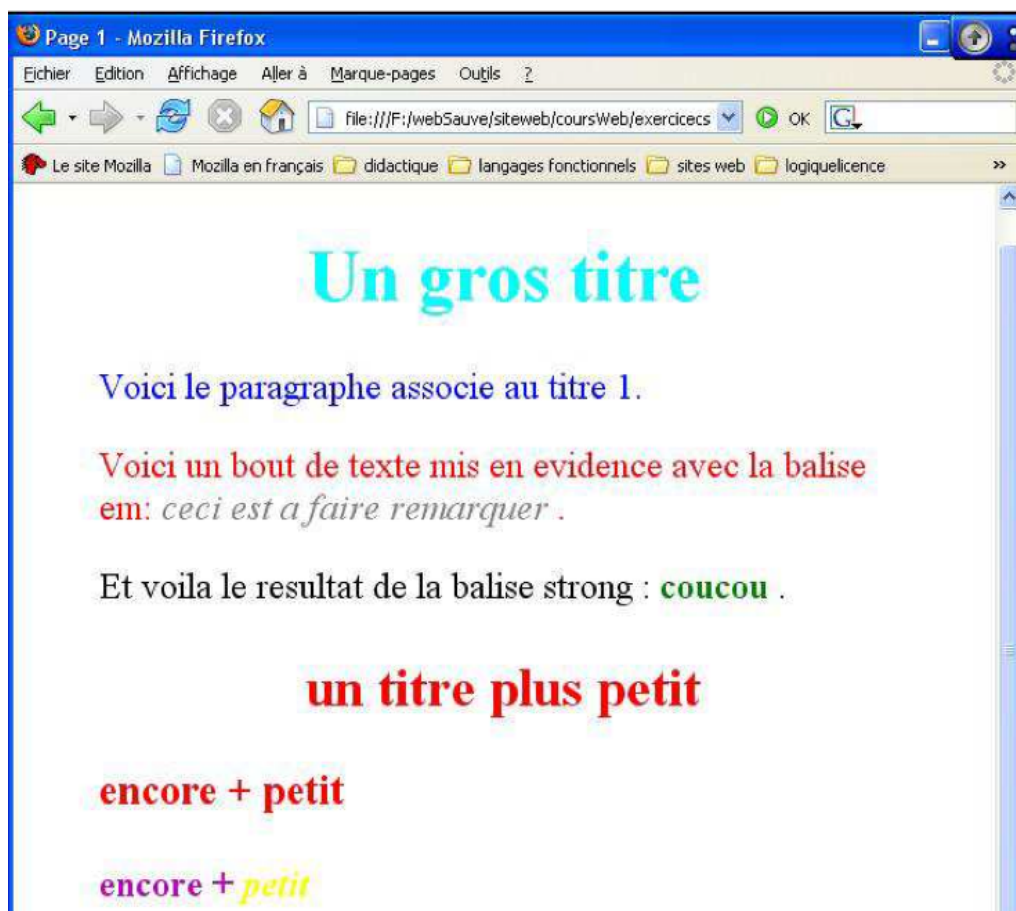
?

## Exercice

```
<h1>Un gros titre</h1>
<p class="bleu">Voici le paragraphe
  associe au titre 1.
</p>
<p class="important">Voici un bout de texte
  mis en evidence avec la balise em:
  <em> ceci est a faire remarquer </em>. </p>
<p> Et voila le resultat de la balise strong :
  <strong class="important"> coucou </strong>.
</p>
<h2 class="important">un titre plus petit</h2>
<h3 class="important">encore + petit</h3>
<h4 class="resume">encore + <em>petit</em></h4>
<h5>encore + petit</h5>
<p>toto</p>
```

```
body {
  color:rgb(183,0,183); /* violet */
  margin:1.5cm;
}
h1,h2{
  color:aqua;
  text-align:center;
}
p{color:black;}
.important {color:red;}
p.bleu {color:blue;}
p em {color:gray;}
p .important {color:green;}
.resume em {color:yellow;}
h5 + p {color:lime;}
```

## Exercice



## div, span et feuilles de style

```
1 body { color:rgb(183,0,183); }
2 h1,h2{ color:aqua; }
3 .important {background-color:grey;}
4 .important h1 {color:yellow;}
5 .important p {color:aqua;}
```

?

```
1 <h1>chapitre 2: les feuilles de style</h1>
2 <div class="important">
3   <h1> Les selecteurs </h1>
4   <p>Différents types de selecteurs</p>
5 </div>
6 <h1> Les propriétés </h1>
7 <p>bla bla bla</p>
```

?

## Autres moyens d'introduire un style

Il existe deux autres manières d'introduire le style. On préfère normalement des feuilles de styles indépendantes.

- dans le document HTML lui-même :

```
1 <head>
2   <title> un beau titre</title>
3   <style type="text/css">
4     h1 {text-align: centered;}
5   </style>
6   ...
```

- comme attribut d'une balise ouvrante ; s'applique à cette balise-là uniquement.

```
1 <p style="text-align: centered; background-color: red;">...</p>
```

*On ne met que les propriétés dans ce cas, pas de sélecteur*

## Contenu d'un style (les propriétés)

Très nombreuses propriétés. Nous allons en voir quelques-unes. webographie :

### Couleur

Trois façons de les désigner :

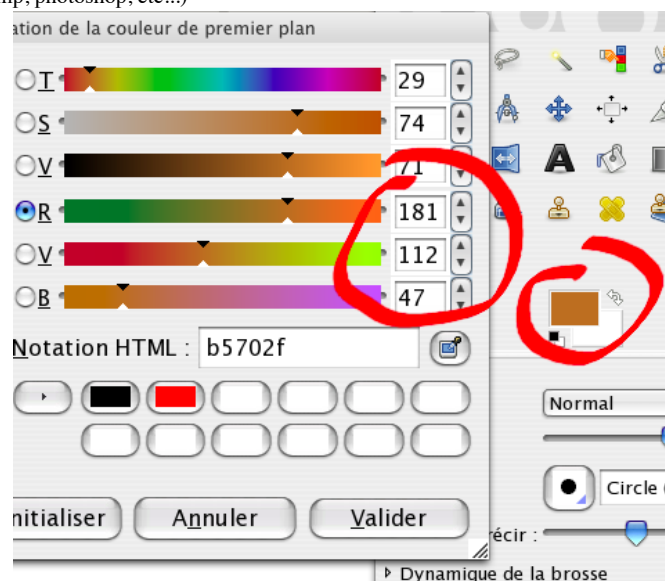
- par nom : 16 noms prédéfinis white, silver, gray, black, red, maroon, lime, green, yellow, blue, navy, fushia, purple, aqua, teal
- schéma RGB : trois valeurs, entre 0 et 255 pour le rouge, le vert, le bleu :

rgb(181, 112, 47)

ou trois pourcentages :

rgb(80%, 10%, 20%)

voir les logiciels de dessin (gimp, photoshop, etc...)



- en hexadécimal (six chiffres hexadécimaux, deux par composante). De #000000 (noir) à #FFFFFF (blanc).

# Propriétés de couleur

## color

fixe la couleur du *texte*

```
1 | p {color: rgb(0,10,200);}
```

?

## background-color

fixe la couleur du fond.

```
1 | p {color: red; background-color: yellow;}
```

?

(texte rouge sur fond jaune).

# Polices de caractères

- groupées en *familles* : même auteur, aspect compatible entre les variations de la police (exemple : Garamond, Times)
- dans une même famille, plusieurs *styles* disponibles : normal, *italique*, *oblique*
- plusieurs graisses disponibles : normal, *fi* et **gras**

# Polices de caractères

Problèmes divers, en particulier de droits.

Le navigateur client n'a pas forcément la police demandée

- utiliser des polices courantes
- proposer plusieurs polices
- proposer une famille comme dernier recours

## font-family

```
p {font-family: helvetica, verdana,sans-serif;}
```

- On donne une liste de noms de polices. La première à être disponible est utilisée
- Si espaces dans un nom : mettre des guillemets :

```
p {font-family: Georgia, "Times New Roman", Times, serif;}
```

- trois familles de base : Serif, Sans-Serif et Monospace.

Normalement, la dernière famille est l'une des trois familles de base.

# Autres caractéristiques des polices

### font-style

valeur : italic, oblique ou normal

### font-weight

valeur : bold, bolder, lighter, normal

### font-size

- valeur numérique absolue: 12pt, 10mm, 1cm, 10px
- valeur numérique relative 1.5em ou 150% (normal=1em ou 100%)
- taille relative prédéfinie :xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger

### text-decoration

valeur underline, overline, line-through (barré) none

### text-align

alignement horizontal du texte : left, right, center, justify

# Les boîtes

L'affichage est composé de *boîtes* contenant le texte. Chaque boîte est composée de plusieurs parties : le contenu (le texte lui-même), l'espace

(margin)

(padding)

le contenu le contenu le contenu le contenu le contenu le contenu le contenu le contenu le contenu le contenu le  
contenu le contenu le contenu le contenu le contenu le contenu le contenu le contenu le contenu le contenu le  
contenu le contenu le contenu le contenu le contenu le contenu le contenu le contenu le contenu le contenu le  
contenu le contenu le contenu le contenu le contenu le contenu

?

```
p {border:medium solid black;}  
h1 {border:4px dotted green;}
```

Le style peut être (entre autres)

none

dotted

dashed

solid

double

groove

ridge

inset

outset

CSS3 introduit border-radius et border-image pour des bordures plus complexes.

```
solid, border-radius: 20px;
```

a aussi border-top, border-bottom, border-left, border-right

## Propriétés de listes

list-style-type : apparence des signes d'une liste. *decimal*

1.,2.,3.,4. etc...

**lower-roman**

i.,ii.,iii.,iv. etc.

**upper-roman**

I.,II.,III.,IV. etc...

**lower-alpha**

a.,b.,c.,d. etc.

**upper-alpha**

A.,B.,C.,D. etc...

**disc**

rond plein comme puce

**circle**

puce ronde

**square**

puce rectangulaire

**none**

pas de puce, pas de numérotation

list-style-image

```
list-style-image:url([nom de fichier])
```

Vous permet de mentionner un graphique pour vos propres puces.

## Règles de résolutions de conflits

supposons que nous ayons :

```
h1 {color: blue; font-size: large;}  
.titre {color:red;}
```

Comment se présente donc

```
<h1 class="titre">un titre</h1>
```

- Les deux sélecteurs s'appliquent ;
- pour la taille, pas de conflit : large
- pour la couleur, intuitivement : le sélecteur le *plus spécifique* s'applique. C'est `.titre` ⇒ texte en rouge



## Règles de résolutions de conflits

- Pour des CSS de même origine, la priorité se fait selon la spécificité des règles.
- Pour une même spécificité, on choisit la dernière règle spécifiée.

### Calcul de la spécificité

On construit un triplet d'entiers :

1.  $n1$  = le nombre d'ids d'attributs dans le sélecteur
2.  $n2$  = le nombre d'attributs et de classes dans le sélecteur
3.  $n3$  = le nombre de noms d'éléments et de pseudo-éléments (`h1`, `h2`, `em`, `p`, `:first-line` ...) dans le sélecteur.

## Règles de résolutions de conflits

On compare les triplets par ordre *lexicographique* et le plus grand est le plus spécifique.

$(a1, b1, c1) > (a2, b2, c2)$  si :

- $a1 > a2$
- ou  $a1 = a2$  et  $b1 > b2$
- ou  $a2 = a2$  et  $b1 = b2$  et  $c1 > c2$

Exemples :

- `p em` : spécificité 0,0,2
- `.important p` : spécificité 0,1,1

En cas d'égalité, la dernière règle spécifiée l'emporte.

## Positionnement et feuilles de styles

Problème : obtenir une mise en page complexe

- au départ, les CSS sont assez pauvres dans ce domaine.
- plusieurs mécanismes en concurrence
- pas mal de bricolage

à la base, mise en page de base : composition de boîtes de la largeur de l'écran. Empilement des boîtes en question

### Titre

Paragraphe 1 Paragraphe

1 Paragraphe 1

Paragraphe 1

Paragraphe 2 Paragraphe

12 Paragraphe 2

Paragraphe 2 Paragraphe

2

- liste ...
- liste...

## Positionnement

Réalisé par plusieurs propriétés :

### *display*

comportement général de l'élément : bloc, en-ligne, etc....

### *position*

manière dont l'élément se positionne par rapport aux autres ou à la page

### *top, bottom, left, right*

position de l'élément par rapport à son point de repère (fixé par *position*)

*display* (block, inline, none, etc.)

*position* (static | relative | absolute | fixed)

$$\mathbf{display}$$

- ### Valeurs possibles

l'élément n'est pas affiché, mais la mise en page tient compte de sa taille.

## Position

## Emplacement

- ## Positionnement relatif

## Positionnement absolu

NFA016 : CSS

Au départ: positionnement des illustrations dans le texte... comme ça :

[illegible]

Rapidement détourné pour faire de la mise en page plus complexe

- left : flotte à gauche. Le texte contourne l'élément.
- right: flotte à droite. Le texte contourne l'élément.

Les flottants ne sont pas pris en compte dans le calcul de la taille du parent

```
1 <div style="border: 1px solid;"> un essai.  
2    Vous voyez, ça ne  
4   tient pas.  
5 </div>
```

Propriété qui permet d'interdire la présence d'un flottant d'un côté ou de l'autre d'un élément. On déplace l'élément après les flottants concernés.

- right : pas de flottant à droite
- left: pas de flottant à gauche
- both: pas de flottant du tout

```
1 <div style="border: 1px solid;"> un essai.  
2     
4   <p style="clear:both;">Avec "clear", on saute après le flottant</p>  
5 </div>
```

On met parfois un div vide avec "clear: both" pour forcer l'élément à s'agrandir pour contenir les flottants.

Un essai.

Float permet de créer une liste d'éléments complexes côte à côte.

légende	légende	légende	légende	légende
---------	---------	---------	---------	---------

```
1 <div style="border: 1px solid black; padding: 3px; float:left;"> ?
2    <p> légende</p></div>
3 <div style="border: 1px solid black; padding: 3px; float:left;">
4    <p> légende</p></div>
5 <div style="border: 1px solid black; padding: 3px; float:left;">
6    <p> légende</p></div>
7 <div style="border: 1px solid black; padding: 3px; float:left;">
8    <p> légende</p></div>
9 <div style="border: 1px solid black; padding: 3px; float:left;">
10   <p> légende</p></div>
```

autre solution pour le même effet : inline-block (mais "certains" navigateurs le gèrent mal)



```
1 <div style="border: 1px solid black; padding: 3px; display:inline-block;"> ?
2    <p> légende</p></div>
```

## Combinaison de plusieurs propriétés

Généralement, utilisation simultanée de float, position, margin, etc...

### Exemple

Menu en float à gauche. Largeur fixeLe contenu

Le contenu. Marge de 16 em.