



Cours HTML

3WAcademy

Cours d'intégration Web

1. Présentation générale du langage

1.1. Implémentation

1.1.1. Utilisation de l'attribut style

1.1.2. Utilisation de la balise style dans le HEAD du document HTML

1.1.3. Utilisation d'une feuille de style externe

1.2. Les commentaires

1.3. Présentation des déclarations CSS

1.4. Les sélecteurs

1.4.1. Le sélecteur de balise

1.4.2. Les classes

1.4.3. Les identifiants

1.4.4. Le sélecteur universel

1.4.5. Sélection multiple

1.4.6. Les descendants

1.4.7. Les enfants directs

1.4.8. Les éléments adjacents

1.4.9. Les éléments de la même fratrie

1.4.10. Les pseudo-classes

1.4.11. Les pseudos éléments

::before

::after

::first-letter

::first-line

1.5. Les valeurs et les différentes unités

Les unités de longueur absolues

Les principales unités de longueur relatives

1. Présentation générale du langage

Le CSS est l'abréviation de l'anglais Cascading Style Sheets, littéralement feuilles de style en cascades. C'est un langage qui permet de décrire la présentation d'un document HTML. Ce n'est pas un langage de programmation, dit autrement, ce langage est interprété par le navigateur.

Le CSS permet avant tout de séparer le plus possible la mise en forme et la présentation du document de son contenu, celui-ci étant décrit à l'aide du HTML. Cette séparation offre beaucoup d'avantages, elle favorise une meilleure structure du code HTML, elle favorise une meilleure accessibilité, elle permet de changer plus facilement de présentation et permet également d'améliorer la maintenance du code du site internet avec la possibilité de travailler séparément la présentation du contenu.

Le CSS permet d'ajouter plusieurs feuilles de styles au même document (la cascade). De plus, certaines des propriétés peuvent être héritées à travers la structure du document HTML afin d'éviter de les déclarer plusieurs fois.

1.1. Implémentation

Il existe trois possibilités pour implémenter du CSS

1.1.1. Utilisation de l'attribut style

Par exemple, pour mettre le texte d'un paragraphe en rouge nous pouvons utiliser ce code:

```
<p style="color:red">Du texte contenu dans un paragraphe</p>
```

Le style CSS est directement placé comme valeur de l'attribut *style*.

Cette pratique est fortement déconseillée puisque dans ce cas, la mise en forme se retrouve mélangée avec le contenu, à l'inverse de ce que l'on souhaite.

Cependant, ce genre de code reste fréquent puisque, du style peut être généré dynamiquement par du JavaScript sous cette forme.

1.1.2. Utilisation de la balise style dans le HEAD du document HTML

Par exemple, pour mettre le texte de tous les paragraphes du document en rouge nous pouvons utiliser ce code:

```
<style>
p {
  color: red;
}
</style>
```

Le style CSS est directement placé dans une balise **STYLE** elle même placée dans le **HEAD** du document HTML.

Le code est alors organisé sous la forme d'un groupe de propriétés dédiées à un sélecteur. Le sélecteur (ici la balise p) permet de spécifier pour quel(s) élément(s) s'applique la ou les propriétés.

Cette pratique est recommandée dans le cas où le code HTML ne concerne que le document dans lequel il se trouve.

1.1.3. Utilisation d'une feuille de style externe

Par exemple, pour mettre le texte de tous les paragraphes du document en rouge nous pouvons utiliser ce code:

```
p {
  color: red;
}
```

Code écrit dans un fichier nommé style.css

... et la déclaration suivante dans le(s) document(s) HTML concerné(s):

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

Une version plus complète est parfois visible:

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

Cette déclaration, sous la forme d'une balise LINK est à placer dans le HEAD de tous les documents HTML concernés. Seule la mention *rel="stylesheet"* est obligatoire (en plus du href).

Cette pratique est la plus recommandée surtout si vous voulez mutualiser le style entre plusieurs documents HTML.

1.2. Les commentaires

Il est possible d'utiliser des commentaires en CSS

```
/* Commentaire sur une ligne */  
  
/* Commentaire sur  
plusieurs lignes */
```

1.3. Présentation des déclarations CSS

Chaque déclaration se compose ainsi:

```
sélecteur {  
    nom-propriété-1: valeur-1;  
    nom-propriété-2: valeur2;  
}
```

Le sélecteur permet de spécifier pour quel(s) élément(s) s'applique la ou les propriétés.

Le groupe de propriétés doit être déclaré entre des accolades ouvrantes et fermantes. Chaque propriété se compose du nom de celle-ci, puis de la valeur séparé du nom par le caractère : (deux points). Enfin chaque propriété doit être terminée par un ; (point virgule).

1.4. Les sélecteurs

La séparation de la mise en forme du contenu repose sur la large gamme de sélecteurs offerte par le CSS. Ils permettent en effet de cibler précisément des éléments du contenus à l'aide une syntaxe très intuitive.

Certains sélecteurs permettent même de cibler des éléments dynamiquement: les pseudo classes et les pseudo éléments.

1.4.1. Le sélecteur de balise

En nommant une balise en particulier, ici la balise `article`, on applique les propriétés à l'ensemble des balises correspondante de l'ensemble du document.

Dans le CSS

```
article {  
    ...  
}
```

Dans le document HTML

```
...  
<article>Un article...</article>  
...  
<article>Un autre article...</article>  
...
```

1.4.2. Les classes

Les classes permettent de cibler très précisément le ou les éléments ayant l'attribut `class` correspondant. Le nom d'une classe ne peut comprendre que des lettres, des chiffres, le tiret et le trait de soulignement, il ne peut pas commencer par un chiffre.

Le nom de la classe est précédé par un point dans le code CSS. Il ne faut mettre le point dans l'attribut HTML.

Dans le CSS

```
.colored {  
    ...  
}
```

Dans le document HTML

```
...  
<section class="colored">...</section>  
...  
<span class="colored">...</span>  
...
```

Il est possible de définir plusieurs classes au même élément en séparant leur nom par un espace dans l'attribut *class* de celui-ci.

```
<section class="colored text-centered">...</section>
```

1.4.3. Les identifiants

Il est possible de cibler un élément par son identifiant, c'est à dire son attribut *id*. A l'inverse d'une classe, l'identifiant d'un élément doit être unique pour l'ensemble du document HTML.

Les identifiants sont très souvent utilisé par le JavaScript. Il servent également comme ancre. C'est pourquoi, il est conseillé de ne pas utiliser d'identifiant comme sélecteur et de préférer l'utilisation des classes.

Comme les classes, le nom de l'identifiant ne peut comprendre que des lettres, des chiffres, le tiret et le trait de soulignement, il ne peut pas commencer par un chiffre.

Le nom de l'identifiant est précédé par le symbole # dans le code CSS. Il ne faut mettre le # dans l'attribut HTML.

Dans le CSS

```
#menu {  
    ...  
}
```

Dans le document HTML

```
<nav id="menu">...</nav>
```

1.4.4. Le sélecteur universel

Il est possible de cibler chaque élément du document avec le sélecteur universel ***.

Dans le CSS

```
* {  
    ...  
}
```

Dans le document HTML, tous les éléments seront concernés.

1.4.5. Sélection multiple

Il est possible d'appliquer les mêmes règles à plusieurs sélecteurs simultanément, il faut pour cela les séparer par une virgule.

Dans le CSS

```
h1, h2, h3 {  
    ...  
}
```

1.4.6. Les descendants

Il est possible de combiner plusieurs sélecteurs simples. Pour cibler certains éléments descendants d'un élément, on combine le sélecteur du parent et celui des enfants séparés par un espace.

Par exemple, afin d'éviter de mettre des classes à tous les liens d'un menu, on peut utiliser la notation suivante:

Dans le CSS

```
.menu a {  
    ...  
}
```

Dans le document HTML

```
<nav class="menu">  
    <a href="home.html">Home</a>  
    <a href="portfolio.html">Portfolio</a>  
    <a href="contact.html">Contact us</a>  
</nav>
```

Ce sélecteur permet de cibler l'ensemble des liens (balise A) du menu parent dont la classe est "menu".

Cette notation fonctionne également avec une structure plus complexe:

Dans le document HTML

```
<nav class="menu">  
    <ul>  
        <li><a href="home.html">Home</a></li>
```

```
    <li><a href="portfolio.html">Portfolio</a></li>
    <li><a href="contact.html">Contact us</a></li>
  </ul>
</nav>
```

Ici, les liens seront également ciblés bien que présents dans une liste.

1.4.7. Les enfants directs

Dans certains cas, le sélecteur des enfants est trop imprécis, ciblant l'ensemble des éléments enfants quelque soit le parent. On peut alors utiliser le sélecteur des enfants directs qui ne cible que les éléments directement inclus dans le sélecteur parent.

Par exemple

Dans le CSS

```
article > p {
  ...
}
```

Dans le document HTML

```
<article>
  <p>Lorem ipsum ... </p>
  <aside>
    <p>Lorem ipsum ... </p>
    <p>Lorem ipsum ... </p>
  </aside>
  <p>Lorem ipsum ... </p>
</nav>
```

Ici seuls les paragraphes en rouge sont ciblés, pas ceux contenus dans l'élément aside.

1.4.8. Les éléments adjacents

Pour cibler l'élément directement adjacent (au sein de la même fratrie). Très utile pour appliquer un style aux éléments d'une fratrie, excepté pour le premier, par exemple afficher une bordure entre les éléments d'une liste.

Il ne faut pas répéter tout le sélecteur après le + mais seulement la partie concernant l'élément adjacent.

Dans le CSS

```
.menu li + li {  
    ...  
}
```

Dans le document HTML

```
<nav class="menu">  
    <ul>  
        <li><a href="home.html">Home</a></li>  
        <li><a href="portfolio.html">Portfolio</a></li>  
        <li><a href="contact.html">Contact us</a></li>  
    </ul>  
</nav>
```

1.4.9. Les éléments de la même fratrie

Pour cibler tous les éléments adjacent au sein de la même fratrie.

Dans le CSS

```
h1 ~ p {  
    ...  
}
```

Dans le document HTML

```
<article>  
    <h1> ... </h1>  
    <p>Lorem ipsum ... </p>  
    <aside> ... </aside>  
    <p>Lorem ipsum ... </p>  
</nav>
```

Ici, tous les paragraphes de la même fratrie sont sélectionnés.

1.4.10. Les pseudo-classes

Une pseudo-classe est un sélecteur CSS particulier qui permet de cibler des éléments dans certaines conditions comme si une classe leur était définie automatiquement.

Par exemple la pseudo-classe `:hover` permet de sélectionner les éléments survolé comme si ils avait une classe `.hover` ajoutée automatiquement lors du survol.

Les pseudo classes sont prédéfinies, leur nom est précédé du symbole "deux points" dans la déclaration CSS.

Dans le CSS

```
a:hover {  
    ...  
}
```

Liste des principales pseudo-classes:

| Nom | Description / exemple |
|-------------------------|--|
| <i>:link</i> | Permet de définir les règles qui s'appliquent à un lien quand celui ci n'est ni survolé, ni cliqué et pas encore visité. <i>a:link { ... }</i> |
| <i>:hover</i> | Appliqué lorsque l'élément est survolé. Pour les liens et les boutons (de préférence) mais fonctionne avec tous les éléments. <i>a:hover { ... }</i> |
| <i>:active</i> | Appliqué lorsque le lien ou le bouton est cliqué. <i>a:active { ... }</i> |
| <i>:visited</i> | Appliqué lorsque le lien a été visité, on a cliqué dessus. <i>a:visited { ... }</i> |
| <i>:first-child</i> | Cible le premier élément de la même fratrie si il correspond à la balise spécifiée (la balise LI dans l'exemple). <i>li:first-child { ... }</i> |
| <i>:last-child</i> | Cible le dernier élément de la même fratrie si il correspond à la balise spécifiée (la balise LI dans l'exemple). <i>li:last-child { ... }</i> |
| <i>:nth-child(expr)</i> | Cible le éléments de la même fratrie si il correspondent à la fois à l'expression donnée ainsi qu'à la balise spécifiée (la balise LI dans l'exemple). L'expression peut être un mots clés odd et even ou une toute expression de type A n + B (ou n représente l'index au sein de la fratrie, A et B sont des entiers). <i>li:nth-child(odd)</i> cible tous les éléments LI dont l'index est impair. <i>li:nth-child(even)</i> cible tous les éléments LI dont l'index est pair. <i>li:nth-child(4)</i> cible le quatrième élément LI. <i>li:nth-child(3n+2)</i> cible un élément LI tous les 3 éléments en |

| | |
|---------------------------|---|
| | <p>commençant au deuxième index.</p> <p><i>li:nth-child(odd) { ... }</i></p> |
| <i>:first-of-type</i> | <p>Cible le premier élément de la même fratrie correspondant à la balise spécifiée.</p> <p><i>p:first-of-type { ... }</i></p> |
| <i>:last-of-type</i> | <p>Cible le dernier élément de la même fratrie correspondant à la balise spécifiée.</p> <p><i>p:last-of-type { ... }</i></p> |
| <i>:nth-of-type(expr)</i> | <p>Cible les éléments de la même fratrie correspondant à la fois à l'expression donnée ainsi qu'à la balise spécifiée.</p> <p><i>p:nth-of-type(odd) { ... }</i></p> |
| <i>:not(expr)</i> | <p>Permet d'inverser une sélection.</p> <p><i>article:not(.large)</i> cible les articles qui n'ont pas la classe "large"</p> |

1.4.11. Les pseudos éléments

Un pseudo-élément est un sélecteur CSS particulier qui permet de cibler des éléments virtuellement dans certaines conditions prédéfinies.

Par exemple le pseudo-élément `::first-letter` permet de sélectionner la première lettre d'un contenu comme si elle était placée dans une balise spécifique.

Les pseudo-éléments sont prédéfinies, leur nom est précédé d'un double symbole "deux points" dans la déclaration CSS.

Dans le CSS

```
p::first-letter {
    ...
}
```

`::before`

Le pseudo-élément `::before` ajoute un sous élément à l'élément ciblé. Ce sous-élément est de type inline, par défaut. Il est placé **avant** le contenu de l'élément ciblé. Le contenu de ce sous élément est défini par la propriété *content* qui est obligatoire.

Dans le CSS

```
blockquote::before {  
    content: "«";  
    opacity: 0.5;  
}
```

::after

Le pseudo-élément `::after` ajoute un sous élément à l'élément ciblé. Ce sous-élément est de type inline, par défaut. Il est placé **après** le contenu de l'élément ciblé. Le contenu de ce sous élément est défini par la propriété *content* qui est obligatoire.

Dans le CSS

```
blockquote::after {  
    content: "»";  
    opacity: 0.5;  
}
```

::first-letter

Le pseudo-élément `::first-letter` permet de manipuler la première lettre de l'élément ciblé, par exemple, pour faire une lettrine.

Dans le CSS

```
p::first-letter {  
    float: left;  
    padding-right: 0.5em;  
    font-size: 2em;  
}
```

::first-line

Le pseudo-élément `::first-line` permet de manipuler la première ligne de l'élément ciblé.

Dans le CSS

```
p::first-line {  
    font-weight: bold;  
}
```

1.5. Les valeurs et les différentes unités

Il existe toute une variété de valeurs possible au sein du langage CSS.

Certaines propriétés acceptent des constantes prédéfinies, par exemple:

*text-decoration: **underline**;*

D'autres propriétés acceptent des couleurs:

*color: **red**;*

*color: **#FF0000**;*

*color: **rgb(255,0,0)**;*

*color: **rgba(255,0,0,0.1)**;*

D'autres propriétés acceptent des valeurs numériques avec unité:

*font-size: **20px**;*

... ou sans unité:

*line-height: **2**;*

Enfin certaines propriétés acceptent de fonctions:

*transform: **rotate(45deg)**;*

Les unités de longueur absolues

Pour les écrans:

| | |
|----|--|
| px | le pixel, qui dépend de la résolution de l'écran, permet un affichage net. |
|----|--|

Pour l'impression:

| | |
|----|---|
| cm | le centimètre. |
| mm | le millimètre. |
| in | le pouce (inch) 1in est égal à 2.54cm. |
| pt | le point. 1pt est égal à 1/72 de pouce. |
| pc | le picas. 1pc est égal à 12pt. |

Leur relation est 1in = 2.54cm = 25.4mm = 72pt = 12pc.

Les principales unités de longueur relatives

| | |
|-----|---|
| % | le pourcentage est: <ul style="list-style-type: none">• relatif à la taille de la fonte héritée pour la propriété <i>font-size</i>,• relatif à la taille de la font courante pour les autres propriétés liées au texte,• relatif à la largeur ou à la hauteur de l'élément parent pour les propriétés liées à la boîte: <i>width</i>, <i>height</i>, <i>margin</i>, <i>padding</i>. |
| em | le em est <ul style="list-style-type: none">• relatif à la taille de la fonte héritée pour la propriété <i>font-size</i>,• relatif à la taille de la font courante pour les autres propriétés. |
| rem | le rem (root em) est toujours relatif à la taille de la fonte de l'élément root (la balise HTML). |
| vw | le vw (viewport width) correspond à 1% du de la largeur du viewport, c'est à dire le cadre de la fenêtre du navigateur. |
| vh | le vw (viewport height) correspond à 1% du de la hauteur du viewport, c'est à dire le cadre de la fenêtre du navigateur. |

Les valeurs raccourcies

Il est possible dans certain cas de simplifier l'écriture de certaines valeurs avec l'utilisation de raccourcis. Par exemple au lieu d'utiliser les différentes déclinaisons (top, bottom, left, right) de la propriété margin comme *margin-top: 1em; margin-bottom: 1em; ...* il est possible d'utiliser un des raccourcis suivant:

| | | |
|-----------|-------------------------|---|
| 1 valeur | margin: 1em | s'applique aux 4 côtés |
| 2 valeurs | margin: 0 auto | haut et bas / gauche et droite |
| 3 valeurs | margin: 0 auto 1em | haut / gauche et droite / bas |
| 4 valeurs | margin: 0 10px 20px 1em | haut / droite / bas / gauche (dans le sens des aiguilles d'une montre) |