

## **Leçon 1 : Qu'est-ce que CSS ?**

Peut-être avez-vous déjà entendu parler de CSS sans vraiment savoir de quoi il s'agissait ? Dans cette leçon, vous en saurez plus sur CSS et sur ce que vous pouvez en tirer. CSS est l'abréviation de « **Cascading Style Sheets** ».

### **Qu'est-ce qu'on peut faire avec CSS ?**

CSS est un langage de style qui définit la présentation des documents HTML. Par exemple, CSS couvre les polices, les couleurs, les marges, les lignes, la hauteur, la largeur, les images d'arrière-plan, les positionnements évolués et bien d'autres choses. Attendez de voir !

HTML peut être (mal) utilisé pour la présentation des sites Web. Mais CSS offre plus d'options et se montre plus précis et sophistiqué. CSS est pris en charge par tous les navigateurs actuels.

Après seulement quelques leçons dans ce tutoriel, vous pourrez fabriquer vos propres feuilles de style avec CSS pour donner un aspect tout nouveau à votre site Web.

### **Quelle est la différence entre CSS et HTML ?**

HTML sert à structurer le contenu, CSS sert à formater un contenu structuré.

Bon, ça sonne un peu technique et confus. Mais continuez à lire, ça prendra tout son sens bientôt.

Retour au bon vieux temps où Madonna était une vierge et un type nommé Tim Berners Lee inventait le World Wide Web, le langage HTML était seulement utilisé pour structurer du texte. Un auteur pouvait marquer son texte en déclarant "voici un titre" ou "voici un paragraphe", avec les balises HTML `<h1>`, ou `<p>`.

La popularité du Web croissant, les concepteurs se mirent à rechercher des moyens pour ajouter de la présentation aux documents en ligne. Pour satisfaire à cette demande, les éditeurs de navigateurs (en ce temps-là, Netscape et Microsoft) inventèrent de nouvelles balises HTML, comme par exemple `<font>`, qui différaient des éléments originaux en cela qu'ils définissaient une présentation et non une structure.

Cela conduisit aussi à une situation où les balises de structure originales, telle que `<table>`, étaient de plus en plus utilisées à tort pour présenter les pages et non pour ajouter une structure au texte. Beaucoup de nouvelles balises de présentation, telle que `<blink>`, n'étaient reconnues que par un type de navigateur. "Vous devez utiliser le navigateur X pour voir cette page" devint un avertissement courant sur les sites Web.

CSS fut inventé pour remédier à cette situation en offrant aux concepteurs Web des possibilités de présentations sophistiquées, gérées par tous les navigateurs. En même temps, séparer le style de présentation des documents de leur contenu rend leur entretien beaucoup plus facile.

### **Quels sont les avantages de CSS ?**

CSS fut une révolution dans l'univers de la conception Web. Les avantages concrets de CSS sont les suivants :

- Le contrôle de la présentation de plusieurs documents par une seule feuille de style ;
- Un contrôle plus précis de la présentation ;
- Des présentations différentes appliquées à des types de médias différents (à l'écran, à l'impression, etc.) ;
- De nombreuses techniques évoluées et sophistiquées.

## Leçon 2 : Comment CSS fonctionne-t-il ?

Dans cette leçon, vous apprendrez à fabriquer votre première feuille de style. Vous saurez ce qu'est le modèle CSS de base et quels sont les codes nécessaires pour utiliser CSS dans un document HTML.

Beaucoup de propriétés des feuilles de style en cascade (CSS) sont similaires à celles de HTML. Si vous avez l'habitude d'utiliser HTML pour la présentation, vous reconnaîtrez donc beaucoup de ces codes. Voyons un exemple concret.

## La syntaxe CSS de base

Supposons que nous voulions que le fond d'une page Web soit d'un beau rouge : Avec **HTML**, nous l'aurions fait comme ceci :

<body bgcolor="#FF0000">

Avec CSS, on peut obtenir le même résultat comme cela :

```
body {background-color: #FF0000;}
```

Comme vous l'aurez remarqué, les codes sont plus ou moins identiques pour HTML et CSS. Cet exemple illustre également le modèle fondamental de CSS :

```
selector {property: value;}
```

↑  
 À quelle(s)  
 balise(s) HTML  
 la propriété  
 s'applique  
 (p.ex. "body")

↑  
 La propriété, par  
 exemple, la couleur  
 d'arrière-plan  
 ("background-color")

↙ La valeur de la  
 propriété de couleur  
 d'arrière-plan, par  
 exemple, le rouge  
 ("#FF0000")

Mais où place-t-on le code CSS ? C'est précisément ce que nous allons voir maintenant.

## Appliquer CSS à un document HTML

Il y a trois façons d'appliquer le style CSS à un document HTML. Elles sont toutes expliquées ci-dessous. Nous vous recommandons d'examiner la troisième méthode, c'est-à-dire celle externe.

### Méthode 1 : Local - Dans la ligne (l'attribut style)

Une façon d'appliquer du style CSS à HTML est celle avec l'attribut HTML `style`. En reprenant l'exemple précédent avec l'arrière-plan rouge, on peut l'appliquer ainsi :

```
<html>
<head>
  <title>Exemple</title>
</head>
<body style="background-color: #FF0000;">
  <p>Cette page est rouge</p>
</body>
</html>
```

## Méthode 2 : Global - Interne (l'élément style)

Une autre méthode consiste à inclure le code CSS avec la balise HTML `<style>`. Par exemple, comme ceci :

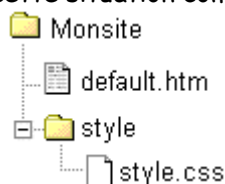
```
<html>
<head>
  <title> Exemple </title>
  <style type="text/css">
    body {background-color: #FF0000;}
  </style>
</head>
<body>
  <p>Cette page est rouge</p>
</body>
</html>
```

## Méthode 3 : Externe (un lien vers une feuille de style: un fichier)

La méthode recommandée est celle avec un lien vers ladite feuille de style externe. Pour le reste de ce tutoriel, nous utiliserons cette méthode pour tous les exemples.

Une feuille de style externe est simplement un fichier texte ayant l'extension « .css ». Comme n'importe quel fichier, la feuille de style peut être rangée sur un serveur Web ou sur un disque dur.

Par exemple, disons que votre feuille de style se nomme « style.css » et se trouve dans un dossier appelé « style ». On peut illustrer cette situation comme ceci :



L'astuce consiste à créer un lien depuis le document HTML (default.htm) vers la feuille de style (style.css). Ce lien peut être créé en une ligne de code HTML :

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

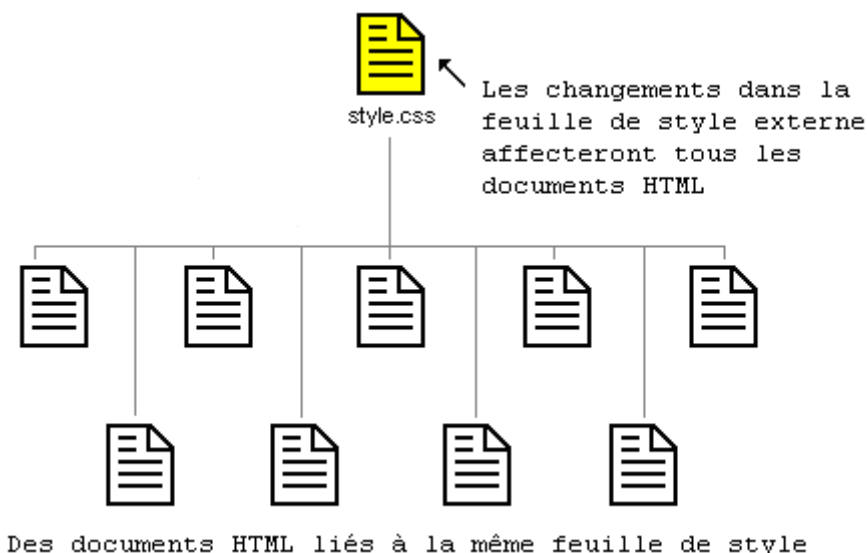
Remarquez comment on indique le chemin de la feuille de style avec l'attribut href.

La ligne de code doit s'inscrire dans la section d'en-tête du code HTML, c'est-à-dire entre les balises `<head>` et `</head>`. Comme ceci :

```
<html>
<head>
  <title>Mon document</title>
  <link rel="stylesheet" type="text/css" href="style/style.css" />
</head>
<body>
```

Ce lien instruit le navigateur d'utiliser la présentation du fichier CSS pour afficher le fichier HTML.

Ce qui est vraiment futé, c'est que plusieurs documents HTML peuvent être reliés à la même feuille de style. En d'autres termes, on peut utiliser un seul fichier CSS pour contrôler la présentation de plusieurs documents HTML.



Cette technique est susceptible d'économiser beaucoup d'efforts. Par exemple, si vous voulez changer la couleur d'arrière-plan d'un site Web contenant 100 pages, une feuille de style peut vous épargner la modification manuelle des 100 documents HTML. Avec CSS, la modification peut intervenir en quelques secondes en changeant juste une seule ligne de code de la feuille de style centrale.

Mettons donc en pratique ce que nous venons d'apprendre.

### Essayez vous-même :

Lancez Bloc-notes (ou votre éditeur de texte habituel) et créez deux fichiers, un fichier HTML et un fichier CSS, avec les contenus suivants :

#### **default.htm**

```
<html>
<head>
  <title>Mon document</title>
  <link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
  <h1>Ma première feuille de style</h1>
</body>
</html>
```

#### **style.css**

```
body {
  background-color: #FF0000;
}
```

Placez maintenant les deux fichiers dans le même dossier. Songez à sauvegarder les fichiers avec les bonnes extensions (respectivement « .htm » et « .css »)

Ouvrez « **default.htm** » dans votre navigateur et constatez le fond rouge de la page. Félicitations ! Vous avez fabriqué votre première feuille de style !

## **Leçon 3 : Les couleurs et les arrières-plans**

Dans cette leçon, vous apprendrez comment appliquer des couleurs et des couleurs d'arrière-plan à vos sites Web. Nous étudierons également des méthodes évoluées pour positionner et contrôler les images d'arrière-plan. Les propriétés CSS suivantes seront expliquées :

- ✓ **color**
- ✓ **background-color**
- ✓ **background-image**
- ✓ **background-repeat**
- ✓ **background-attachment**
- ✓ **background-position**
- ✓ **background**

### **La couleur d'avant-plan : la propriété 'color'**

La propriété **color** décrit la couleur d'avant-plan d'un élément.

Par exemple, supposons que nous voulions tous les titres du document en rouge foncé. Les titres sont tous balisés avec l'élément HTML `<h1>`. Le code suivant donne aux éléments `<h1>` une couleur rouge.

```
h1 {  
    color: #ff0000;  
}
```

On peut définir les couleurs avec des valeurs hexadécimales comme dans l'exemple précédent (`#ff0000`), ou avec les noms des couleurs ("red"), ou avec des valeurs RGB (`rgb(255,0,0)`).

### **La propriété 'background-color'**

La propriété **background-color** décrit la couleur d'arrière-plan des éléments.

L'élément `<body>` est le réceptacle de tout le contenu du document HTML. Pour changer la couleur d'arrière-plan d'une page entière, il faudrait donc appliquer la propriété 'background-color' à l'élément `<body>`.

On peut aussi appliquer une couleur d'arrière-plan à d'autres éléments y compris les titres et le texte. Dans l'exemple ci-dessous, Des couleurs d'arrière-plan différentes sont appliquées aux éléments `<body>` et `<h1>`.

```
body {  
    background-color: #FFCC66;  
}  
  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

Remarquez que nous avons appliqué deux propriétés à `<h1>` en les séparant par un point-virgule.

### **Les images d'arrière-plan [background-image]**

La propriété CSS **background-image** sert à insérer une image d'arrière-plan.

Comme exemple d'image d'arrière-plan, nous utilisons le papillon ci-dessous. Vous pouvez télécharger l'image pour l'utiliser sur votre propre ordinateur (en cliquant sur le bouton de

droite de la souris et en sélectionnant « Enregistrer l'image sous », ou vous pouvez utiliser une autre image si vous voulez.



Pour insérer l'image du papillon en arrière-plan d'une page Web, appliquez simplement la propriété `background-image` à l'élément `<body>` et indiquez le chemin de l'image.

```
body {  
    background-color: #FFCC66;  
    background-image: url("butterfly.gif");  
}  
  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

Remarquez la façon dont nous avons indiqué le chemin de l'image avec `url("butterfly.gif")`. Cela signifie que l'image se trouve dans le même dossier que la feuille de style. On peut aussi appeler des images dans d'autres dossiers avec `url("../images/butterfly.gif")`, ou même sur Internet en donnant l'adresse complète du fichier : `url("http://www.monsite.net/butterfly.gif")`.

## Répéter l'image d'arrière-plan [background-repeat]

Avez-vous remarqué, dans l'exemple précédent, que le papillon était répété par défaut horizontalement et verticalement pour couvrir la totalité de l'écran ? Ce comportement est contrôlé par la propriété `background-repeat`.

Le tableau suivant décrit les quatre valeurs différentes de `background-repeat`.

Valeur	Description
<code>Background-repeat: repeat-x</code>	L'image se répète horizontalement
<code>background-repeat: repeat-y</code>	L'image se répète verticalement
<code>background-repeat: repeat</code>	L'image se répète horizontalement et verticalement
<code>background-repeat: no-repeat</code>	L'image ne se répète pas

Par exemple, pour éviter la répétition d'une image d'arrière-plan, le code devrait ressembler à ceci :

```
body {  
    background-color: #FFCC66;  
    background-image: url("butterfly.gif");  
    background-repeat: no-repeat;  
}  
  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

## Bloquer l'image d'arrière-plan [background-attachment]

La propriété background-attachment définit si l'image d'arrière-plan est fixe ou bien défile avec l'élément conteneur.

Une image d'arrière-plan bloquée ne bougera pas avec le texte lorsque le lecteur fait défiler la page, au contraire d'une image d'arrière-plan non bloquée qui défilera avec le texte de la page Web.

Le tableau suivant décrit les deux valeurs différentes de background-attachment. Cliquez sur les exemples pour voir la différence entre défiler et fixé.

Valeur	Description
Background-attachment: scroll	L'image défile avec la page (non bloquée)
Background-attachment: fixed	L'image est bloquée

Par exemple, le code suivant bloquera l'image d'arrière-plan.

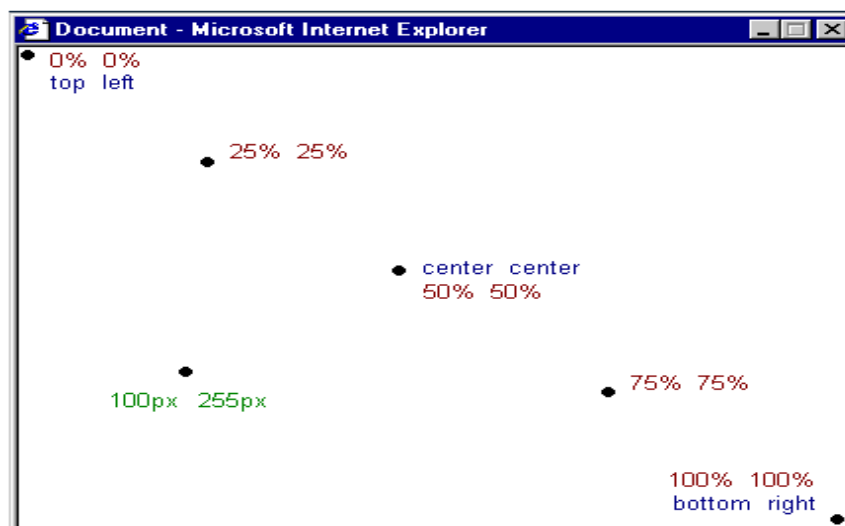
```
body {  
    background-color: #FFCC66;  
    background-image: url("butterfly.gif");  
    background-repeat: no-repeat;  
    background-attachment: fixed;  
}  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

## Positionner une image d'arrière-plan [background-position]

Par défaut, l'image d'arrière-plan se positionnera dans le coin supérieur gauche de l'écran. La propriété background-position permet de changer cette position prédéfinie et de placer l'image d'arrière-plan n'importe où à l'écran.

Il y a plusieurs méthodes pour fixer la valeur de background-position. Par contre, elles se présentent toutes sous la forme d'un jeu de coordonnées. Par exemple, la valeur '100px 200px' positionne l'image d'arrière-plan à 100px depuis la gauche et à 200px depuis le haut de la fenêtre du navigateur.

Les coordonnées peuvent s'exprimer en pourcentages de la largeur de l'écran, ou en unités fixes (pixels, centimètres, etc.), ou on peut utiliser les mots-clés "top", "bottom", "center", "left" ou "right". Le modèle suivant illustre ce système :



Le tableau suivant fournit quelques exemples.

Valeur	Description
background-position: 2cm 2cm	L'image est positionnée à 2 cm de la gauche et à 2 cm du haut de la page
background-position: 50% 25%	L'image est positionnée au centre et à un quart de la page vers le bas
background-position: top right	L'image est positionnée au coin supérieur droit de la page

L'exemple de code à suivre positionne l'image d'arrière-plan dans le coin inférieur droit de la page :

```
body {  
    background-color: #FFCC66;  
    background-image: url("butterfly.gif");  
    background-repeat: no-repeat;  
    background-attachment: fixed;  
    background-position: right bottom;  
}  
  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

## Concision [background]

La propriété background est un raccourci pour toutes les propriétés listées dans cette leçon.

Avec background, on peut comprimer plusieurs propriétés et donc écrire une feuille de style plus courte, ce qui en facilite la lecture.

Par exemple, les cinq lignes suivantes :

```
background-color: #FFCC66;  
background-image: url("butterfly.gif");  
background-repeat: no-repeat;  
background-attachment: fixed;  
background-position: right bottom;
```

On peut obtenir le même résultat avec background en une seule ligne de code :

```
background: #FFCC66 url("butterfly.gif") no-repeat fixed right bottom;
```

La liste de commande est la suivante :

[background-color] | [background-image] | [background-repeat] | [background-attachment] | [background-position]

Si une propriété manque, elle prend automatiquement sa valeur par défaut. Par exemple, si on supprime background-attachment et background-position de l'exemple :

```
background: #FFCC66 url("butterfly.gif") no-repeat;
```

Ces deux propriétés non définies prendront tout simplement leurs valeurs par défaut, qui sont comme chacun sait "scroll" et "top left".



## Leçon 4 : Les polices

Dans cette leçon, vous apprendrez ce que sont les polices et comment les appliquer avec CSS. Nous étudierons également comment contourner le problème des polices spécifiques choisies pour un site Web qui ne sont visibles que si elles sont installées sur l'ordinateur utilisé pour accéder au site Web. Les propriétés CSS suivantes seront décrites :

- ✓ **font-family**
- ✓ **font-style**
- ✓ **font-variant**
- ✓ **font-weight**
- ✓ **font-size**
- ✓ **font**

### La famille de polices [font-family]

La propriété font-family sert à indiquer la liste prioritaire des polices à utiliser pour l'affichage d'un élément donné ou d'une page Web. Si la première police de la liste n'est pas disponible sur l'ordinateur utilisé pour accéder au site, la police suivante est essayée et ainsi de suite jusqu'à ce qu'il y en ait une qui convienne.

Il y a deux types de noms pour catégoriser les polices : les noms de familles et les familles génériques. Les deux termes sont expliqués ci-dessous.

Le nom de famille :

Comme exemples de noms de familles (souvent appelées « polices »), on trouve "Arial", "Times New Roman" ou "Tahoma".

La famille générique

On peut décrire les familles génériques comme des groupes de noms de familles ayant des aspects uniformes. Un exemple est celui de la famille sans sérif, qui est un ensemble de polices sans « empattement ».

On peut illustrer cette différence comme ceci :

Times New Roman  
Garamond  
Georgia

Ces trois familles de polices appartiennent au genre **sérif**. Elles se caractérisent par leurs « empattements »

Trebuchet  
Arial  
Verdana

Ces trois familles de polices appartiennent au genre **sans sérif**. Elles se caractérisent par l'absence d'« empattements »

Courier  
Courier New  
Andale Mono

Ces trois familles de polices appartiennent au genre **monospace**. Elles se caractérisent par une chasse fixe

Lorsqu'on liste des polices pour un site Web, on commence naturellement par la police préférée suivie par des polices de remplacement. Il est recommandé de terminer la liste par une famille générique. Ainsi, la page s'affichera au moins avec une police de la même famille si aucune des polices indiquées n'était disponible.

Un exemple de liste prioritaire de polices ressemblerait à ceci :

```
h1 {font-family: arial, verdana, sans-serif;}  
h2 {font-family: "Times New Roman", serif;}
```

Les titres marqués par des éléments `<h1>` s'afficheront dans la police "Arial". Si elle n'est pas installée dans l'ordinateur de l'utilisateur, la police "Verdana" sera utilisée à la place. Si ces deux polices sont indisponibles, les titres s'afficheront dans une police de la famille sans-serif.

Remarquez que le nom de la police "Times New Roman" contient des espaces, et elle est donc listée entre des guillemets.

## Le style des polices [font-style]

La propriété font-style définit si le style de la police concernée doit être **normal**, **italic** ou **oblique**. Dans l'exemple à suivre, tous les titres balisés par des éléments `<h2>` s'afficheront en italiques.

```
h1 {font-family: arial, verdana, sans-serif;}
h2 {font-family: "Times New Roman", serif; font-style: italic;}
```

## Les variantes de polices [font-variant]

La propriété font-variant sert à choisir entre les variantes **normal** ou **small-caps** (N.d.T. petites capitales) d'une police. Une police **small-caps** utilise des lettres en capitales de taille réduite à la place des lettres en minuscules. Embrouillé ? Étudiez ces exemples :

Sans Book SC	Sans Bold SC	Serif Book SC	Serif Bold SC
ABCABC	ABCABC	ABCABC	ABCABC

Si la propriété font-variant vaut **small-caps** et qu'aucune police en petites capitales n'est disponible, le navigateur affichera très probablement le texte en capitales à la place.

```
h1 {font-variant: small-caps;}
h2 {font-variant: normal;}
```

## La graisse des polices [font-weight]

La propriété font-weight décrit avec quel degré de graisse (ou de « noir ») présenter la police. Elle peut avoir une graisse **normal** ou **bold** (N.d.T. caractères gras). Certains navigateurs permettent même d'utiliser des nombres entre 100 et 900 (les centaines) pour décrire la graisse de la police.

```
p {font-family: arial, verdana, sans-serif;}
td {font-family: arial, verdana, sans-serif; font-weight: bold;}
```

## Le corps des polices [font-size]

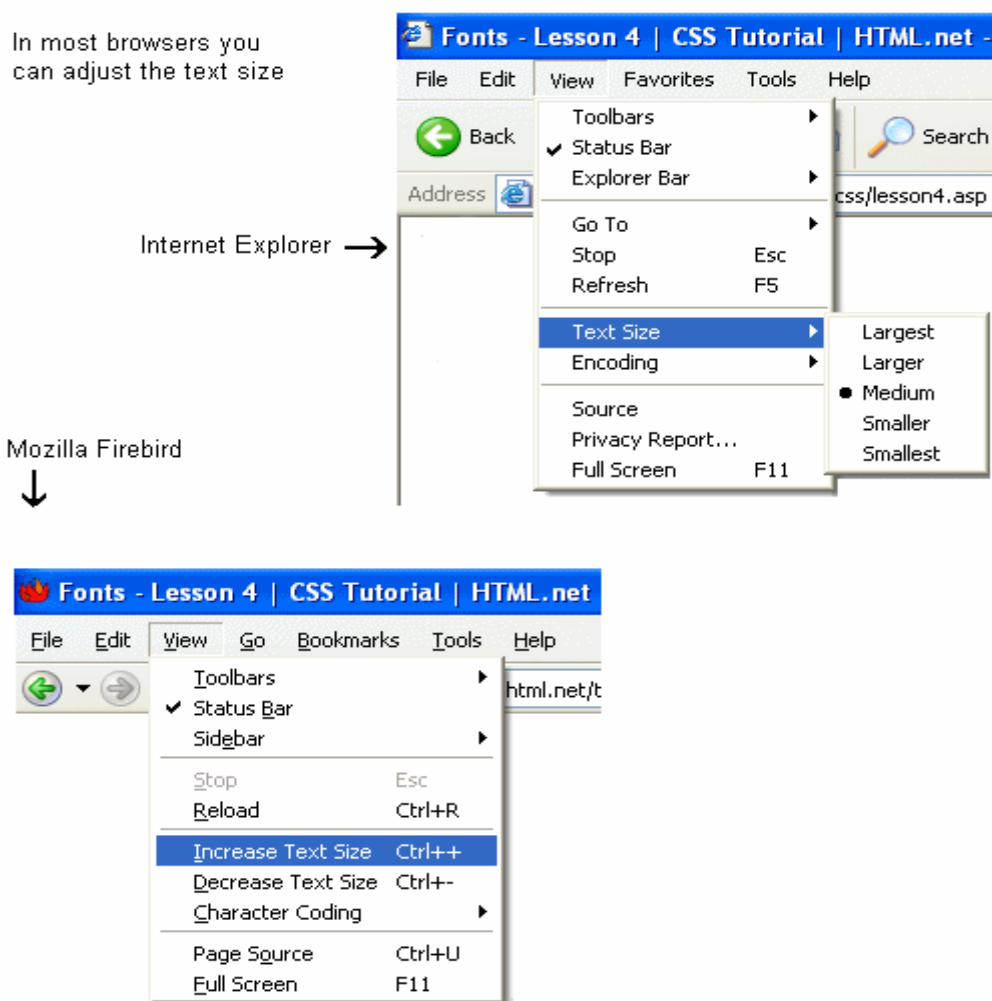
Le corps d'une police se règle par la propriété font-size.

On peut choisir parmi beaucoup d'unités différentes (par exemple, pixels et pourcentages) pour décrire les corps des polices. Dans ce tutoriel, nous nous concentrerons sur les unités les plus courantes et appropriées. Les exemples comprennent :

```
h1 {font-size: 30px;}
h2 {font-size: 12pt;}
h3 {font-size: 120%;}
p {font-size: 1em;}
```

Il existe une différence essentielle entre les quatre unités précédentes. Les unités « **px** » et « **pt** » font que le corps de la police est absolu, tandis que « **%** » et « **em** » permettent à l'utilisateur de l'ajuster si nécessaire. Beaucoup d'utilisateurs ont des handicaps, sont âgés ou simplement pâtissent d'une mauvaise vue ou de la piètre qualité de leur moniteur. **Pour rendre votre site Web accessible** à tous, vous devriez utiliser des unités ajustables telles que « **%** » ou « **em** ».

Voici une illustration de la façon d'ajuster la taille du texte dans Mozilla Firefox et dans Internet Explorer. Essayez vous-même (super, n'est-ce pas ?)



## Concision [font]

Avec la propriété raccourcie font, il est possible de couvrir toutes les différentes propriétés de police en un coup.

Par exemple, prenons ces quatre lignes de code servant à décrire les propriétés de police de <p> :

```
p {
    font-style: italic;
    font-weight: bold;
    font-size: 30px;
    font-family: arial, sans-serif;
}
```

Avec la propriété raccourcie, on peut simplifier le code :

```
p {
    font: italic bold 30px arial, sans-serif;
}
```

L'ordre des valeurs de la propriété font est le suivant :  
font-style | font-variant | font-weight | font-size | font-family

## L'indentation du texte [text-indent]

La propriété text-indent permet d'ajouter une touche d'élégance aux paragraphes de texte en appliquant une indentation à la première ligne du paragraphe. Dans l'exemple suivant, un alinéa de **30px** est appliqué à tous les paragraphes balisés par un élément <p> :

```
p {
    text-indent: 30px;
}
```

## L'alignement du texte [text-align]

La propriété CSS text-align correspond à l'attribut align utilisé dans les versions anciennes de HTML. Le texte peut être aligné à gauche (**left**), à droite (**right**) ou centré (**center**). Outre cela, la valeur **justify** étirera chaque ligne de sorte que les marges à gauche et à droite soient droites toutes les deux. Cette mise en page est courante, par exemple, dans les journaux et les magazines.

Dans l'exemple suivant, le texte des titres du tableau <th> est aligné à droite, tandis que les données du tableau <td> sont centrées. Enfin, les paragraphes de texte normaux sont justifiés :

```
th {
    text-align: right;
}

td {
    text-align: center;
}

p {
    text-align: justify;
}
```

## La décoration du texte [text-decoration]

La propriété text-decoration permet d'ajouter des « décorations » ou « effets » différents au texte. Par exemple, on peut souligner le texte, le barrer, ou tracer un trait au-dessus, etc. Dans l'exemple suivant, les éléments <h1> sont des titres soulignés, les éléments <h2> sont des titres avec un trait au-dessus et les éléments <h3> des titres barrés.

```
h1 {
    text-decoration: underline;
}

h2 {
    text-decoration: overline;
}

h3 {
    text-decoration: line-through;
}
```

## L'espacement des lettres [letter-spacing]

L'espacement entre les caractères du texte peut être défini avec la propriété letter-spacing. La valeur de la propriété est simplement celle de l'espacement désiré. Par exemple, si vous voulez un espacement de **3px** entre les lettres d'un paragraphe de texte <p>, et de **6px** entre les lettres des titres <h1>, voici le code à utiliser :

```
h1 {  
    letter-spacing: 6px;  
}  
  
p {  
    letter-spacing: 3px;  
}
```

## La transformation du texte [text-transform]

La propriété text-transform contrôle la capitalisation du texte. On peut choisir une **capitalisation initiale**, une **mise en majuscules** ou une **mise en minuscules**, indépendamment de l'aspect du texte original dans le code HTML.

Voici un exemple avec le mot « titre » que l'on peut présenter à l'utilisateur comme « TITRE » ou « Titre ». La propriété text-transform admet quatre valeurs possibles :

**capitalize**

Elle capitalise la première lettre de chaque mot. Par exemple, « marcel dupond » deviendra « Marcel Dupond ».

**uppercase**

Elle convertit toutes les lettres en majuscules. Par exemple, « marcel dupond » deviendra « MARCEL DUPOND ».

**lowercase**

Elle convertit toutes les lettres en minuscules. Par exemple, « MARCEL DUPOND » deviendra « marcel dupond ».

**none**

Aucune transformation, et le texte se présente tel qu'il apparaît dans le code HTML.

Pour exemple, nous utiliserons une liste de noms, balisés avec des éléments <li> (list-item). Disons que nous voulons que les noms soient capitalisés et les titres soient en lettres majuscules. Examinez le code HTML de cet exemple et vous constaterez que le texte est en fait en minuscules.

```
h1 {  
    text-transform: uppercase;  
}  
  
li {  
    text-transform: capitalize;  
}
```

## **Leçon 6 : Les liens**

Vous pouvez appliquer aux liens ce que vous avez déjà vu dans les leçons précédentes (c'est-à-dire à changer les couleurs et les polices, les soulignés, etc.). La nouveauté, c'est que CSS permet de définir ces propriétés différemment, selon que le lien est visité, non visité, activé, ou si le curseur le survole. Cela permet d'ajouter des effets plaisants et utiles à vos sites Web. Pour contrôler ces effets, on utilise ce qu'on appelle des pseudo-classes.

### **Qu'est-ce qu'une pseudo-classe ?**

Une pseudo-classe permet de tenir compte de conditions et événements différents pour la définition d'une propriété sur une balise HTML.

Voyons un exemple. Comme vous le savez, les liens sont définis en HTML avec des balises `<a>`. On peut donc utiliser `a` comme sélecteur dans CSS :

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

Un lien est susceptible d'avoir plusieurs états. Par exemple, il peut être visité ou non. Vous pouvez utiliser des pseudo-classes pour assigner des styles différents aux liens visités et non visités.

```
a:link {  
    color: blue;  
}  
  
a:visited {  
    color: red;  
}
```

Utilisez respectivement `a:link` et `a:visited` pour les liens non visités et pour ceux visités. Les liens actifs ont pour pseudo-classe `a:active`, et la pseudo-classe `a:hover` intervient lorsque le curseur survole le lien.

Nous allons maintenant examiner chacune des quatre pseudo-classes avec des exemples et des explications supplémentaires.

### **La pseudo-classe `:link`**

La pseudo-classe `:link` est utilisée pour les liens menant aux pages que l'utilisateur n'a pas visitées.

Dans le code suivant, les liens non visités seront en bleu clair.

```
a:link {  
    color: #6699CC;  
}
```

### **La pseudo-classe `:visited`**

La pseudo-classe `:visited` est utilisée pour les liens menant aux pages que l'utilisateur a visitées. Par exemple, le code suivant donnera la couleur mauve foncé à tous les liens visités :

```
a:visited {  
    color: #660099;  
}
```

### La pseudo-classe :active

La pseudo-classe :active est utilisée pour les liens qui sont activés.

Cet exemple montre des liens activés dont la couleur d'arrière-plan est jaune :

```
a:active {  
    background-color: #FFFF00;  
}
```

### La pseudo-classe :hover

La pseudo-classe :hover est utilisée lorsque le pointeur de la souris survole un lien.

On peut s'en servir pour créer des effets intéressants. Par exemple, si nous voulons que nos liens soient orange et en italiques au survol du pointeur, le code CSS devrait être le suivant :

```
a:hover {  
    color: orange;  
    font-style: italic;  
}
```

### Exemple 1 : Effet au survol du lien par le curseur

Créer des effets différents pour le survol du pointeur sur les liens est très populaire. Nous examinerons donc quelques exemples de plus pour la pseudo-classe :hover.

#### Exemple 1a : Espacement des lettres

Comme indiqué dans la leçon 5, on peut ajuster l'espacement entre les lettres avec la propriété letter-spacing. Cela s'applique aux liens pour un effet spécial :

```
a:hover {  
    letter-spacing: 10px;  
    font-weight: bold;  
    color: red;  
}
```

#### Exemple 1b : MAJUSCULES et minuscules

Dans la leçon 5, nous avons vu la propriété text-transform qui permet de basculer entre des lettres majuscules et minuscules. Elle peut aussi servir pour créer un effet sur les liens :

```
a:hover {  
    text-transform: uppercase;  
    font-weight: bold;  
    color: blue;  
    background-color: yellow;  
}
```

Les deux exemples donnent un aperçu des possibilités de combinaison des différentes propriétés presque illimitées. Vous pouvez créer vos propres effets, essayez- donc !

### Exemple 2 : Supprimer le trait sous les liens

On pose souvent la question : « Comment supprimer le trait sous les liens ? »

Vous devriez évaluer soigneusement s'il est nécessaire de supprimer le souligné car cela pourrait réduire considérablement la convivialité de votre site Web. Le public est habitué aux liens bleus soulignés des pages Web et sait qu'on peut cliquer dessus. Même ma maman le sait ! Si vous changez le souligné et la couleur des liens, il y a de fortes chances que l'utilisateur s'embrouille et donc n'exploite pas tout le contenu de votre site Web.

Ceci dit, il est très facile de supprimer le souligné des liens. Comme vous vous en rappelez de la leçon 5, on peut utiliser la propriété `text-decoration` pour indiquer de souligner le texte ou non. Pour supprimer le souligné, il suffit de lui donner la valeur `none`.

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

Autrement, vous pouvez fixer la valeur de `text-decoration` en même temps que celles d'autres propriétés pour les quatre pseudo-classes.

```
a:link {  
    color: blue;  
    text-decoration: none;  
}  
  
a:visited {  
    color: purple;  
    text-decoration: none;  
}  
  
a:active {  
    background-color: yellow;  
    text-decoration: none;  
}  
  
a:hover {  
    color: red;  
    text-decoration: none;  
}
```

## Résumé

Dans cette leçon, vous avez appris les pseudo-classes, en utilisant quelques propriétés vues dans les leçons précédentes. Cela devrait vous donner une idée des possibilités offertes par CSS.



## Leçon 7 : L'identification et le regroupement d'éléments (class et id)

Parfois, on veut appliquer un style spécial à un élément ou à un groupe d'éléments particuliers. Dans cette leçon, nous étudierons comment utiliser les attributs class et id pour définir les propriétés d'éléments sélectionnés.

Comment donner à un titre particulier une autre couleur que celle des autres titres du site Web ? Comment regrouper les liens en différentes catégories et donner à chacune un style spécial ? Ce sont les types de questions auxquelles nous répondrons ici.

### Le regroupement d'éléments dans une classe

Supposons que nous ayons deux listes de liens vers les différents cépages utilisés pour des vins blancs et des vins rouges. Voici à quoi ressemble le code HTML :

```
<p>Cépages de vins blancs :</p>
<ul>
<li><a href="ri.htm">Riesling</a></li>
<li><a href="ch.htm">Chardonnay</a></li>
<li><a href="pb.htm">Pinot Blanc</a></li>
</ul>

<p>Cépages de vins rouges :</p>
<ul>
<li><a href="cs.htm">Cabernet Sauvignon</a></li>
<li><a href="me.htm">Merlot</a></li>
<li><a href="pn.htm">Pinot Noir</a></li>
</ul>
```

Nous voulons ensuite que les liens des vins blancs soient jaunes, que ceux des vins rouges soient rouges, et que les autres liens de la page Web restent bleus.

Pour y parvenir, nous répartissons les liens en deux catégories. On assigne une classe à chaque lien avec l'attribut class.

Essayons de définir quelques classes pour l'exemple précédent :

```
<p>Cépages de vins blancs :</p>
<ul>
<li><a href="ri.htm" class="whitewine">Riesling</a></li>
<li><a href="ch.htm" class="whitewine">Chardonnay</a></li>
<li><a href="pb.htm" class="whitewine">Pinot Blanc</a></li>
</ul>

<p>Cépages de vins rouges :</p>
<ul>
<li><a href="cs.htm" class="redwine">Cabernet Sauvignon</a></li>
<li><a href="me.htm" class="redwine">Merlot</a></li>
<li><a href="pn.htm" class="redwine">Pinot Noir</a></li>
</ul>
```

Par la suite, on peut définir des propriétés spéciales pour les liens appartenant respectivement aux catégories « whitewine » et « redwine ».

```
a {
    color: blue;
}

a.whitewine {
    color: #FFBB00;
}

a.redwine {
    color: #800000;
}
```

Comme affiché dans l'exemple, on peut définir les propriétés des éléments appartenant à une certaine classe en utilisant `.le_nom_de_ladite_classe` dans la feuille de style du document.

## L'identification d'un élément avec id

Outre le regroupement d'éléments, on peut avoir besoin d'identifier un élément unique. Pour cela, on utilise l'attribut `id`.

L'attribut `id` a ceci de spécial qu'il ne peut pas y avoir deux éléments avec le même `id` dans le document. Chaque `id` doit être unique. Dans le cas contraire, il faudrait utiliser l'attribut `class` à la place. Voyons maintenant un exemple d'utilisation possible de `id` :

```
<h1>Chapitre 1</h1>
....
<h2>Chapitre 1.1</h2>
....
<h2>Chapitre 1.2</h2>
....
<h1>Chapitre 2</h1>
....
<h2>Chapitre 2.1</h2>
....
<h3>Chapitre 2.1.2</h3>
....
```

Il pourrait s'agir des titres d'un document partagé en chapitres ou en paragraphes. Il semble naturel d'assigner un `id` à chaque chapitre comme ceci :

```
<h1 id="c1">Chapitre 1</h1>
....
<h2 id="c1-1">Chapitre 1.1</h2>
....
<h2 id="c1-2">Chapitre 1.2</h2>
....
<h1 id="c2">Chapitre 2</h1>
....
<h2 id="c2-1">Chapitre 2.1</h2>
....
<h3 id="c2-1-2">Chapitre 2.1.2</h3>
....
```

Disons que le titre du chapitre 1.2 doit apparaître en rouge. Avec CSS, cela donne :

```
#c1-2 {  
    color: red;  
}
```

Comme affiché dans l'exemple précédent, on peut définir les propriétés d'un élément spécifique en utilisant `#le_nom_dudit_id` dans la feuille de style du document.

## Résumé

Dans cette leçon, nous avons vu que l'on pouvait définir les propriétés d'éléments spécifiques à l'aide des sélecteurs class et id.

## **Leçon 8 : Le regroupement d'éléments (span et div)**

Les éléments `<span>` et `<div>` servent à regrouper et structurer un document et seront souvent utilisés avec les attributs `class` et `id`.

Dans cette leçon, nous verrons de plus près l'utilisation des deux éléments `<span>` et `<div>`, et leur importance centrale pour CSS.

- Le regroupement avec `<span>`
- Le regroupement avec `<div>`

### **Le regroupement avec `<span>`**

L'élément `<span>` est ce qu'on pourrait appeler un élément neutre, qui n'ajoute rien au document en lui-même. Mais CSS peut utiliser l'élément `<span>` pour ajouter des caractéristiques visuelles à des parties spécifiques du texte des documents.

Cette citation de Benjamin Franklin en constitue un exemple :

`<p>`Tôt couché et tôt levé  
fortifie, enrichit et rend avisé.`</p>`

Supposons que les avantages aux yeux de M. Franklin de ne pas gaspiller son sommeil soient accentués en rouge. Dans ce but, nous pouvons baliser les avantages avec des éléments `<span>`. Chaque `span` reçoit alors un attribut `class` défini ainsi dans la feuille de style :

```
<p>Tôt couché et tôt levé  
<span class="benefit">fortifie</span>,  
<span class="benefit">enrichit</span>  
et <span class="benefit">rend avisé</span>.</p>
```

Le code CSS associé :

```
span.benefit {  
    color:red;  
}
```

Bien sûr, vous pouvez aussi utiliser l'attribut `id` pour ajouter du style aux éléments `<span>`, tant que vous appliquez un `id` unique à chacun des trois éléments `<span>`, comme appris dans la leçon précédente.

### **Le regroupement avec `<div>`**

Alors que `<span>` s'utilise dans un élément de type bloc comme vu précédemment, `<div>` sert à regrouper un ou plusieurs éléments de type bloc.

Hormis cette différence, le regroupement avec `<div>` fonctionne plus ou moins de la même façon. Voyons un exemple avec deux listes de présidents des États-Unis répartis selon leur affiliation politique :

```
<div id="democrats">  
<ul>  
<li>Franklin D. Roosevelt</li>  
<li>Harry S. Truman</li>  
<li>John F. Kennedy</li>  
<li>Lyndon B. Johnson</li>  
<li>Jimmy Carter</li>  
<li>Bill Clinton</li>  
</ul>  
</div>
```

```
<div id="republicans">
<ul>
<li>Dwight D. Eisenhower</li>
<li>Richard Nixon</li>
<li>Gerald Ford</li>
<li>Ronald Reagan</li>
<li>George Bush</li>
<li>George W. Bush</li>
</ul>
</div>
```

Dans notre feuille de style, nous pouvons utiliser le regroupement exactement de la même façon que précédemment :

```
#democrats {
    background:blue;
}

#republicans {
    background:red;
}
```

Dans les exemples précédents, nous avons utilisé les éléments `<div>` et `<span>` pour des choses très simples telles que les couleurs du texte et de l'arrière-plan. Ces deux éléments ont le potentiel pour des choses beaucoup plus évoluées, mais nous ne les abordons pas ici. Cela interviendra plus loin dans le tutoriel.

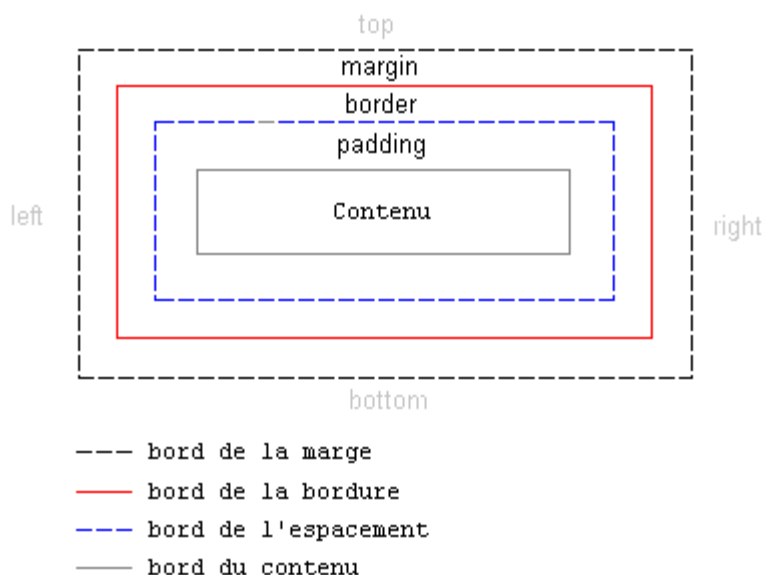
## Résumé

Dans les leçons 7 et 8, vous avez vu les sélecteurs `id` et `class`, et les éléments `span` et `div`.

## Leçon 9 : Le modèle des boîtes

Le modèle des boîtes de CSS décrit les boîtes générées pour les éléments HTML. Il contient également des options détaillées pour l'ajustement des marges, des bordures, des espacements et du contenu de chaque élément. Le diagramme suivant montre la structure du modèle des boîtes :

### Le modèle des boîtes dans CSS



L'illustration ci-dessus peut sembler très théorique, essayons-donc d'utiliser le modèle dans un cas réel avec un titre et du texte. Le code HTML de notre exemple est le suivant (d'après la Déclaration universelle des droits de l'homme) :

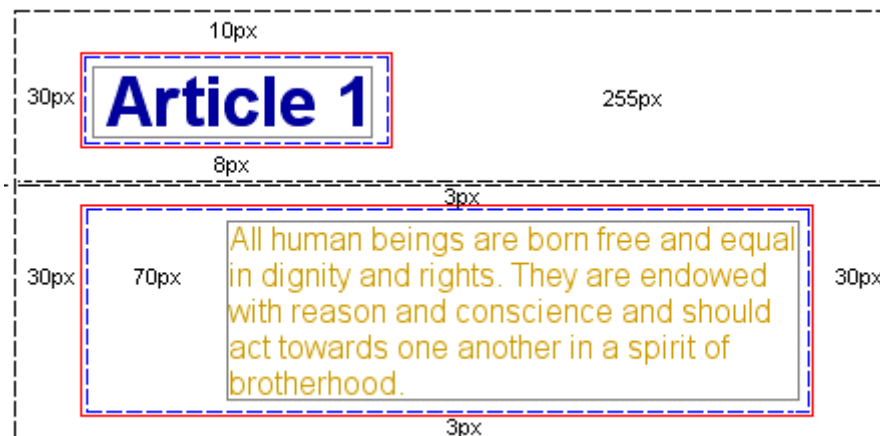
```
<h1>Article 1:</h1>
<p>All human beings are born free
and equal in dignity and rights.
They are endowed with reason and conscience
and should act towards one another in a
spirit of brotherhood</p>
```

En ajoutant de la couleur et des informations de police, l'exemple pourrait se présenter comme suit :

## Article 1

All human beings are born free and equal  
in dignity and rights. They are endowed  
with reason and conscience and should  
act towards one another in a spirit of  
brotherhood.

L'exemple contient deux éléments : `<h1>` et `<p>`. On peut illustrer le modèle de boîtes des deux éléments de la façon suivante :



Quoique cela puisse sembler un peu compliqué, le schéma montre chaque élément HTML entouré par des boîtes. Des boîtes que nous pouvons ajuster avec CSS.

## Résumé

Cette leçon introduisait le modèle des boîtes. Dans les trois leçons suivantes, nous étudierons la création et le contrôle des éléments dans le modèle des boîtes.

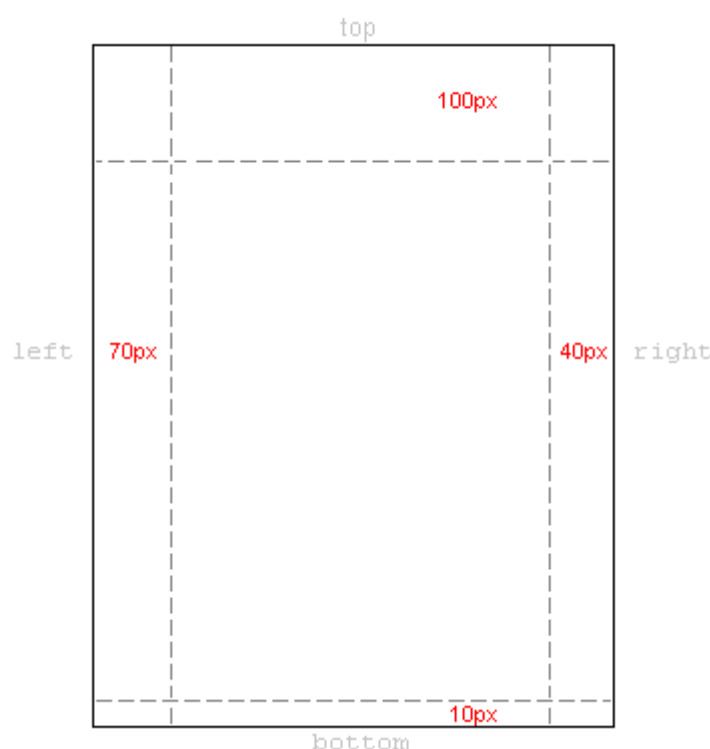
## Leçon 10 : Margin et padding

Dans la leçon précédente, nous introduisons le modèle des boîtes. Dans cette leçon, nous verrons comment changer la présentation des éléments avec les propriétés margin et padding.

### Fixer la marge d'un élément

Un élément a quatre côtés : droit ("right"), gauche ("left"), supérieur ("top") et inférieur ("bottom"). La marge est la distance entre chaque côté et l'élément avoisinant (ou les bordures du document). Cf. également l'illustration dans la leçon 9.

Comme premier exemple, nous allons voir comment définir les marges du document même, c'est-à-dire de l'élément <body>. L'illustration suivante montre les marges voulues pour les pages.



Le code CSS serait le suivant :

```
body {  
    margin-top: 100px;  
    margin-right: 40px;  
    margin-bottom: 10px;  
    margin-left: 70px;  
}
```

Ou bien plus élégant et concis :

```
body {  
    margin: 100px 40px 10px 70px;  
}
```



Vous pouvez fixer pareillement les marges sur presque tous les éléments. Par exemple, on peut choisir de définir les marges de tous les paragraphes de texte balisés avec <p> :

```
body {  
    margin: 100px 40px 10px 70px;  
}  
  
p {  
    margin: 5px 50px 5px 50px;  
}
```

## Fixer l'espacement dans un élément

L'espacement peut aussi être assimilé à un « remplissage ». Cela peut se comprendre dans la mesure où l'espacement n'affecte pas la distance de l'élément aux autres éléments mais définit seulement la distance interne entre la bordure et le contenu de l'élément.

L'utilisation de l'espacement peut être illustrée par ce simple exemple où tous les titres ont des couleurs d'arrière-plan :

```
h1 {  
    background: yellow;  
}  
  
h2 {  
    background: orange;  
}
```

En définissant l'espacement des titres, on change le remplissage autour du texte dans chaque titre :

```
h1 {  
    background: yellow;  
    padding: 20px 20px 20px 80px;  
}  
  
h2 {  
    background: orange;  
    padding-left: 120px;  
}
```

## Résumé

Vous commencez à maîtriser le modèle des boîtes dans CSS. Dans la leçon suivante, nous étudierons de plus près comment colorier les différentes bordures et comment dessiner le contour des éléments.

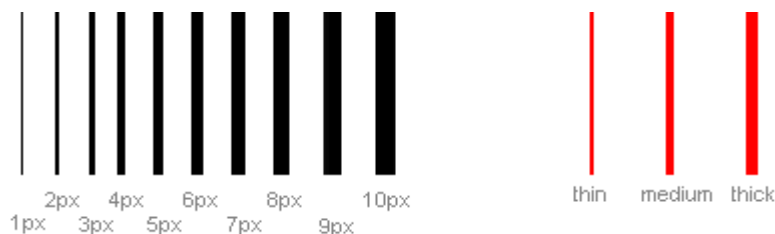
## Leçon 11 : Les bordures

On peut utiliser des bordures pour beaucoup de choses, par exemple comme élément décoratif ou pour souligner la séparation de deux choses. CSS offre des options à n'en plus finir pour l'utilisation des bordures dans vos pages.

- **border-width**
- **border-color**
- **border-style**
- **Exemples de définitions de bordures**
- **border**

### L'épaisseur des bordures [border-width]

L'épaisseur des bordures est définie par la propriété border-width, dont les valeurs peuvent être "thin", "medium" et "thick", ou une valeur numérique en pixels. La figure ci-dessous illustre le système :



### La couleur des bordures [border-color]

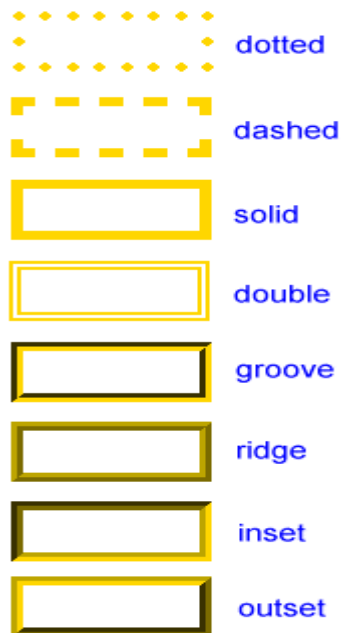


La propriété border-color définit la couleur d'une bordure. Des valeurs de couleur normales sont, par exemple, "#123456", "rgb(123,123,123)" ou "yellow".

### Les types de bordures [border-style]

On peut choisir parmi plusieurs types de bordures. Voici huit types de bordures différentes tels qu'Internet Explorer 5.5 les interprète. Toutes les bordures ont la couleur "gold" et une épaisseur valant "thick" mais elles peuvent naturellement s'afficher dans d'autres couleurs et épaisseurs.

On peut utiliser les valeurs "none" ou "hidden" si on ne veut aucune bordure.



## Exemples de définitions de bordures

Les trois propriétés décrites précédemment peuvent être réunies pour chaque élément et donc produire des bordures différentes. Pour illustrer cela, voyons un document où nous définissons des bordures différentes pour `<h1>`, `<h2>`, `<ul>` et `<p>`. Le résultat n'est peut-être pas très beau mais il montre quelques-unes des nombreuses possibilités :

```
h1 {  
    border-width: thick;  
    border-style: dotted;  
    border-color: gold;  
}  
  
h2 {  
    border-width: 20px;  
    border-style: outset;  
    border-color: red;  
}  
  
p {  
    border-width: 1px;  
    border-style: dashed;  
    border-color: blue;  
}  
  
ul {  
    border-width: thin;  
    border-style: solid;  
    border-color: orange;  
}
```

Il est également possible de déclarer des propriétés spéciales pour les bordures supérieures, inférieures, gauches ou droites. L'exemple suivant montre comment le faire :

```
h1 {  
    border-top-width: thick;  
    border-top-style: solid;  
    border-top-color: red;  
  
    border-bottom-width: thick;  
    border-bottom-style: solid;  
    border-bottom-color: blue;  
  
    border-right-width: thick;  
    border-right-style: solid;  
    border-right-color: green;  
  
    border-left-width: thick;  
    border-left-style: solid;  
    border-left-color: orange;  
}
```

## Concision [border]

Comme pour beaucoup d'autres propriétés, on peut concentrer plusieurs propriétés en une seule avec border. Voici un exemple :

```
p {  
    border-width: 1px;  
    border-style: solid;  
    border-color: blue;  
}
```

Il peut se concentrer en :

```
p {  
    border: 1px solid blue;  
}
```

## Résumé

Dans cette leçon, vous avez eu un aperçu des options illimitées offertes par CSS pour les bordures de vos pages.

## **Leçon 12 : La hauteur et la largeur**

Nous ne nous sommes pas beaucoup préoccupés, jusqu'à maintenant, des dimensions des éléments avec lesquels nous avons travaillé. Dans cette leçon, nous verrons comment définir facilement la hauteur et la largeur d'un élément.

### **Fixer la largeur [width]**

La propriété width permet de définir la largeur d'un élément.

L'exemple simplifié ci-dessous nous montre une boîte où on peut taper du texte :

```
div.box {  
    width: 200px;  
    border: 1px solid black;  
    background: orange;  
}
```

### **Fixer la hauteur [height]**

Dans l'exemple précédent, remarquez la hauteur de la boîte fixée par son contenu. On peut affecter la hauteur d'un élément avec la propriété height. Essayons de rendre la boîte de l'exemple haute de 500px :

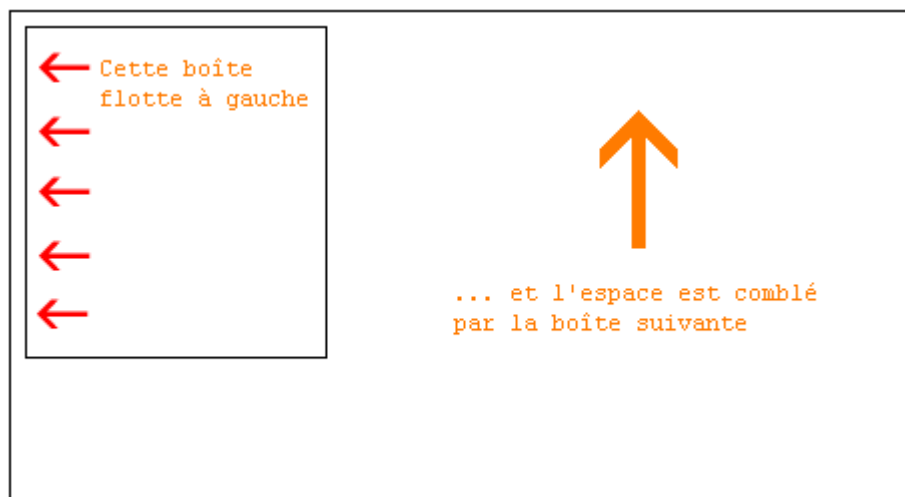
```
div.box {  
    height: 500px;  
    width: 200px;  
    border: 1px solid black;  
    background: orange;  
}
```

## **Résumé**

Les leçons 9, 10, 11 et 12 ont introduit le modèle des boîtes dans CSS. Comme vous pouvez le constater, le modèle des boîtes offre beaucoup d'options nouvelles. Vous auriez pu utiliser des tables HTML pour vos dispositions, mais avec CSS et le modèle des boîtes vous devez maintenant être capable d'obtenir des présentations élégantes, plus précises et conformes aux recommandations du W3C.

## Leçon 13 : Les éléments flottants (les flottants)

Un élément peut avoir une position flottante à droite ou à gauche avec la propriété float. C'est-à-dire que la boîte et son contenu peut flotter soit à droite, soit à gauche dans un document (ou dans la boîte conteneur), cf. la description du modèle des boîtes dans la leçon 9. La figure suivante illustre le principe :



Si par exemple nous voulions qu'un texte s'enroule autour d'une image, le résultat serait le suivant :



### Comment ça fonctionne ?

Voici le code HTML de l'exemple précédent :

```
<div id="picture">
  
</div>
```

```
<p>causas naturales et antecedentes,
idcirco etiam nostrarum voluntatum...</p>
```

Pour que l'image flotte à gauche et que le texte l'entoure, il suffit de définir la largeur de la boîte entourant l'image puis de donner à la propriété float la valeur "left" :

```
#picture {  
    float:left;  
    width: 100px;  
}
```

## Un autre exemple avec des colonnes

Les flottants peuvent aussi servir pour le colonnage d'un document. Pour créer les colonnes, il suffit de les structurer dans le code HTML avec des éléments <div> de la façon suivante :

```
<div id="column1">  
    <p>Haec disserens qua de re agatur  
    et in quo causa consistat non videt...</p>  
</div>  
  
<div id="column2">  
    <p>causas naturales et antecedentes,  
    idcirco etiam nostrarum voluntatum...</p>  
</div>  
  
<div id="column3">  
    <p>nam nihil esset in nostra  
    potestate si res ita se haberet...</p>  
</div>
```

Une fois fixée la largeur des colonnes (par exemple à 33%), on fait simplement flotter chaque colonne à gauche en définissant la propriété float :

```
#column1 {  
    float:left;  
    width: 33%;  
}  
  
#column2 {  
    float:left;  
    width: 33%;  
}  
  
#column3 {  
    float:left;  
    width: 33%;  
}
```

La propriété float admet les valeurs "left", "right" ou "none".

## La propriété clear

la propriété clear sert à contrôler le comportement des éléments qui viennent après les éléments flottants dans le document.

Par défaut, les éléments à suivre remontent pour combler l'espace disponible libéré lorsqu'une boîte flotte d'un côté. Examinez l'exemple précédent dans lequel le texte se place automatiquement sous la photo de Bill Gates.

La propriété clear admet les valeurs "left", "right", "both" ou "none". Le principe est tel que, par exemple, si la valeur de clear d'une boîte est fixée à "both", la bordure supérieure de la marge de cette boîte se trouvera toujours sous la bordure de la marge inférieure des boîtes flottantes éventuelles venant du dessus.

```
<div id="picture">
  
</div>

<h1>Bill Gates</h1>

<p class="floatstop">causas naturales et antecedentes,
idcirco etiam nostrarum voluntatum...</p>
```

Pour empêcher que le texte ne flotte à côté de l'image, nous pouvons ajouter le code CSS suivant :

```
#picture {
  float:left;
  width: 100px;
}

.floatstop {
  clear:both;
}
```

## Résumé

Les flottants sont très utiles dans beaucoup de situations, et on les utilisera souvent en même temps qu'un positionnement. Dans la leçon suivante, nous étudierons comment placer une boîte en position relative ou bien absolue.

### Référence

<http://fr.html.net/tutorials/css/>