

420-KB2-LG, Conception d'interfaces Web

Préparé par Saliha Yacoub

Les secteurs CSS

Plan de la séance:

- Définition
- Sélecteurs simples
 - De type
 - De classe
 - D'identifiant
 - Universel
 - D'attribut
- Sélecteurs relationnels
 - Voisin direct
 - De voisin
 - Élément enfant
 - Élément descendant
- Exemple et conclusion

Retour sur la
dernière
séance

Point de vue des
étudiants

Point de vue de
l'enseignante

Les sélecteurs: Définition et Exemples

- Les sélecteurs définissent les éléments sur lesquelles s'applique un ensemble de règles CSS.
- Les sélecteurs de type ou d'élément. (déjà vu): Les sélecteurs de type ciblent des éléments en fonction de leur nom. Ainsi, lorsqu'un sélecteur de type est utilisé seul, il ciblera tous les éléments de ce type contenus dans le document.

Exemple :

```
h4{  
  text-align: center;  
}
```

```
élément {  
  propriété: valeur;  
}
```

Les sélecteurs: Définition et exemples

- Les sélecteurs de classe: (déjà vu) Les sélecteurs de classe permettent de cibler des éléments d'un document en fonction du contenu de l'attribut class de chaque élément.

```
.nom_de_classe {  
  propriété: valeur;  
}
```

```
/* tous les éléments de classe "biz" */  
  
.biz {  
  background-color: lavender;  
}  
  
/* tous les titres de niveau 5 de classe "biz" */  
  
h5.biz {  
  text-decoration: underline;  
}
```

Les sélecteurs: Définition et exemple

- Les sélecteurs d'identifiant: (déjà vu): Un sélecteur d'identifiant (ID selector) permet, pour un document HTML, de cibler un élément grâce à la valeur de son attribut id. Il faut que la valeur soit exactement la même que celle du sélecteur pour que l'élément soit effectivement ciblé.

Syntaxe:

```
#identifiant {  
    propriété: valeur;  
}
```

Les sélecteurs: Définition et exemple

- Sélecteur universel: *: L'astérisque (*) est le sélecteur universel en CSS. Il correspond à un élément de n'importe quel type. Avec le code CSS suivant, tous les éléments auront une marge de 2 pixels :

Syntaxe:

```
* {  
  propriété: valeur;  
}
```

```
/* tous les éléments */  
  
* {  
  margin: 2px;  
}
```

Les sélecteurs: Définition et exemple

- **Sélecteur d'attribut:** Les sélecteurs d'attribut permettent de cibler un élément selon la présence d'un attribut ou selon la valeur donnée d'un attribut.

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".

Avec le code CSS suivant, tous les éléments ayant un attribut "alt" auront une bordure pointillée et tous ceux ayant un attribut "alt" dont la valeur est "exemple" auront une bordure pleine :

```
[alt] {  
  border: dotted black 1px;  
}  
  
[alt=exemple] {  
  border: solid black 1px;  
}
```

Les sélecteurs: Définition et exemple

Sélecteurs relationnels: Les sélecteurs relationnels, aussi appelés combinateurs, mettent en jeu un élément par rapport à un autre.

Sélecteurs de voisin direct:

Ce combinateur permet de sélectionner un élément uniquement si celui-ci suit (immédiatement) un élément donné et que les deux éléments sont les fils d'un même élément parent.

```
premier_element + element_cible {  
  propriété: valeur;  
}
```

Les sélecteurs: Définition et exemple

Voici le code html et le code css . em et strong sont fils de l'élément p.

```
<p>Lorem ipsum dolor sit , amet <em>consectetur </em>  
adipisicing elit. <strong>  
    Quisquam, accusamus.</strong>  
    <em>Lorem ipsum dolor sit amet.</em>  
    <em> un autre em lui est vert</em>  
</p>
```

```
em{  
    color:green  
}  
  
strong + em {  
    color: red;  
}
```

Lorem ipsum dolor sit , amet *consectetur* adipisicing elit. **Quisquam, accusamus.** *Lorem ipsum dolor sit amet. un autre em lui est vert*

Les sélecteurs: Définition et exemple

Sélecteurs de voisin

Ce combinateur permet de sélectionner un élément si celui-ci correspond au second sélecteur et est précédé (mais pas forcément voisin immédiat) d'un autre élément correspondant au premier sélecteur qui partage le même parent.

```
premier_élément ~ second_élément {  
    propriété: valeur;  
}
```

Les sélecteurs: Définition et exemple

Voici le code html et le code css . em et strong sont fils de l'élément p.

```
<p>Lorem ipsum dolor sit , amet <em>consectetur </em>
adipisicing elit. <strong>
    Quisquam, accusamus.</strong>
    <em>Lorem ipsum dolor sit amet.</em>
    <em> un autre em lui est rouge</em>
</p>
```

```
em{
    color:green
}

strong ~ em {
    color: red;
}
```

Lorem ipsum dolor sit , amet *consectetur* adipisicing elit. **Quisquam, accusamus.** *Lorem ipsum dolor sit amet. un autre em lui est rouge*

Les sélecteurs: Définition et exemple

Sélecteurs d'éléments enfants

Ce combinateur cible les éléments correspondant au second sélecteur et qui sont des enfants directs des éléments ciblés par le premier sélecteur.

```
selecteur1 > selecteur2 {  
  propriété: valeur;  
}
```

Les sélecteurs: Définition et exemple

Tous les p qui sont dans des div seront rouge et centrés selon le css. Les p en dehors des div n'ont pas de style.

```
<p>
  Ce texte n'est ni rouge ni centré.</p>
<div>
  <p>Ce texte est rouge et centré.</p>
</div>
<p> Un autre texte qui n'a pas de style.</p>
<div>
  <p>Cet autre texte est rouge et centré.</p>
</div>
```

```
div > p {
  color:red;
  text-align: center;
}
```

Les sélecteurs: Définition et exemple

- **Sélecteurs d'éléments descendants**

Ce combinateur cible les éléments correspondant au second sélecteur et qui sont des descendants (enfants directs ou indirects) des éléments ciblés par le premier sélecteur.

```
sélecteur1 sélecteur2 {  
  propriété: valeur;  
}
```

Les sélecteurs: Définition et exemple

Le style sera appliqué au div qui sont dans le div dont le id est « conteneur »

```
<p> Ce texte n'a aucun style.</p>
<div id="conteneur">
  <div>
    <p> 1 - ici le style sera appliqué</p>
  </div>
  <div>
    <p> 2- ici aussi le style sera appliqué</p>
  </div>
</div>
<div>
  <p>Lorem ipsum dolor sit amet.</p>
</div>
```

```
#conteneur {
  color: blue;
}
#conteneur div {
  background: grey;
  padding: 20px;
}
```

Les sélecteurs: Tableau récapitulatif

Sélecteur...	Syntaxe	Cible
Sélecteurs simples		
de type	A {...}	Tout élément A .
de classe	.classe {...}	Tout élément ayant l'attribut class="classe" .
d'identifiant	#identifiant {...}	L'élément (unique) ayant l'attribut id="identifiant" .
universel	* {...}	Tout élément.
d'attribut	[attr] {...}	Tout élément ayant l'attribut attr .
d'attribut	[attr=valeur] {...}	Tout élément ayant l'attribut attr="valeur" .

Les sélecteurs: Tableau récapitulatif

Sélecteur...	Syntaxe	Cible
Sélecteurs relationnels (combinateurs)		
de voisin direct	$A + B \{...\}$	Tout élément B qui suit immédiatement un élément A.
de voisins	$A \sim B \{...\}$	Tout élément B qui suit, dans le même parent, un élément A.
d'éléments enfants	$A > B \{...\}$	Tout élément B qui a un parent direct A.
d'éléments descendants	$A B \{...\}$	Tout élément B qui a un ancêtre A.

Exemple et conclusion

```
<p>Voici la liste des <em>mammifères</em> et des<em>reptiles</em> </p>
  <ul>
    <li>Mammifères
      <ul id="liste_mammif">
        <li>Lapin</li>
        <li>Castor</li>
        <li>Porc-épic <em>à longues dents</em></li>
      </ul>
    </li>
    <li>Reptiles
      <ul id="liste_rept">
        <li>Serpent</li>
        <li>Lézard</li>
      </ul>
    </li>
  </ul>
  <p>Les animaux sont des <em>êtres vivants</em> <mark>fascinants</mark> et
  <em>beaux</em></p>
```

```
ul em {
  color: red;
}
p > em {
  color: blue;
}
em + em {
  color: fuchsia;
}
```

On s'intéresse
particulièrement à
l'élément em

Exemple et conclusion

Ici, nous n'avons pas encore défini le css des listes: `liste_mammif` et `liste_rept`

Voici la liste des *mammifères* et des *reptiles*

- Mammifères
 - Lapin
 - Castor
 - Porc-épic *à longues dents*
- Reptiles
 - Serpent
 - Lézard

Les animaux sont des *êtres vivants* **fascinants** et *beaux*

Exemple et conclusion

- La première règle ne s'applique qu'à la chaîne "à longues dents", car c'est le seul élément em à l'intérieur d'un élément ul, même s'il n'est pas un enfant immédiat.
- La deuxième règle devrait s'appliquer aux chaînes "mammifères" et "reptiles" de la première ligne, de même qu'aux chaînes "êtres vivants" et "beaux" de la dernière ligne.
- C'est ce qui arrive, sauf pour la chaîne "reptiles" qui est soumise à la troisième règle étant donné qu'il s'agit d'un élément em qui suit immédiatement un autre élément em.
- Notons que cette règle ne s'applique pas à la dernière ligne étant donné que les éléments em sont séparés par un élément mark.
- Pour cela il faudrait la règle suivante : em - em {color: fuchsia;}.

Exemple et conclusion

```
<p>Voici la liste des <em>mammifères</em> et des<em>reptiles</em> </p>
<ul>
  <li>Mammifères
  <ul id="liste_mammif">
    <li>Lapin</li>
    <li>Castor</li>
    <li>Porc-épic <em>à longues dents</em></li>
  </ul>
</li>
<li>Reptiles
  <ul id="liste_rept">
    <li>Serpent</li>
    <li>Lézard</li>
  </ul>
</li>
</ul>
<p>Les animaux sont des <em>êtres vivants</em>
<mark>fascinants</mark> et <em>beaux</em></p>
```

```
ul em {
  color: red;
}
p > em {
  color: blue;
}
em + em {
  color: fuchsia;
}
#liste_mammif li {
/* ou #liste_mammif > li */
  font-family: sans-serif;
}
#liste_rept li {
/* ou #liste_rept > li */
  font-family: monospace;
}
```

Exemple et conclusion

Les deux listes ont maintenant des polices de caractères différentes :

Voici la liste des *mammifères* et des *reptiles*

- Mammifères
 - Lapin
 - Castor
 - Porc-épic *à longues dents*
- Reptiles
 - Serpent
 - Lézard

Les animaux sont des *êtres vivants* **fascinants** et *beaux*

Questions



CONCLUSION



QUESTIONS ??

Sources

- <https://www.w3schools.com/html/>
- <https://prog101.com/cours/kb2/>