

# Introduction aux technologies WEB



**Pr. Hajar IGUER**  
**[hajar.iguer@uic.ac.ma](mailto:hajar.iguer@uic.ac.ma)**



# CSS : CASCADING STYLE SHEET I



# Historique

- CSS a été introduit avec HTML 4, afin de fournir une meilleure façon de mettre en forme les éléments (balises) HTML.
- CSS peut être ajouté au code HTML à l'aide de différentes manières
- Le meilleur moyen d'ajouter CSS pour HTML, est de mettre la syntaxe CSS dans des fichiers CSS séparés.



# Insertion CSS dans une page Web

- **Inline:** en utilisant l'attribut de style des balises HTML
  - `<p style="color:blue;margin-left:20px;">This is a paragraph.</p>`
- **Interne:** en utilisant l'élément `<style>` dans la section `<head>`
  - `<style>`  
body {background-color:yellow;}  
p {color:blue;}  
`</style>`
- **Externe:** à l'aide d'un fichier CSS externe
  - `<head>`  
`<link rel="stylesheet" type="text/css" href="mystyle.css">`  
`</head>`



# HTML/CSS

- L'essentiel est de séparer le contenu (codé en HTML) de la mise en forme (feuilles de style cascadées -CSS).
- Cette méthode présente plusieurs avantages, notamment la clarté du code et la possibilité de définir des styles communs à plusieurs pages.



# Intérêt des feuilles de Style Nous innovons pour votre réussite !

- L'utilisation des feuilles de style n'a pas pour seul but de répondre aux normes du W3C.
- La dissociation du contenu (HTML) et de la mise en forme (feuille de style) permet :
  - de retrouver et corriger plus facilement le texte des pages ;
  - d'utiliser une feuille de style externe, commune aux différentes pages d'un site. Il en résulte une meilleure unité graphique entre ces pages et aussi des mises à jour plus simples par la suite. Une modification dans la feuille de style externe se répercute d'un seul coup sur toutes les pages du site.



# Avantages

- La mise en page est beaucoup plus légère, car elle ne nécessite plus l'utilisation de tableaux. Les CSS permettent en effet de positionner les différentes parties d'une page web :
  - **soit de façon rigoureuse** : blocs fixes dont les coordonnées sont choisies ;
  - **soit d'une manière souple** : blocs flottants qui s'alignent les uns par rapport aux autres



# Structure d'un fichier css externe

- Créez un fichier texte avec une extension .css et mettez-en le code suivant:

Balise1

```
{  
Propriété1 : valeur1;  
Propriété2 : valeur2;  
}
```

Balise2

```
{  
    propriete1: valeur1;  
}  
/*
```

Insérez un commentaire

```
*/
```





# Structure d'un fichier css

Nous innovons pour votre réussite !

- Exemple:

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



# Attributs spéciaux

- Afin de différencier entre les styles d'un même paragraphe, on peut utiliser des attributs spéciaux:
  - l'attribut **class** ;
  - l'attribut **id**: ne peut être utilisé qu'une seule fois dans le code ainsi que pour des balises uniques
- Ce sont des attributs que l'on peut mettre sur n'importe quelle balise, aussi bien titre que paragraphe, image, etc.



# Structure d'un fichier html/css

Nous innovons pour votre réussite !

- En utilisant les attributs spéciaux, on écrit dans la page HTML:
  - `<p class=" test "> .....</p>`
  - `<img src= " " id= "essai " >`
- Et dans le fichier CSS
  - **.test**
  - {
  - **color:black;**
  - }
  - **#essai**
  - {
  - **Color: red;**
  - }



# Structure d'un fichier html/css

- Afin de faire valoir des propriétés de style à un mot qui n'est pas entouré par une balise, il faut utiliser les deux balises:
  - `<span>....</span>`
  - `<div>.....</div>`
- Exemple:
  - La partie HTML:
    - `<p>Bonjour et <span class= "test">bienvenue</span> </p>`
  - La partie CSS:
    - `.test`
    - `{`
    - `color: blue;`
    - `}`



# L'héritage en CSS

- Lorsqu'on applique un style à une balise , toutes les sous balises (à l'intérieur) prennent le même style.
- Par exemple si on donne une couleur précise à un paragraphe, tous les paragraphes de la page web s'écriront avec la même couleur.



# Propriétés de style

- Exemples:
  - Aligner un texte:
    - `style="text-align:center; "`
  - Changer la couleur du fond
    - `style="background-color:green;`
  - La Police, Couleur et taille d'un texte
    - `style="font-family:arial;color:red;font-size:20px;"`



# Propriétés de style

- font-size: 16px; ou font-size : xx-small;
- Vous pouvez remplacer « px » par « cm » ou « mm ».
- Vous pouvez aussi indiquer la taille en em.
- Exemple:
  - p
  - {
  - font-size: 0.8em;
  - }



# Propriétés de style

- Modifier la police: p

```
p
```

```
{
```

```
font-family: " Times New Roman ", "Arial Black",  
Georgia, Verdana, sans-serif;
```

```
}
```

- Texte en italique: valeurs(italic ,oblique,normal )

```
em
```

```
{
```

```
font-style: normal;
```

```
}
```





# Propriétés de style

- Aligner un texte en gras:

```
h1
```

```
{ text-align: justify;  
  font-weight: bold;  
  text-decoration: overline;  
}
```

- Autoriser la coupure de mots ou de paragraphes:

```
p
```

```
{  
  word-wrap: break-word;  
}
```



# Propriétés de style

- **Color: teal;**
- **color: #FFFFFF;** : Couleur Blanche
- **color: #000000;** : Couleur Noire
- **color: rgb(240,96,204);** : Couleurs comprises entre 0 et 255, Vous pouvez utiliser le logiciel « Boite à couleurs » pour déterminer le code d'une couleur.
- **background-color: grey; /\*** La couleur fond de la page sera grise **\*/**



# Propriétés de style

- Surligner un mot en une couleur:

```
<mark>texte</mark>
```

```
Mark{
```

```
Color : black;
```

```
Background-color: blue;}
```

- Mettre une Image de fond et la fixer:
  - background-image: url(" image.png");
  - background-attachment: fixed; ou scroll;
  - background-repeat : no-repeat;
  - background-position: 30px 50px; ou top left;
  - opacity: 0.6;



# Propriétés de style

- Pour que le texte ne dépasse pas des limites du paragraphe, il va falloir utiliser la propriété `overflow`. Voici les valeurs qu'elle peut accepter :
  - **visible** (par défaut) : si le texte dépasse les limites de taille, il reste visible et sort volontairement du bloc.
  - **hidden** : si le texte dépasse les limites, il sera tout simplement coupé. On ne pourra pas voir tout le texte.
  - **scroll** : là encore, le texte sera coupé s'il dépasse les limites. Sauf que cette fois, le navigateur mettra en place des barres de défilement pour qu'on puisse lire l'ensemble du texte. C'est un peu comme un cadre à l'intérieur de la page.
  - **auto** : c'est le mode « pilote automatique ». En gros, c'est le navigateur qui décide de mettre ou non des barres de défilement (il n'en mettra que si c'est nécessaire). C'est la valeur que je conseille d'utiliser le plus souvent.



# Habillage d'un texte

- L'habillage d'un texte est une propriété qui permet de flotter un élément (balise). Deux possibilités sont permises:
  - `float: left` : l'élément flottera à gauche.
  - `float: right` : l'élément flottera à droite.
- Exemple
  - La partie HTML:

```
<p>Le texte qui va habiller l'image.</p>
```
  - La partie CSS:

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



# Stopper l'habillage

- Exemple
  - La partie HTML:  
`<p class= "texte_dessous">Ce texte est écrit sous l'image flottante.</p>`
  - La partie CSS:  
`.texte_dessous  
{  
 clear: both;  
}`
- Propriétés clear est ses valeurs sont:
  - **left** : le texte se poursuit en-dessous après un float: left;
  - **right** : le texte se poursuit en-dessous après un float: right;
  - **both** : le texte se poursuit en-dessous, que ce soit après un float: left; ou après un float: right;.

