Cours d'intégration

3W Academy

CSS 2. Les sélecteurs

2. 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.

2.1. 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>
...
```

2.2. 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 pas 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>
```

2.3. 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 pas mettre le # dans l'attribut HTML.

Dans le CSS

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

Dans le document HTML

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

2.4. 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.

2.5. 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 {
    ...
}
```

2.6. 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'ensembles 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.

2.7. 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.

2.8. 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>
```

2.9. 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>
</article>
```

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

2.10. 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
:link	Permet de définir les règles qui s'appliquent à un lien quand celui ci n'est ni survolé, ni cliqué et pas encore visité. *a:link { ... }*
:hover	Appliqué lorsque l'élément est survolé. Pour les liens et les boutons (de préférence) mais fonctionne avec tous les éléments. *a:hover { ... }*
:active	Appliqué lorsque le lien ou le bouton est cliqué. *a:active { ... }*
:visited	Appliqué lorsque le lien a été visité, on a cliqué dessus. *a:visited { ... }*

:first-child	Cible le premier élément de la même fratrie si il correspond à la balise spécifiée (la balise LI dans l'exemple). *li:first-child { ... }*
:last-child	Cible le dernier élément de la même fratrie si il correspond à la balise spécifiée (la balise LI dans l'exemple). *li:last-child { ... }*
:nth-child(expr)	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). *li:nth-child(odd)* cible tous les éléments LI dont l'index est impair. *li:nth-child(even)* cible tous les éléments LI dont l'index est pair. *li:nth-child(4)* cible le quatrième élément LI. *li:nth-child(3n+2)* cible un élément LI tous les 3 éléments en commençant au deuxième index. *li:nth-child(odd) { ... }*
:first-of-type	Cible le premier élément de la même fratrie correspondant à la balise spécifiée. *p:first-of-type { ... }*
:last-of-type	Cible le dernier élément de la même fratrie correspondant à la balise spécifiée. *p:last-of-type { ... }*
:nth-of-type(expr)	Cible les éléments de la même fratrie correspondant à la fois à l'expression donnée ainsi qu'à la balise spécifiée. *p:nth-of-type(odd) { ... }*
:not(expr)	Permet d'inverser une sélection. *article:not(.large)* cible les articles qui n'ont pas la classe "large"

2.11. 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 ::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 ::permet de manipuler la première ligne de l'élément ciblé.

Dans le CSS

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