

# Langages de l'Internet

<https://mavromatis.org>

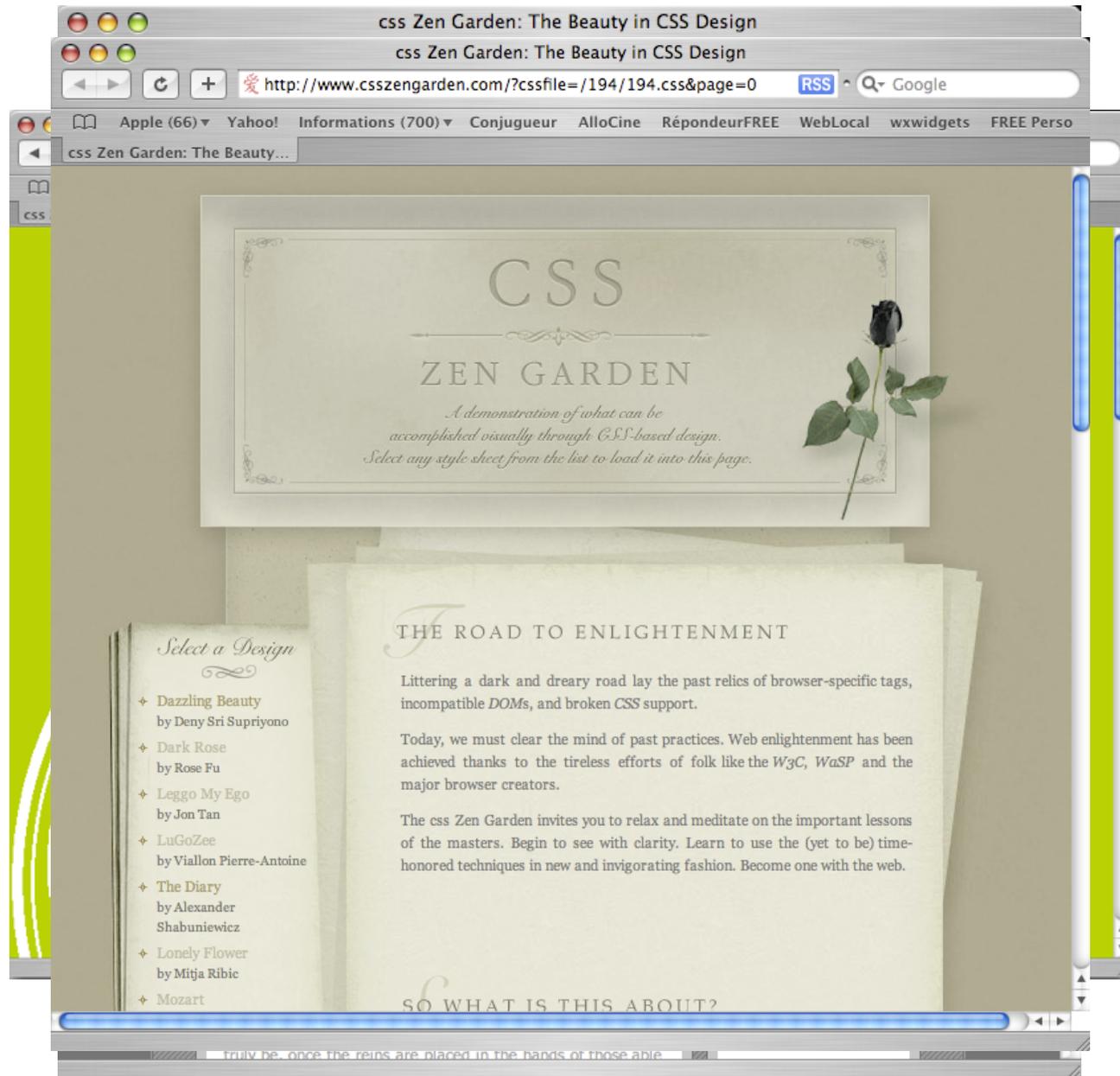
# Les feuilles de style

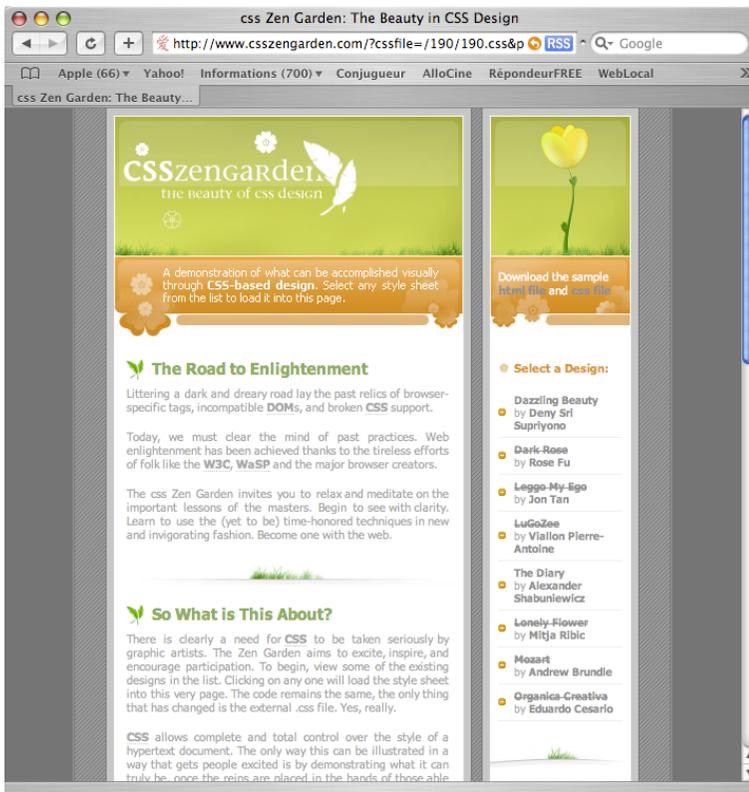
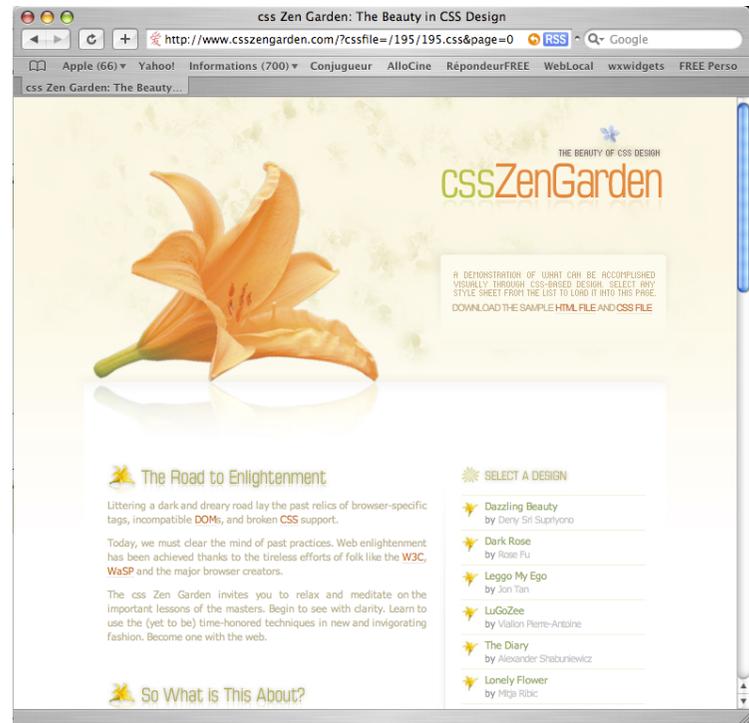
## CSS : Cascading Style Sheet

- Recommandation du W3C
- CSS1 (1996), CSS2 (1998), CSS2.1 (2005), CSS3  
Support par les navigateurs inégal
- Principe :
  - [X]HTML décrit la **structure** et le **contenu**
  - CSS décrit la **mise en forme**

# Pour une démonstration

- <http://www.csszengarden.com/>





# Les feuilles de style

## CSS : Cascading Style Sheet

- *Associer une feuille de style à une page*
- *Syntaxe CSS*
- *Propriétés CSS*
- *Appliquer le style (class)*
- *Sections <div> et <span>*
- *Positionnement absolu et relatif (CSS 2)*

# Utilisation d'une feuille de style

- Différentes manières d'utiliser une feuille de style à une page HTML :

- Faire un lien vers une feuille de style externe (fichier texte avec l'extension .css) :

```
<head>  
  <title>titre de la page</title>  
  <link rel="stylesheet" href="style.css" type="text/css">  
</head>
```

- Définir le style dans l'en-tête de la page :

```
<head> ..... </head>
```

- Utiliser l'attribut STYLE :

```
<div style="font-size: 12px;color: green">
```

# Utilisation d'une feuille de style

- Plusieurs feuilles de style peuvent s'appliquer simultanément :
  - Plusieurs balises `<link rel="stylesheet">`
  - Directive `@import` d'une feuille de style `@import url(feuille_annexe.css);`
  - Feuille de style de l'utilisateur (Mozilla, Opera...)
- Au sein même d'une feuille de style, plusieurs règles peuvent être en conflit.
- **Cascade** : mécanismes gérant les conflits.
  - Si `!important` est précisé après la valeur, la règle sera prioritaire.
  - Sinon, plus une règle est spécifique, plus elle est prioritaire.
  - Sinon, la dernière règle s'applique.

# Syntaxe CSS

- Utilisation de **sélecteurs**
  - Sélecteur simple : nom d'une balise.
  - Sélecteur multiple : plusieurs sélecteurs séparés par des virgules.
  - Sélecteur universel : \*, sélectionne tout.
- Par exemple :
  - `ul { color: blue; }` met l'ensemble du contenu des listes à puces en bleu.
  - `h1,h2,h3,h4,h5,h6 { color: red; }` met l'ensemble des titres en rouge.
  - `* { color: black; }` met tout en noir. Dans ce cas précis, on préférera une règle `body { color: black; }`

# Syntaxe CSS

- Utilisation de sélecteurs (suite)
  - Sélecteur de classe : nom d'une classe, préfixée d'un .
- Par exemple :
  - .personne { font-face: bold; } met l'ensemble des balises de classe "personne" en gras.
  - p.comment { font-style: italic; } met l'ensemble des balises "p" de classe "comment" en italique.
  - Utilisation de "personne"

```
<p>Je voudrais particulièrement remercier  
<span class="personne">Madame Machin</span> et  
<span class="personne">Monsieur Bidule</span>.</p>
```

# Syntaxe CSS

- Utilisation de sélecteurs (suite)
  - Sélecteur contextuel : 2 sélecteurs ou plus séparés par des espaces. A B sélectionne les B seulement s'ils sont contenus dans des A.
- Par exemple :
  - `h1 em { color: blue; }` met les mots en emphase à l'intérieur d'un h1 en bleu.
  - `ul ol, ol ul, ul ul, ol ol { font-size: 80%; }` diminue la taille du texte des listes imbriquées.

# Syntaxe CSS

- Pseudo-classes : permet de “sélectionner” un élément dans certains contextes.
  - a:link : les a qui sont des liens.
  - a:visited : les a qui sont des liens vers des pages qui ont déjà été visitées.
  - a:hover : les a qui sont des liens qu’on est en train de désigner (i.e. la souris est en train de passer dessus).
  - a:active : les a qui sont des liens qu’on est en train d’activer (i.e. de cliquer).
- Priorités => Cascade

# Propriétés CSS

- Les propriétés admettent différentes unités de mesure se répartissant suivant trois types de valeurs :
  - Les valeurs fournies sous forme de pourcentage (par rapport à la valeur courante)
  - Les valeurs relatives :
    - **em** la valeur de la propriété font-size de la police utilisée (largeur du caractère m)
    - **ex** la hauteur du caractère x dans la police utilisée
  - Les valeurs relatives à l'écran
    - **px** le nombre de pixels
  - Les valeurs absolues (pas de sens à l'écran !)
    - mm millimètre
    - cm centimètre (1cm= 10mm)
    - in inch (1in= 2.54cm)
    - pt point (1pt= 1/72in)
    - pc pica (1pc= 12pt)

# Propriétés CSS

- Il existe différents moyens pour contrôler les polices de caractères.
- Les principales propriétés (font-family, font-size, font-weight et line-height) seront responsables de l'aspect de votre texte.
- Par exemple (paragraphe en Times 13pt avec un espace interligne de 10pt)

```
p {  
  font-family: times;  
  font-size: 13pt;  
  font-weight: bold;  
  line-height: 10pt;  
}
```

# Propriétés CSS

- **font-family** spécifie la police ou famille de police (serif, sans-serif, cursive, fantasy, monospace) à utiliser par le navigateur. Il est recommandé de déclarer une famille générique après les polices (séparer par une virgule) pour obliger le navigateur si la police n'est pas trouvée à lui substituer une police similaire.
- **font-size** définit la taille de la police (exprimée en points, pourcentage, taille absolue ou relative, pixels).
- **font-style** définit le style de police à utiliser à savoir italic, oblique ou normal.
- **font-variant** définit la casse des caractères : normal ou small-caps (petites majuscules).
- **font-weight** définit la graisse des caractères : normal, bold (gras).

# Propriétés CSS

- **line-height** définit l'espace interligne ou la hauteur intrinsèque de tout élément non remplacé. Une bonne règle pour le texte dans le corps est de régler l'espace interligne à 1.5 fois la taille de la police (1.5em ou 150%).
- **text-decoration** définit l'ornement du texte : none, underline (souligné), overline (surligné), blink (clignotant) et line-through (barré).
- Les propriétés **letter-spacing** et **word-spacing** permettent de définir l'espacement entre les lettres d'un mot et les mots dans une phrase.
- Cet espacement peut être positif ou négatif selon si on cherche à éloigner ou rapprocher ces éléments.
- Comme les couleurs, ces propriétés doivent être utilisées avec discernement.

# Propriétés CSS

- **text-align** définit la justification et l'alignement d'un texte : left, right, center et justify (aligne le texte sur les marges gauche et droite).
- **vertical-align** définit l'alignement vertical de la ligne de base d'un élément en fonction de la propriété line-height : super (texte en exposant), sub (texte en indice), baseline (normal), etc.
- **text-indent** permet de décaler le début de la première ligne de texte d'une valeur fixe ou proportionnelle à la valeur du paragraphe.
- **list-style-image** définit l'image qui sera employée comme marqueur d'item de liste. Quand l'image est indisponible, elle remplace le marqueur produit par la propriété list-style-type.
- **list-style-type** définit l'apparence d'une liste numérotée ou non : disc (cercle plein), circle (cercle vide), square (carré), decimal (nombres), upper-alpha ou lower-alpha (lettres de l'alphabet), upper-roman ou lower-roman (chiffres romains) et none.

# Propriétés CSS

- Les propriétés qui s'appliquent aux bordures ont entre 1 et 4 valeurs :
  - Si une seule valeur est spécifiée : elle s'applique à tous les côtés.
  - Si 2 valeurs sont spécifiées :
    - la première s'applique aux bordures horizontales
    - la seconde aux bordures verticales
  - Si 3 valeurs sont spécifiées :
    - la première s'applique à la bordure du haut
    - la deuxième aux bordures verticales
    - la troisième à la bordure du bas
  - Si 4 valeurs sont spécifiées : celles-ci s'appliquent respectivement aux bordures du haut, de droite, du bas et de gauche.
- **border-style** définit le style des bordures : none, dotted, dashed, solid, double, groove, ridge, inset ou outset.
- **border-color** définit la couleur des côtés de la bordure.
- **border-width** spécifie l'épaisseur des bordures.

# Propriétés CSS

- La **couleur** doit être utilisée avec **précaution**.
- Son rôle est de différencier les éléments d'un texte en jouant sur la notion de contraste.
- Il existe différentes manières de spécifier les valeurs d'une couleur :
  - En indiquant le nom d'une couleur prédéfinie parmi aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white et yellow
  - En utilisant le code RGB pour (Red, Green, Blue) avec des valeurs de 0 à 255
  - En utilisant la valeur hexadécimale
- <http://www.colorschemer.com/online.html>

# Propriétés CSS

- La propriété couleur s'applique généralement au texte (`color`) ou à l'arrière-plan (`background-color`).
- La propriété `background-image` définit l'image d'arrière-plan d'un ou plusieurs éléments ou encore de tous les éléments. Il est fortement recommandé de spécifier aussi une couleur normale d'arrière-plan, qui sera utilisé en remplacement.
- Par exemple :

```
background-image: url("mesImages/monImage.jpg");
```

```
background-color: white;
```

# Les blocs

- En XHTML, deux sortes d'éléments
  - Les blocs : `<p>`, `<h1>`, `<ul>` ...  
`<div>` comme bloc générique
  - Les éléments "en ligne" qui se placent à l'intérieur des blocs : `<a>`, `<img>` ...  
`<span>` comme élément générique
- Comment placer les blocs les uns par rapport aux autres ?

margin (marge)

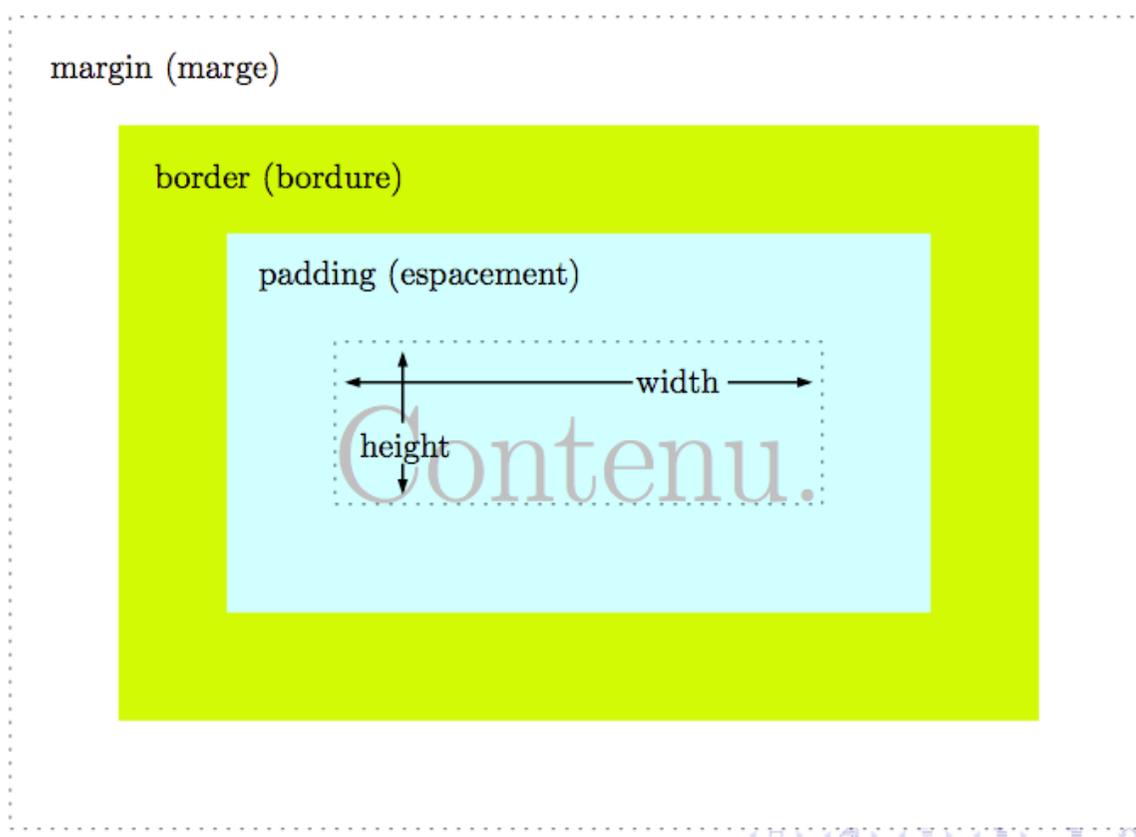
border (bordure)

padding (espacement)



# Largeur, Hauteur, Débordement

- width : largeur du contenu de l'élément. Par défaut à "auto", prend la taille maximale disponible.
- height : hauteur du contenu de l'élément. Par défaut à "auto", s'adapte à la taille du contenu.
- Si une taille est contrainte, un contenu peut déborder !
  - La propriété overflow contrôle le comportement:
    - visible : le contenu déborde de l'élément (par défaut)
    - hidden : le contenu est tronqué
    - scroll : des barres de défilement sont ajoutées
    - auto : des barres de défilement sont ajoutées si nécessaire



- padding-top
- padding-right
- padding-bottom
- padding-left

Par défaut à zéro

- margin-top
- margin-right
- margin-bottom
- margin-left

Par défaut à zéro  
Il existe "auto"

# Flottement et Positionnement

- Mise en page complexe : colonnes, chevauchement d'éléments ...
- Positionnement : position relative par rapport à un autre élément.
- Flottement : les éléments sont “pris dans le flux”, les autres éléments les “contournent”

# Flottement

- Propriété **float** :
  - left : l'élément est contourné par la droite
  - right : l'élément est contourné par la gauche
  - none : valeur par défaut (écrasement)
- Tout bloc peut être flottant.
- Une dizaine de règles régissent le comportement.

# Dégagement

- Propriété **clear** :
  - **left** : l'élément se positionne en-dessous de tout élément flottant à gauche
  - **right** : l'élément se positionne en-dessous de tout élément flottant à droite
  - **both** : l'élément se positionne en-dessous de tout élément flottant
  - **none** : défaut

# Positionnement

- Propriété **position** :
  - **static** : l'ordre des déclarations dans le code est respecté (par défaut)
  - **relative** : distance “dans la page”
  - **absolute** : retire l'élément du flux. Position par rapport à son conteneur
  - **fixed** : retire l'élément du flux. Position par rapport à la fenêtre d'affichage
- Le plus souvent : absolu par rapport à la page
  - `body { position: relative; }`

# Visibilité

- Propriété **visibility** :
  - visible
  - hidden : l'élément est invisible mais la "mise en page" continu à le prendre en compte.

# Tables

- Les bordures
  - fusionnées : `border-collapse : collapse`
  - séparées : `border-collapse : separate`  
`border-spacing` précise l'espacement entre deux bordures
- Cellules vides : propriété `empty-cells`
- Largeur et hauteur : `width` et `height`

# Tables

## CSS

```
table {  
border-style:solid;  
border-width:1px;  
border-color:black;  
width:50%;  
}  
td {  
border-style:solid;  
border-width:1px;  
border-color:red;  
width:50%;  
}
```

## HTML

```
<table>  
  <tr>  
    <td>  
      <p>Oh qu'il est beau !</p>  
    </td>  
    <td>  
      <p>  
        Ce joli tableau à traits fins !!  
      </p>  
    </td>  
  </tr>  
</table>
```

Oh qu'il est beau !	Ce joli tableau à traits fins !!
------------------------	---

# Tables

## CSS

```
table {  
border-collapse : collapse;  
border-style:solid;  
border-width:1px;  
border-color:black;  
width:50%;  
}  
td {  
border-style:solid;  
border-width:1px;  
border-color:red;  
width:50%;  
}
```

Oh qu'il est beau !	Ce jouli tableau à traits fins !!
---------------------	-----------------------------------

Oh qu'il est beau !	Ce jouli tableau à traits fins fusionnés...
---------------------	---

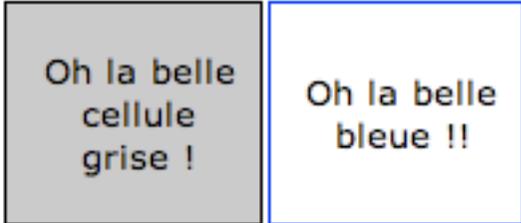
# Tables

## CSS

```
.grise {  
border-style:solid;  
border-width:1px;  
border-color:black;  
background-color:silver;  
}  
  
.blue {  
border-style:solid;  
border-width:1px;  
border-color:blue;  
}  
  
p {  
background-color:transparent;  
color:black;  
}
```

## HTML

```
<table class="none">  
  <tr>  
    <td class="grise">  
      <p>  
        Oh la belle cellule grise !  
      </p>  
    </td>  
    <td class="blue">  
      <p>  
        Oh la belle bleue !!  
      </p>  
    </td>  
  </tr>  
</table>
```



Oh la belle  
cellule  
grise !

Oh la belle  
bleue !!

# Un exemple ...

- Les balises `<div>` et `<span>` permettent d'appliquer localement un style sans tenir compte du style de la balise courante
  - `div` est utilisé pour des blocs de texte
  - `span` sert à des portions de texte

```
<body>
  <p class="stylebase">
    Texte marron
    <span class="vertgras"> texte vert et gras </span>
    Texte marron
  </p>
</body>
```