



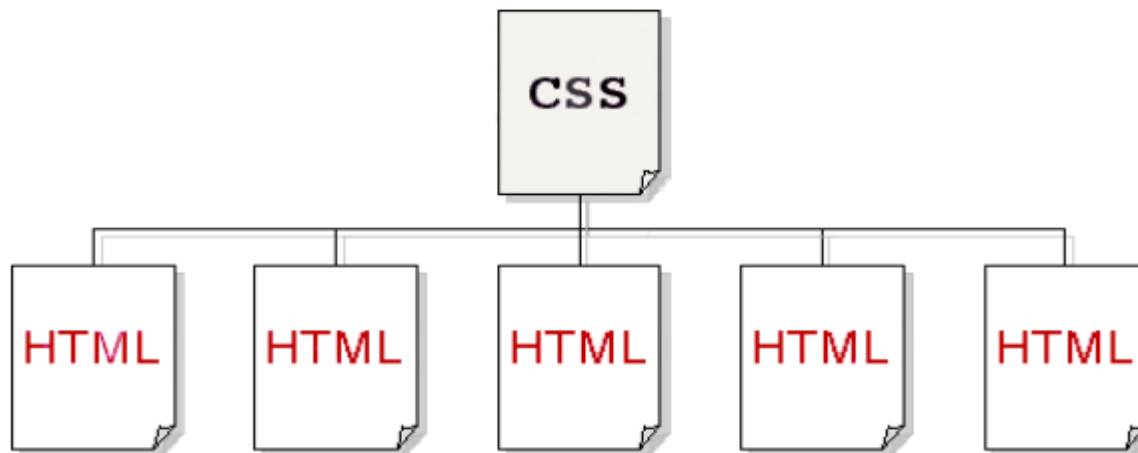
Les notions de bases CSS

420-JLA-JQ

Introduction au Web

CSS

- Cascading Style Sheets : Feuilles de Style en Cascade.
- C'est dans la feuille de style CSS que l'on va déclarer toute la mise en forme des pages : le positionnement des éléments, l'image de fond, les polices de caractère, les couleurs, etc.
- Celle-ci sera liée à chaque page html.
- Lorsqu'on en modifiera un élément, cela se répercutera immédiatement sur toutes les pages html. Puissant, pratique, bref : indispensable.



Sélecteurs, propriétés, valeurs

- Un sélecteur détermine à quel élément un style doit être appliqué.
- Une propriété détermine le style qui va être appliqué à un élément.
- Une valeur détermine le comportement d'une propriété.

```
sélecteur {  
    propriété1: valeur;  
    propriété2: valeur;  
}
```



Sélecteurs, propriétés, valeurs

- Exemple

sélecteur → p {
propriétés → color: blue; ← valeurs
 → font-size: 16px; ← valeurs
 }
 }

OÙ écrire le css?

3 possibilités :

- Dans l'élément head du document HTML,
- Dans la balise ouvrante d'un élément,
- Dans un fichier CSS séparé.

Il est recommandé d'utiliser la dernière méthode dans un fichier séparé.

- Pour lier un fichier HTML dans l'élément head avec un fichier CSS, on écrit :

```
<link rel="stylesheet" href="fichiercss.css">
```

Les commentaires en CSS

- Les fichiers CSS sont très vite très longs : il est donc essentiel de les commenter proprement !
- Un commentaire en CSS :
- `/* Je suis un commentaire CSS */`

Les sélecteurs simples

- On appellera sélecteurs simples les éléments HTML (ex : p)
- Ces sélecteurs doivent être préférés autant que possible car ils requièrent moins de code que des sélecteurs plus complexes et le code s'exécute donc plus vite.
- Limite : Comment appliquer un style différent à deux paragraphes ?

Sélecteur

- Simple: nom d'une balise

```
h1 {  
    color: black; /* met les titres h1 en noir */  
}
```

- Multiple: plusieurs balises séparées par virgule

```
h1, h2, h3, h4, h5, h6 {  
    color: red; /* met l'ensemble des titres en rouge */  
}
```

- Contextuel: plusieurs sélecteurs

```
h1 em {  
    color: blue; /* met les mots en emphase à l'intérieur d'un h1 en bleu */  
}
```


Class et Id

- Class et Id sont deux attributs HTML, créés pour pouvoir appliquer différents styles à des éléments de même type.
- Class permet également d'appliquer le même style à différents élément HTML (en leur appliquant la même valeur pour l'attribut class).
- Différence ? Id ne peut être utilisé que pour cibler un unique élément, au contraire de class.

Classe et Identifiant

- Classe

```
.person {  
    font-weight: bold;  
}
```

```
<div class="person">  
    <p>John Smith</p>  
</div>  
<p>John <span class="person">Doe</span></p>
```

- Identifiant

```
#monster {  
    color: green;  
}
```

```
<div id="monster">  
    <p>Shrek</p>  
</div>
```

DIV et SPAN

- Div et Span sont deux éléments HTML créés, entre autres, pour pouvoir appliquer un style à du contenu n'ayant pas de balise.
- Div et Span servent de containers mais ne possèdent aucune valeur sémantique.
- Containers permet de diviser un document pour regrouper des éléments
- Il peut être utilisé afin de grouper d'autres éléments pour leur appliquer un style (en utilisant les attributs id ou class).

Éléments de types block et inline

- Tout élément, en HTML, est soit un élément de type block, soit un élément de type inline.
- Div = élément de type block
- Span = élément de type inline

Éléments de types block et inline

Les éléments de type block...

- Commencent sur une nouvelle ligne,
- Occupent toute la largeur disponible,
- Peuvent être imbriqués les uns dans les autres et peuvent contenir des éléments de type inline.

Les éléments de type inline...

- Ne commencent pas sur une nouvelle ligne,
- Occupent seulement la largeur nécessaire,
- Peuvent contenir d'autres éléments de type inline mais pas d'éléments de type block.

Éléments de types block et inline

Éléments block	Éléments inline
p	em
h1, h2, h3...	strong
header, article, footer...	mark
ol, ul, dl	a
table	img