

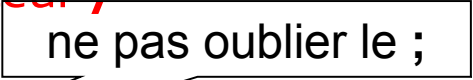
M1105 - CDIN - WEB

04 - Sélecteurs

Rappel

- Les styles sont définis entre { }
- par une suite de paires propriété:valeur ;

```
p {  
  color:blue ;  
  background-color:red ;  
}
```


- Ils sont appliqués selon ce qui précède l'accolade: sélecteur

Application à toutes les instances d'un élément

- Pour que le texte de tous les paragraphes en bleu :

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

- le contenu de toutes les balises <p> sera bleu, sauf si une autre couleur est attribuée plus spécifiquement
- Regroupement : ,
 - Pour que les titres de tous niveaux soient rouges
h1, h2, h3, h4, h5, h6 { color: #ff0000; }

Application à tous les éléments

- sélecteur universel *
- agit sur tous les éléments HTML.
- Définir une couleur d'encre rouge par défaut :
 - * {
color: #ff0000;
}
- tous les éléments susceptibles d'avoir une couleur d'avant-plan, comme les paragraphes, titres, bordures... seront rouges, (sauf si une autre couleur leur est spécifiquement attribuée)

Application à une instance unique

- Pour que le texte d'une balise unique dans la page soit en vert,
- Définir l'id `#vert` :

```
#vert {  
  color: #008000;  
}
```
- Le contenu de la seule balise de la page ayant un attribut `id="vert"` sera vert
- Pour n'appliquer que si la balise unique est d'un certain type (ici `<p>`) :
- Définir l'id `p#vert` :

```
p#vert {  
  color: #008000;  
}
```
- Le contenu de la seule balise `<p id="vert">` sera vert
 - l'apparence des autres paragraphes ne change pas ,

Applications à certaines instances

- Pour que le texte des balises ayant un attribut `class="vert"` soit de couleur verte p.ex.
- Définir la classe `.vert`
 - par convention le nom commence par un "." (non repris dans la balise)

```
.vert {  
    color: #008000;  
}
```
- sera appliqué à `<p class="vert" >`, `<li class="vert" >`...
- l'apparence des autres balises ne change pas
- Pour n'appliquer que si la balise est d'un certain type (ici `<p>`) :
- Définir la classe `p.vert` :

```
p.vert {  
    color: #008000;  
}
```

Application à une portion de page

➤ `<div>... </div>` : permet de regrouper un **ensemble de lignes**

1) Définir une division délimitant la "partie de page" `<div>`

2) Définir la classe `.gris` :

```
.gris {
    color: #808080;
}
```

3) L'appliquer à la balise `<div>`

```
<div class="gris">
    <h1>...</h1>
    <p>...</p>
</div>
```

revient à appliquer le style à chacun des éléments :

```
<h1 class="gris">...</h1>
<p class="gris">...</p>
```

Application à une portion de paragraphe

- `...` : permet d'isoler une **portion de ligne**
- Pour qu'**une partie** d'un paragraphe soit grise
 - Créer un **span** de style gris autour des mots à griser

`<p>...ceci sera en gris...</p>`

DOM Document Object Model

- Dans la mesure du possible il est préférable de ne pas nommer les balises (id) et de ne pas créer de classe
 - altération du code html, perte de portabilité
- Pour désigner un/des éléments particuliers, on préférera définir leur contexte dans la page
 - i.e. leur place dans l'arborescence
 - i.e. les balises dans lesquelles elle est contenu

Exemple :

- pour les lignes() des listes ordonnées () incluses dans les listes non ordonnée() : `ul ol li {...}`

Application à un élément directement inclus

- Pour un élément **directement** ou **indirectement** contenu dans un autre élément (sans autre balise entre les deux)
- Pour que le contenu de l'élément **em** soit en rouge **seulement dans** les titres **h3**
- Définir la règle :

```
h3 em {  
  color: #ff0000;  
}
```
- S'applique aussi bien à : `<h3>...en rouge ...</h3>`
- qu'à : `<h3>... <code>...</code> ...</h3>`

Applicxation un élément **directement** contenu

- Pour que le contenu de l'élément em soit en rouge dans un paragraphe, mais pas dans les citations (<q>) qu'il contient
- Définir la règle

```
p>em {  
  color: #ff0000;  
}
```
- S'applique à : <p>... en rouge ...</p>
- mais pas à : <p> <q>...</q></p>

A un élément **successeur** d'un autre

- Pour que le paragraphe suivant une image soit rouge
- Définir la règle
 - `img + p { color: #ff0000;}`
 - S'applique à : `...<p>en rouge</p>`
 - mais pas à : `<h3>...</h3><p>...</p>`

- Plusieurs feuilles de styles peuvent être associées à un document
- Un style peut être (re)défini dans plusieurs feuilles de styles
- Un style peut être défini en plusieurs fois
 - complément ou partiellement
- L'apparence finale dépend :
 - du style de la balise
 - des styles définis dans ses conteneurs
(balises dans lesquelles la balise est contenue)

Priorités

- calcul de la priorité : concaténation de 3 valeurs

nb d'id+**nb de classes**+**nb d'éléments html** /pseudo (:hover,...)

p {...} : **001** (0 identifiant, 0 classe, 1 élément)

section p {...} : **002** (0 identifiant, 0 classe, 2 éléments)

.class {...} : **010** (0 identifiant, 1 classe, 0 élément)

p.class {...} : **011** (0 identifiant, 1 classe, 1 élément)

section p.class {...} : **012** (0 identifiant, 1 classe, 2 éléments)

#id {...} : **100** (1 identifiant, 0 classe, 0 élément)

p#id {...} : **101** (1 identifiant, 0 classe, 1 élément)

section p#id {...} : **102** (1 identifiant, 0 classe, 2 éléments)

Le tout pour chacune des propriétés prises individuellement

Une propriété fixée par un attribut dans la balise est prioritaire

Le tag **!important** force ce style il devient le plus prioritaire