

Technologies de Web 2.0:

Chapitre 2 : CSS3

Pr. Hajar LAZAR

hajar.lazar@gmail.com

Plan

- ❑ Introduction au langage CSS
- ❑ Appliquer CSS à une page HTML
- ❑ « Block » et « Inline » / « Div » et « Span »
- ❑ Les sélecteurs
- ❑ Mise en forme et Positionnement en CSS
- ❑ Les liens et les menus en CSS

C'est quoi CSS ?

- Les feuilles de styles (en anglais "Cascading Style Sheets", abrégé CSS) sont un langage qui permet de gérer la présentation d'une page Web.
- Les styles permettent de définir des règles appliquées à un ou plusieurs documents HTML.
- Le but de CSS est séparer la structure d'un document HTML et sa présentation.
- Il couvre toute la mise en forme : les polices, les couleurs, les marges, les arrière-plan...
- Quatre versions importantes de CSS :
CSS 1 ; CSS 2.0 ; CSS 2.1 ; CSS 3.

C'est quoi CSS ?

HTML
(pas de CSS)

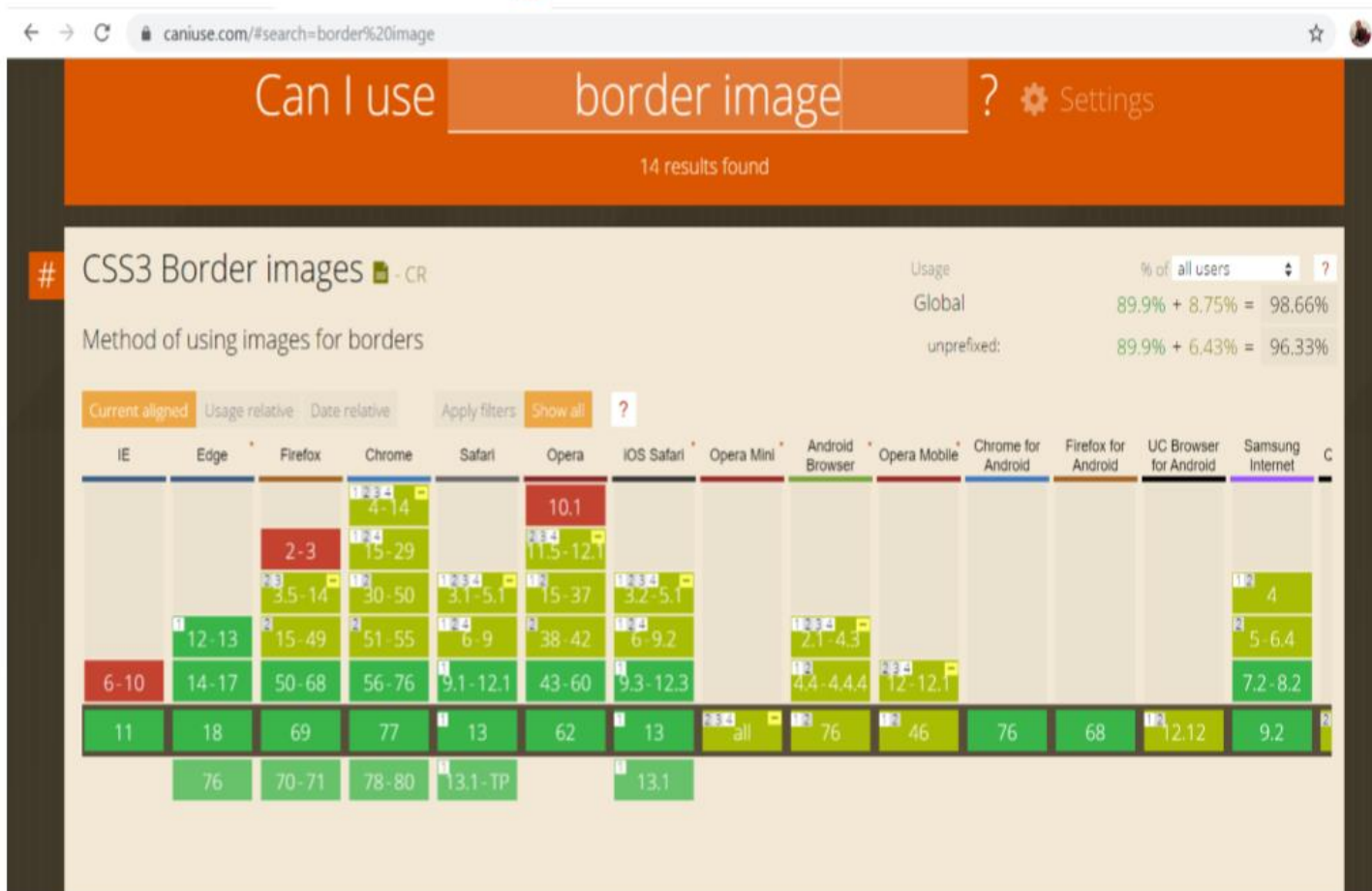


HTML + CSS



CSS et les navigateurs

- Les navigateurs web qui font le travail le plus complexe :
 - Lire le code CSS
 - Comprendre comment afficher la page.
- L'évolution considérable des navigateurs web pour prendre en charge le CSS
- Certain nombre de fonctionnalités de CSS qui ne marchent pas forcément sur tous les navigateurs.
- Le site www.caniuse.com qui propose des tables de compatibilité des fonctionnalités de HTML et CSS sur différents navigateurs (et sur leurs différentes versions)



CSS et les navigateurs

Table de compatibilité CSS sur www.caniuse.com

Mettre en place le CSS

On peut écrire du code en langage CSS à trois endroits différents :

➤ **Méthode 1 :**

– Directement dans les balises du fichier HTML via un attribut style (méthode la moins recommandée).

➤ **Méthode 2 :**

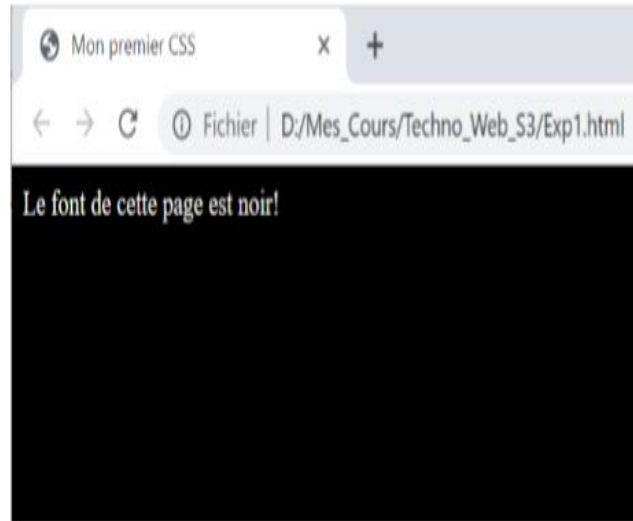
– Dans l'en-tête <head> .. </head> du fichier HTML

➤ **Méthode 3 :**

– Dans un fichier « fichier.css » (méthode la plus recommandée).

```
<html>
<head>
  <Title> Mon premier CSS </title>
</head>

<body style="background-color:black">
  <font color="white">Le font de cette page est noir!</font>
</body>
</html>
```



Appliquer CSS à une page HTML

Méthode 1 :

- Appliquer directement un style CSS à un document HTML avec l'attribut HTML « **style** ».
- Dans ce cas on parle d'un style local ou interne.

Appliquer CSS à une page HTML

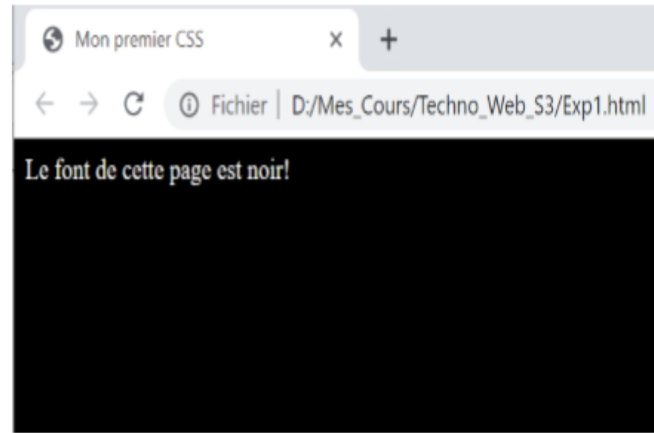
Méthode 2 :

- Mettre du code CSS dans la balise `<style>` déclarée à l'entête du document entre `<head>` et `</head>`.

```
<html>
  <head>
    <Title> Mon premier CSS </title>

    <style>
      body {background-color:black;}
    </style>
  </head>

  <body >
    <font color="white">Le font de cette page est noir!</font>
  </body>
</html>
```

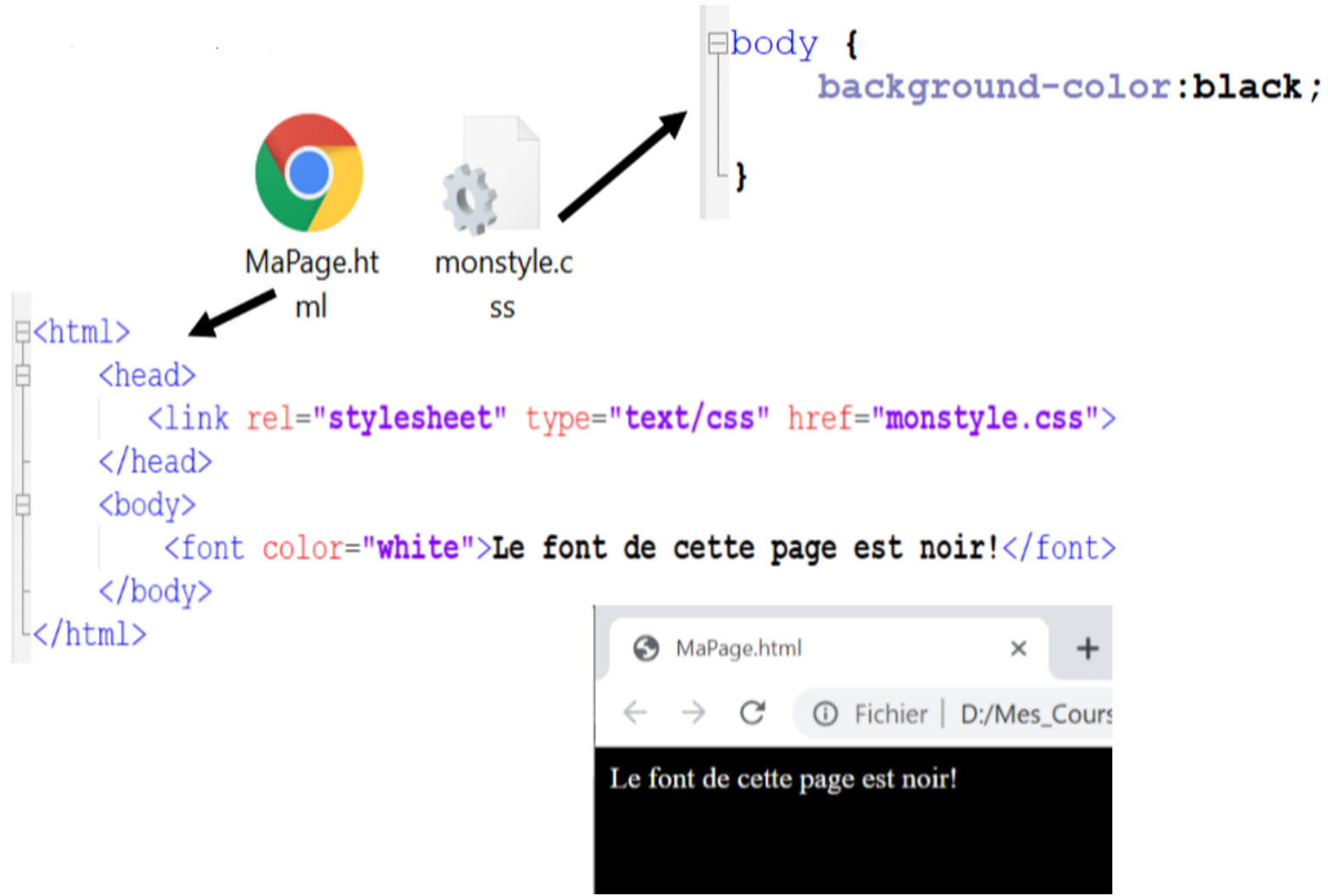


Mettre en place le CSS

Méthode 3 : (C'est la meilleure méthode)

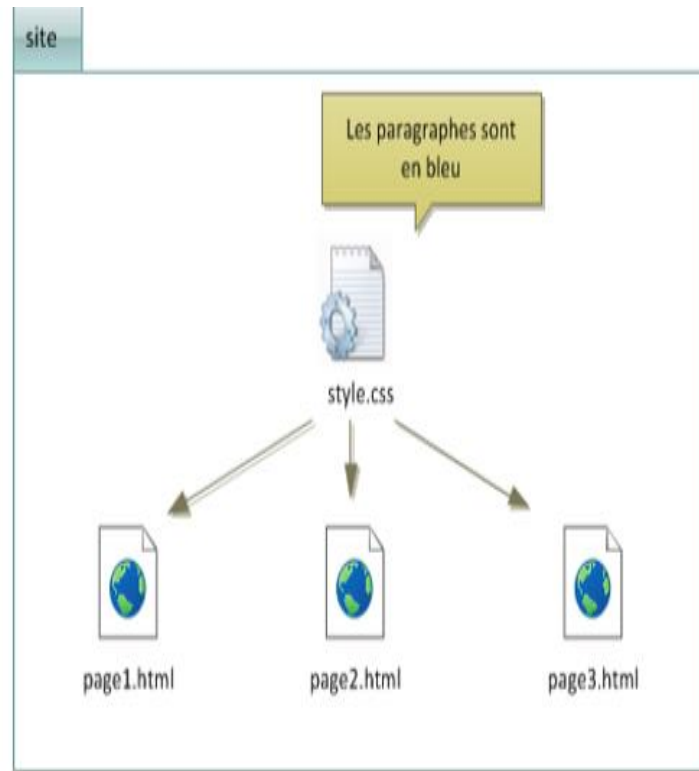
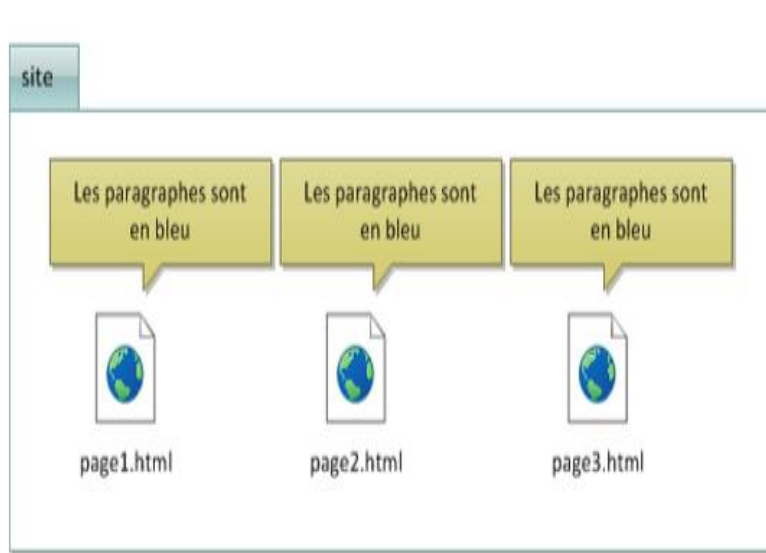
- Mettre du code CSS dans la balise <style> déclarée à l'entête du document entre <head> et </head>.
- Elle consiste à créer un **fichier externe** d'extension «.css » qui ne sera qu'une feuille de style et sera appelé par les documents HTML.
- Pour ce faire on ajoutera dans l'en-tête du document (**balise head**) un lien de la forme suivante :

```
<link  
    rel="stylesheet"  
    href="maFeuilleDeStyle.css"  
    type="text/css" >
```



Mettre en place le CSS

Méthode 3 :



Réutilisation du code CSS

- On peut appeler **un même fichier CSS sur plusieurs pages HTML**.
- Permettre de contrôler et de changer l'apparence de la totalité des pages d'un site web par la modification **d'un seul fichier**.

« block » et « inline » en HTML5

En HTML, la plupart des balises peuvent se ranger dans l'une ou l'autre de deux catégories :

- Les balises « **inline** » : c'est le cas par exemple des liens `<a> .. `.
- Les balises « **block** » : c'est le cas par exemple des paragraphes `<p> .. </p>`.

block :

- Créer automatiquement un retour à la ligne avant et après.
- Il suffit d'imaginer tout simplement un bloc.
- Votre page web sera en fait constituée d'une série de blocs les uns à la suite des autres.

« block » et « inline » en HTML5

inline :

- Une balise de type inline se trouve obligatoirement à l'intérieur d'une balise block.
- Une balise inline ne crée pas de retour à la ligne, le texte qui se trouve à l'intérieur s'écrit donc à la suite du texte précédent, sur la même ligne (c'est pour cela que l'on parle de balise « en ligne »).

Les balises universelles « span » et « div »

- Ce sont des balises qui n'ont aucun sens particulier
- Contrairement à `<p>` qui veut dire «paragraphe», `` «important», ...etc.
- Le principal intérêt de ces balises est que l'on peut leur appliquer une class (ou un id) pour le CSS quand aucune autre balise ne convient.
- ` .. ` → inline
- `<div> .. </div>` → block.

Les balises universelles « span » et « div »

- « span » et « div » sont « pratiques » dans certains cas, certes, mais **attention à ne pas en abuser**.
- Exemple d'un span inutile :
 - ``. Inutile puis qu'il existe la balise `` qui sert à indiquer l'importance !
- Exemple d'un div inutile :
 - `<div class="titre">`. Ceci est complètement absurde puisqu'il existe des balises faites spécialement pour les titres (`<h1>`, `<h2>`...).

Les sélecteurs

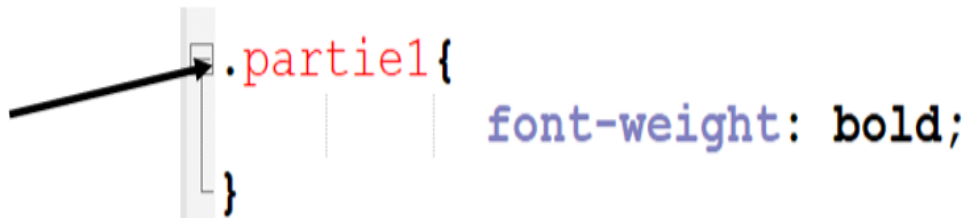
- Un sélecteur est par défaut un élément HTML
 - h1, h2, p, a ... : dont la mise en forme va être redéfinie par les propriétés et valeurs.
- Une règle CSS est de la forme suivante :

```
Sélecteur {  
    propriété1:valeur1;  
    Propriété2:valeur2; }
```

- Exemple

```
h1 {  
    color:red;  
}
```

En CSS, la class commence par un point « . »



```
.partie1{  
    font-weight: bold;  
}
```

```
<p>Un paragraphe n'est pas de la partie 1, sauf  
    <span class="partie1" >ce mot</span>  
</p>  
<p class="partie1" >contenu de la partie 1</p>
```

Un paragraphe n'est pas de la partie 1, sauf **ce mot**

contenu de la partie 1

Les sélecteurs de classe

Si les éléments proposés par HTML ne sont pas assez précis, il faut créer une classe de style. Par ex. on souhaite donner un style à quelques paragraphes.

```
#imageCoin{  
  position:absolute;  
  top:0px;  
  left:0px;  
}
```

Ce style ne sera appliqué qu'à
l'élément dont l'identifiant est
« imageCoin »

```

```

le seul élément affecté par
ce style

Les sélecteurs d'identifiant

Si un style ne doit être appliqué qu'à un élément et un seul, on peut créer un style qui ne correspondra qu'à l'élément dont l'identifiant est celui que vous aurez spécifié (grâce au #) :

Gestion de l'aspect des caractères

Attribut	Explications	Exemples de valeurs possibles
color	Couleur de la police	red, #FF0000 , rgb(255,0,0)
font-family	Police ou collection de polices. Lorsque l'on spécifie une collection de polices, le navigateur utilisera la première police de la liste qui sera installée sur le système.	Verdana, sans-serif;
font-size	Taille de la police sélectionnée	large,medium,small, 32px, 16pt
font-style	Choix entre normal, italique et oblique	normal, italic, oblique
font-weight	Applique une épaisseur de trait à la police. 400 correspond à un caractère normal, 700 à un Gras	normal, bold, 200
font-variant	Possibilité de petites majuscules	normal small-caps
text-decoration	Un texte barré, souligné...	none, Underline, line-through, Overline
text-transform	Choix de majuscules ou minuscules	Capitalize, uppercase, lowercase

Attribut	Explications	Exemples de valeurs possibles
background-color	Définit la couleur d'arrière plan.	red, #FF0000 , rgb(255,0,0)
background-image	Définit l'image d'arrière plan. Si la couleur et l'image d'arrière plan sont définies : <ul style="list-style-type: none"> • Si le lien vers l'image n'est pas bon, alors la couleur s'affiche. • Si l'image contient des zones transparentes, la couleur de fond sera visible dans ces zones. 	url("../image.jpg")
background-repeat	Détermine comment l'image d'arrière-plan est répétée : répétition-x : répétition horizontale répétition-y : répétition verticale	no-repeat repeat-x repeat-y
background-attachment	Détermine si l'image d'arrière plan est fixée (le texte défile dessus) ou si elle défile avec la page	Scroll, fixed
background-position	définit l'emplacement de l'image d'arrière-plan par rapport à la page.	50% 50%, 4cm 2cm, left, right, top, bottom, center

Gestion de l'arrière-plan et images d'arrière plan

Gestion des paragraphes

Attribut	Explications	Exemples de valeurs possibles
word-spacing	Ajoute de l'espace entre les mots	15px
letter-spacing	Ajoute de l'espace entre les lettres	normal, 10px
text-align	Alignement du texte: Gauche / Centré / Droit / Justifié	justify, left, center
text-indent	Retrait de première ligne	20px

Gestion du positionnement

- Positionnement: **float**
- Les blocs d'une page html se disposent par défaut verticalement les uns après les autres.
- Pour changer ce comportement, on utilise l'attribut `style="float:left"` (ou `"float:right"`)
- Avec `"left"`, un élément va se caler le plus à gauche possible, et accepter un élément à sa droite
- Avec `"right"`, un élément va se caler le plus à droite possible, et accepte le suivant à sa gauche
- En un sens, `"float"` transforme un `"bloc"` en `"inline"`

Gestion du positionnement

- Positionnement: **clear**
- Pour empêcher qu'un élément ne soit positionné à côté d'un élément flottant à sa gauche ou à sa droite on utilise l'attribut "clear "
 - style= "clear:left "
 - style="clear:right "
 - style="clear:both"(le plus utilisé)
- En un sens, "clear" permet de transformer un élément "inline" en "block"

Margin = marge externe

Padding = marge interne

Contenu

Border = bordure

Gestion du positionnement

margin, padding, border

Les marges : margin

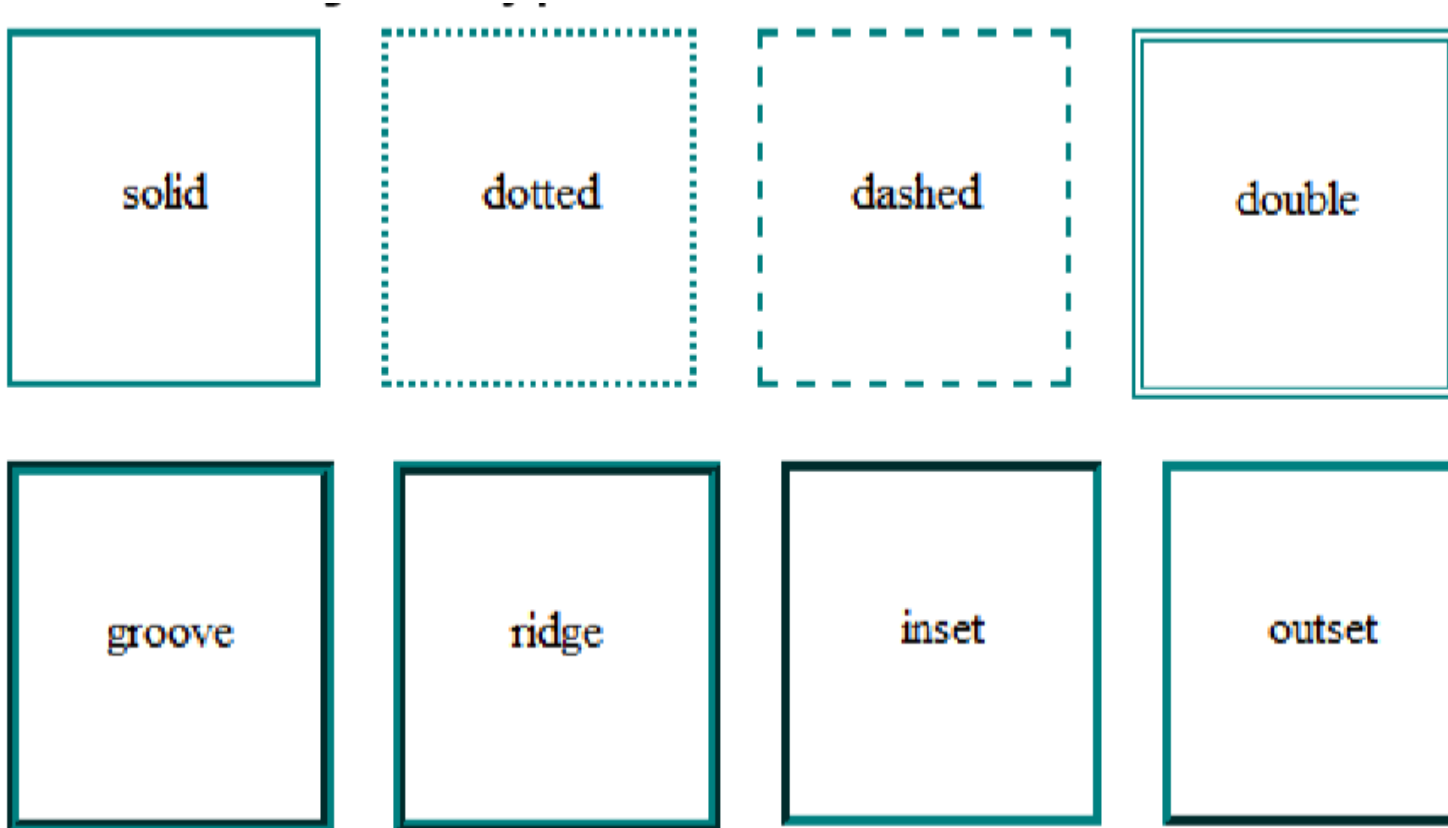
```
body {  
  margin-top: 10px;  
  margin-right: 20px;  
  margin-bottom: 30px;  
  margin-left: 40px;  
}
```



```
margin: 10px 20px 30px 40px;
```

- Définit l'importance de l'espace entre la bordure de l'élément et les autres éléments
- Il est exprimé en :
 - pixels (exemple 10px)
 - Pourcentage (70%)
 - auto (réglage automatique)

Les bordures



- **border-color:** Couleur de la bordure
- **border-width:** Taille de la bordure
- **border-style:** Type de la bordure

Les liens

- En HTML, les liens sont exprimés avec les balises <a>.
- En CSS on utilise « a » comme sélecteur.
- Pour contrôler les effets appliqués aux liens, on utilise ce qu'on appelle des pseudo-classes.
- Pour un lien, la pseudo-classe exprime l'état de celui-ci. Par exemple, le lien est-il visité, actif ou survolé.
- Pour chaque état on peut appliquer des styles CSS tels que les couleurs, les arrière-plans, les propriétés de texte...

```
a {  
    color: blue;  
}
```

Les liens sont alors mis en bleu

Les liens

➤ Lien non visité:

- Ajouter au sélecteur « a » la pseudo-classe « **:link** » pour désigner un lien non visité.

➤ Lien actif :

- Ajouter au sélecteur « a » la pseudo-classe « **:active** » pour désigner un lien actif

Les liens

➤ Lien visité :

- Ajouter au sélecteur « a » la pseudo-classe
- « :visited » pour désigner un lien visité

➤ Lien survolé:

- On ajoute au sélecteur « a » la pseudo-classe « :hover » pour désigner un lien actif:

Lien non visité

```
a:link {  
    color: bleu;  
    text-decoration: none;  
}
```

Lien visité

```
a:visited {  
    color: orange;  
    background-color: green;  
}
```

Lien survolé

quand on place le curseur
de la souris en dessus

```
a:hover {  
    color: purple;  
    letter-spacing: 4px;  
    font-variant: small-caps;  
    text-decoration: underline;  
}
```

Lien actif

```
a:active {  
    color: yellow;  
}
```

Les liens

Les liens

- Le public s'est habitué aux liens hypertextes bleus soulignés.
- Ces effets classiques dégradent un peu le design d'un bon site Web.
- Le mieux est de supprimer le souligné et adopter une couleur commune pour tous les liens du site Web. Comme ça le visiteur de votre site les reconnaît facilement.

Les liens

- Pour supprimer le souligné il suffit d'utiliser le **style: text-decoration** avec la valeur « **none** ».

```
a { text-decoration: none; }
```



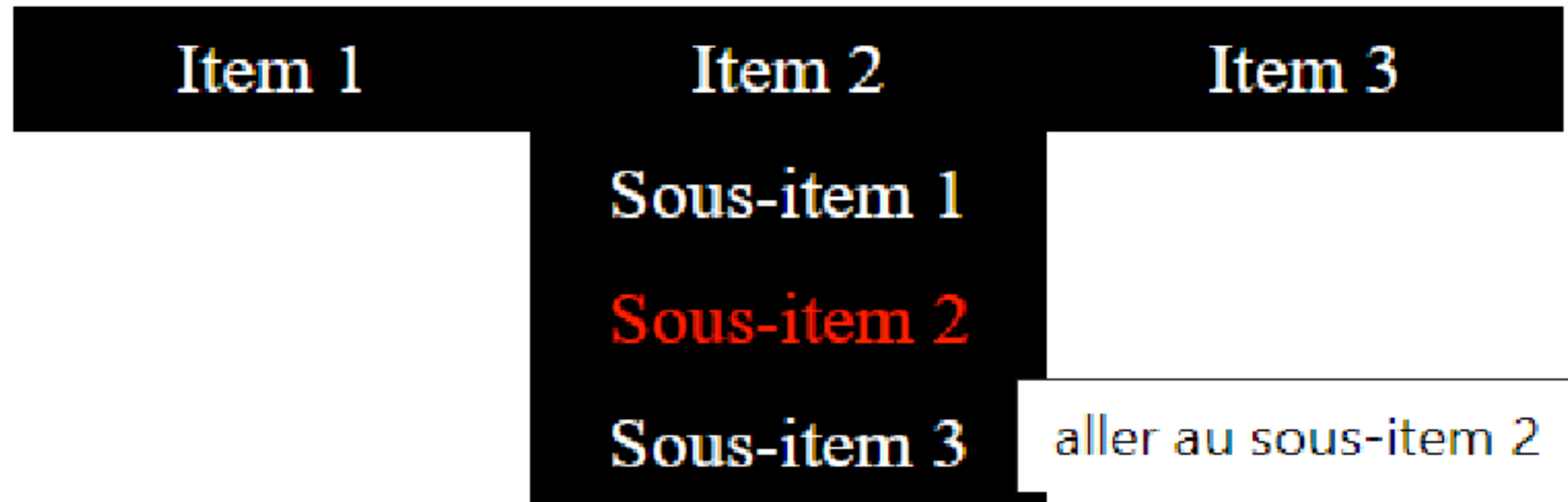
Ceci supprime le souligné quelque soit l'état du lien (visité ou non, actif ou survolé).

```
a:visited { text-decoration: none; }
```



Sinon on associe ce style à l'état désiré uniquement. Par exemple seul un lien visité ne doit pas être souligné

Les menus



Les menus

Code HTML

```
<div id="menu">
  <ul>
    <li><a href="#" title="aller à l'item 1">Item 1</a></li>
    <li><a href="#" title="aller à l'item 2">Item 2</a>
      <ul>
        <li><a href="#" title="aller au sous-item 1">Sous-item 1</a></li>
        <li><a href="#" title="aller au sous-item 2">Sous-item 2</a></li>
        <li><a href="#" title="aller au sous-item 3">Sous-item 3</a></li>
      </ul>
    </li>
    <li><a href="#" title="aller à l'item 3">Item 3</a></li>
  </ul>
</div>
```

- Item 1
- Item 2
 - Sous-item 1
 - Sous-item 2
 - Sous-item 3
- Item 3



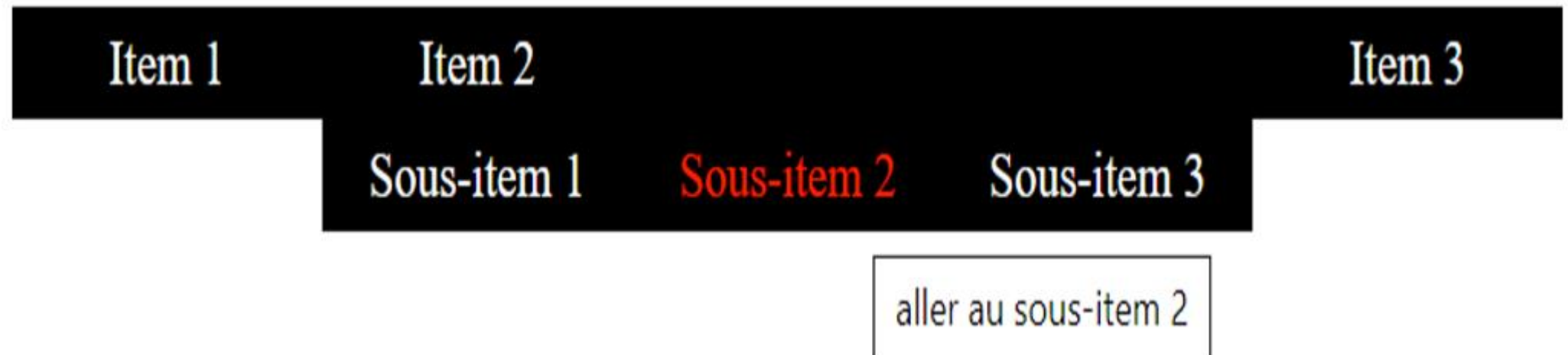
```
#menu ul {  
  margin:0;  
  padding:0;  
  list-style-type:none;  
  text-align:center;  
}  
#menu li {  
  float:left;  
  margin:auto;  
  padding:0;  
  background-color:black;  
}
```

```
#menu li a {  
  display:block;  
  width:100px;  
  color:white;  
  text-decoration:none  
  padding:5px;  
}  
#menu li a:hover {  
  color:#FF2200;  
}
```

Les menus

Code CSS

Les menus: Résultat



```
/* Faire disparaître les sous-items */  
#menu ul li ul {  
  display:none;  
}
```

```
/* Remettre en vertical le sous menu */  
#menu ul li:hover ul {  
  display:block;  
}
```

```
/* Afficher les sous-item en survolant  
l'item principal */  
#menu li:hover ul li {  
  float:none;  
}
```

Les menus

Les menus

Résultat final

