

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**  
**MINISTERE DE L'ENSEIGNEMENT ET DE LA FORMATION PROFESSIONNELS**  
**INSTITUT NATIONAL SPECIALISE DE FORMATION PROFESSIONNELLE**  
**« El chahid Ahmed Mahdi »**

Direction des études :	<b>Chapitre 2 : Effectuer les mises en forme avec CSS</b>
Formateur : M <sup>me</sup> BENAMMOUR	<b>Leçon 02</b>
Spécialité : Dév web et mobile S1 .	
Matière : les interfaces utilisateurs statiques	

## LES SELECTEURS d'ATTRIBUT ET SELECTEURS AVANCES CSS

### RAPPEL :

---

Une règle CSS a la syntaxe suivante :

```
Selecteur {  
    propriété1 : valeur 1;  
    Propriété2 : valeur2 ;  
    .....  
    .....  
}
```

Déclaration

### 1. DEFINITION :

---

**Les sélecteurs** définissent les éléments sur lesquelles s'applique un ensemble de règles CSS.

### 2. Sélecteurs d'attribut :

---

Les sélecteurs d'attributs sont des sélecteurs simples permettent de cibler des éléments HTML précis en fonction de leurs attributs et de leurs valeurs. Entourez le nom de l'attribut cible avec des « [ ] » pour cibler cet attribut puis vous pouvez préciser la valeur que l'attribut doit avoir pour être ciblé en ajoutant sa valeur à la suite précédée d'un « = »

⇒ **Syntaxe :**

[**attr**] : Permet de cibler un élément qui possède un attribut attr.

**[attr=valeur]** : Permet de cibler un élément qui possède un attribut **attr** dont la valeur est exactement valeur.

**[attr~=valeur]** : Permet de cibler un élément qui possède un attribut **attr** dont la valeur est valeur. Cette forme permet de fournir une liste de valeurs, séparées par des blancs, à tester. Si au moins une de ces valeurs est égale à celle de l'attribut, l'élément sera ciblé.

**[attr|=valeur]** : Permet de cibler un élément qui possède un attribut **attr** dont la valeur est exactement valeur ou dont la valeur commence par valeur un suivi immédiat d'un tiret (U+002D). Cela peut notamment être utilisé pour effectuer des correspondances avec des codes de langues.

**[attr^=valeur]** : Permet de cibler un élément qui possède un attribut **attr** dont la valeur commence par valeur.

**[attr\$=valeur]** : Permet de cibler un élément qui possède un attribut **attr** dont la valeur se termine par valeur.

**[attr\*=valeur]** : Permet de cibler un élément qui possède un attribut **attr** et dont la valeur contient au moins une occurrence de valeur dans la chaîne de caractères.

**[attr opérateur valeur i]** : On peut ajouter un i(ou I) avant le crochet de fin. Dans ce cas, la casse ne sera pas prise en compte (pour les caractères contenus sur l'intervalle ASCII).

**[attr opérateur valeur s]**: Ajouter un s(ou S) avant le crochet fermant permettra d'effectuer une comparaison de valeur sensible à la casse (pour les caractères ASCII).

⇒ **Exemples :**

**Code HTML :**

```
<p id="premier">Mon premier paragraphe</p>
<p class="important">Mon deuxième paragraphe</p>
<p id="dernier" class="important">Mon troisième paragraphe</p>
```

**Code CSS :**

```
[id] {
  // cible le premier et troisième paragraphe
}
[id="dernier"] {
  //cible uniquement le troisième paragraphe
}
```

```
[class] {  
  //cible le deuxième et troisième paragraphe  
}
```

### 3. Sélecteurs CSS avancés avec les pseudo-classes :

---

Les pseudo-classes en CSS permettent de sélectionner des éléments pour une règle CSS en fonction d'un état précis qui peut leur être attribué, mais qui n'est pas inscrit dans le document HTML.

Une pseudo-classe est composée d'un mot-clé précédé par le caractère « : » et qui s'ajoute à la suite d'un autre sélecteur CSS sans espace.

⇒ **Exemples :**

- 1) **:hover** :s'applique lorsque l'utilisateur survole un élément avec sa souris (:hover signifie d'ailleurs "survoler"), et va nous permettre d'y appliquer un style.

⇒ **Syntaxe :**

**sélecteur: pseudo-classe {déclaration}**

⇒ **Exemples :**

**Code HTML :**

```
<body>  
  <div class="mon-element">  
    <h1>Ma page de pseudo-classes</h1>  
    <p>Les pseudo-classes sont très importantes afin de créer des apparences  
dynamiques en fonction des interactions avec les utilisateurs.</p>  
  </div>  
</body>
```

**Code css :**

```
.mon-element:hover {  
  background-color: #BEAEE2;  
  color: white;  
}  
h1:hover {  
  color: #CDF0EA;  
}  
p:hover {  
  color: #F9CEEE;  
}
```

**Résultat :****Ma page de pseudo-classes**

Les pseudo-classes sont très importantes afin de créer des apparences dynamiques en fonction des interactions avec les utilisateurs.

**Ma page de pseudo-classes**

Les pseudo-classes sont très importantes afin de créer des apparences dynamiques en fonction des interactions avec les utilisateurs.

2) **:active** : permet d'appliquer un style particulier au moment du clic.

⇒ **Exemple :**

**Code css :**

```
a:link { color: blue; } /* Liens non visités */
a:visited { color: purple; } /* Liens visités */
a:hover { background: yellow; } /* Liens survolés */
a:active { color: red; } /* Liens actifs */
```

**Code HTML :**

```
<p>Ce paragraphe contient un lien :
  <a href="#">Ce lien devient rouge quand vous cliquez dessus.</a>
  Le paragraphe sera sur un fond gris quand vous cliquerez dessus
  ou sur le lien.
</p>
```

**:not()** : C'est une pseudo-classe de négation qui prend un sélecteur en argument à ajouter entre les « ( ) » de la pseudo-classe et qui permet de ne pas prendre en compte le sélecteur donné en argument dans le sélecteur général de la règle CSS. Le sélecteur donné en argument ne peut pas être suivi d'une pseudo-classe de négation ou cibler de pseudo-élément.

**:first-child et :last-child** : Ces deux pseudo-classe permettent de cibler respectivement le premier et le dernier élément enfant selon le sélecteur auquel il est associé dans un même parent.

**1.4. les pseudo-éléments :**

Les pseudo-éléments sont des éléments du document qui ne sont pas inscrits dans le document même et n'ont donc pas de correspondance en code HTML. Ceux sont des mots-clés précédés de deux « :: » qui s'ajoutent à la suite d'un sélecteur donné.

Les pseudo-éléments sont très utiles pour créer des mises en page plus complexe et des changements d'apparences avancés.

Plusieurs pseudo-éléments existent, mais les plus connus et ceux que vous verrez ou utiliserez le plus souvent sont les deux pseudo-élément « ::before » et « ::after » qui permettent d'ajouter respectivement un contenu avant et après un élément HTML.

⇒ **Syntaxe :**

**sélecteur:: pseudo-element {déclaration}**

⇒ **Exemples :**

**Code HTML :**

```
<!doctype html>
<html>
<head>
<title>PSEUDO ELEMENT</title>
<meta charset="utf-8">
<link rel="stylesheet" type="text/css" href="ELT.css">
</head>
<body>
  <p><span>m</span><sup>me</sup> BENAMMOUR</p>
</body>
<html>
```

**Code CSS :**

```
p {
  color:#808000;
}
p::before {
  content: "je suis ";
  color: red;
}
p::after {
  content: " Formatrice en informatique.";
  color: blue;
}
span{text-transform:uppercase;
}
```

**Résultat :**

je suis M<sup>me</sup> BENAMMOUR Formatrice en informatique.