

# Sélecteurs CSS

## 7.1 Rappel sur les sélecteurs les plus connus

### 7.1.1 Sélecteur

Plusieurs sélecteurs peuvent précéder un bloc de déclaration. Il suffit de les séparer par une virgule. Le bloc de déclarations sera alors appliqué à chacun des sélecteurs, comme l'illustre l'exemple suivant :

```
1 h1, h2, h3{font-weight: bold;}
```

### 7.1.2 Sélecteur de classe

Une classe est un nom que l'on choisit librement (en se limitant aux caractères alphanumériques classiques) et dont on baptise les éléments concernés via l'attribut HTML `class`. Un sélecteur de classe reprend ce nom en le préfixant d'un point (par exemple : `.error`, `.bloc`, ...)

Partie HTML :

```
1 <span class="error"> formulaire incomplet </span>
```

Partie CSS :

```
1 span.error {color: red;}
```

### 7.1.3 Sélecteur d'identifiants

Un identifiant est lui aussi un nom choisi librement. Il se distingue de la classe en ce qu'il ne peut être associé, au plus, qu'à un seul élément du document. Le sélecteur CSS s'y réfère par l'emploi d'un caractère dièse (#) suivi de ce nom

```
1 <ul id="nav"> ... </ul>
```

Partie CSS :

```
1 #nav {margin: 0; padding: 0}
```

L'identifiant désigne un élément unique dans le document (par exemple, `#kiwi`), contrairement aux classes pouvant symboliser plusieurs types d'éléments ou des groupes d'éléments (par exemple, `.fruit`) : l'élément dont l'id est `kiwi` sera unique, mais plusieurs éléments peuvent appartenir à la classe `fruit`.

### 7.1.4 Pseudo élément et pseudo classes

Dans l'optique de mettre en forme des éléments selon un état contextuel (survol, premier enfant, première ligne, ...) qui ne serait pas défini dans l'arbre du document, CSS introduit les concepts de pseudo-élément.

Exemples de pseudo éléments permettant de spécifier l'état d'un lien hypertexte : `:visited`, `:link`, `:over`, `:focus` et `:active`. On peut spécifier également les éléments relativement à un noeud. premier enfant d'un élément : `:firstchild`. la première lettre d'un bloc : `:first-letter`, la première ligne : `:first-line` ou le contenu créé avant : `:before` ou après : `:after` un élément.

## 7.2 sélecteurs d'enfants

### 7.2.1 sélecteurs d'enfant non direct

Le sélecteur d'enfant non direct s'applique à l'enfant ou aux enfants DIRECTS d'un élément désigné. Pour cela il suffit d'utiliser le caractère "espace" dans le sélecteur.

```
1 E F {color : red;}
```

Dans l'exemple précédent, le style rouge s'appliquera à l'élément F s'il est un enfant (petit enfant, arrière petit enfant....) de l'élément E

### 7.2.2 sélecteurs d'enfant direct

Le sélecteur d'enfant s'applique à l'enfant ou aux enfants DIRECTS d'un élément désigné.

```
1 E > F {color : red;}
```

Dans l'exemple précédent, le style rouge s'appliquera à l'élément F s'il est un enfant DIRECT de l'élément E

#### Question 1

Dans l'exemple ci-dessous indiquez la couleur du mot "important" pour chaque sélecteur CSS.

```
1 <p> Un paragraphe contenant
2   <a> un lien
3     <em> important</em>
4   </a>
5 </p>
```

```
1 p > em {background : green;}
2 p em {background : red;}
3 a > em {background : blue;}
4 a em {background : yellow;}
5 em {background : orange;}
```

#### Question 2

Dans le code ci-dessous, indiquez quelles sont les balises sélectionnées par le sélecteur css ?

```
1 <ul id="menu">
2   <li> a </li>
3   <li> b </li>
4   <li> c
5     <ul class="submenu">
6       <li> c1 </li>
7       <li> c2 </li>
8       <li> c3 </li>
9     </ul>
10  </li>
11 </ul>
```

```
1 #menu > li {list-style-type : square;}
```

### 7.2.3 Sélecteur de frère adjacent

Le sélecteur de frère adjacent affecte un élément à condition qu'il soit immédiatement précédé d'un élément spécifié. Sa syntaxe est la suivante :

```
1 E + F {propriete : valeur}
```

Cette règle s'applique à l'élément F s'il est frère de E et s'il succède dans l'ordre de déclaration HTML.

```
1 <h1> Le kiwi </h1>
2 <p> Phrase 1 </p>
3 <p> Phrase 2 </p>
```

```
1 h1 + p {font-style : italic;}
```

Dans l'exemple précédent, seul le premier paragraphe <p> directement adjacent au titre <h1>, sera mis en italique. les paragraphes suivants ne changent pas.

## 7.3 Sélecteurs d'attribut

Le sélecteur d'attribut consiste en une extension du sélecteur d'élément, avec la possibilité de cibler un attribut HTML ou la valeur d'un attribut.

Partie HTML

```
1 [attribut] {propriete valeur;}
```

On peut éventuellement préciser l'élément concerné à l'aide de la syntaxe suivante :

```
1 E[attribut] {propriete valeur;}
```

Il est ainsi possible de cibler, par exemple, uniquement les images disposant d'un attribut alt, à l'aide du sélecteur `img[alt]`.

Le sélecteur d'attribut offre également l'opportunité de préciser la valeur attendue de l'attribut ciblé : le sélecteur `a[hreflang="fr"]` va s'appliquer à tous les liens dont l'attribut `hreflang` contient exactement la valeur `fr`. Pour ne cibler les cellules de tableaux dont l'attribut `colspan` vaut 3, il suffit d'écrire `td[colspan="3"]`

Notez qu'il est parfaitement possible de cumuler deux sélecteurs d'attribut, comme le décrit l'exemple ci-dessous :

```
1 img[class="vignette"][width="250"] {border: 1px solid green};
```

Cette règle s'appliquera aux images dont l'attribut `class` a la valeur "vignette" et l'attribut `width` 250.

## 7.4 Poids des sélecteurs

Les spécifications CSS proposent une classification des sélecteurs selon un barème de poids sous forme de notation à quatre poids :

1. Poids a : règle CSS déclarée à l'aide de l'attribut HTML `style=` au sein de l'élément
2. Poids b : sélecteur d'identifiant
3. Poids c : sélecteur de classe, d'attribut ([]) ou de pseudo-classes ( :hover, :focus, :firstchild)
4. Poids d : sélecteurs d'élément (p, div, a, ...) ou de pseudo-éléments ( :first-letter, :before, :after, ...)
5. Poids nul : parenté (>) ou adjacence (+)

soit le code HTML suivant :

```
1 <div id="warning">
2   <a class="error" href="url"> source erreur</a>
3 </div>
```

```
1 /* 1 selecteur d'element (poids "d") */
2 a {color : green;}
3
4 /* selecteur d'identifiant (poids "b") + une selecteur de poids "d" */
5 #warning a {color: blue;}
6
7 /* 1 selecteur de classe (poids "c") + 1 selecteur de poids "d" */
8 a.error {color : red;}
```

Les 3 règles sélectionnent le même élément, mais la couleur du lien indiquant une erreur sera bleue car, même si la règle 3 la succède, la règle 2 est celle avec le plus de poids.

## 7.5 EXERCICE

### Question 3

De quelle manière optimale puis-je cibler à la fois tous les titres de niveau 1 de ma page, tous les éléments de classe `fruit`, ainsi que l'élément dont l'identifiant est `header` ?

### Question 4

Comment puis-je cibler uniquement le premier niveau d'éléments `<div>` contenus au sein de mon document `<body>` ?

### Question 5

Comment faire pour colorer le texte du paragraphe 3 en rouge ?

```
1 <p> paragraphe 1 </p>
2 <p> paragraphe 2 </p>
3 <p> paragraphe 3 </p>
```

### Question 6

De quelle couleur sera le titre de mon document issu du code HTML et du CSS ci-dessous ?

```
1 <div class="container"><h1 id="first" class="fruit"> Potage de
   kiwi</h1></div>
```

```
1 h1 {color : green;}
2 #first {color : blue;}
3 h1.fruit {color : yellow;}
4 h1.fruit:first-child {color: red;}
```

### Question 7

Construire le code HTML correspondant au CSS suivant :

```
1 p + h1.kiwi + p strong
```

### Question 8

Construisez le code HTML correspondant à la règle CSS suivante :

```
1 p[title*="kiwi"] + p > strong.info
```

### Question 9

Construisez le code HTML correspondant à la règle CSS suivante<sup>a</sup> :

```
1 form#contact[action="destination.php"] > p > label[for="kiwi"] +  
  #kiwi[type="text"]
```

a. regardez sur internet comment fonctionnent les balises form, label et input



Ce document est publié sous Licence Creative Commons « By-NonCommercial-ShareAlike ». Cette licence vous autorise une utilisation libre de ce document pour un usage non commercial et à condition d'en conserver la paternité. Toute version modifiée de ce document doit être placée sous la même licence pour pouvoir être diffusée.

<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>