

Développement front

MMI 2 – CM#4 S3



Danielo **JEAN-LOUIS**
Michele **LINARDI**

CSS — Rappels

- Langage de description
 - Complémentaire du HTML
- Permet de définir la présentation d'une page web
- Extension de fichier : .css
- Fonctionne avec un système de sélecteurs

CSS — Sélecteurs (vus jusqu'à présent)

- Sélecteur d'id : #mon-id
- Sélecteur de classe : .ma-classe
- Sélecteur d'attribut : [mon-attribut]
- Sélecteur universel : *
- Sélecteur de pseudo-classe : :pseudo-classe
- Sélecteur de pseudo-element : ::pseudo-element

**CSS 3 est venu ajouter
(progressivement) d'autres
sélecteurs (et pseudo-classes) très
utiles pour le développement**

Des sélecteurs dits “avancés”

Pourquoi les sélecteurs CSS avancés ?

- Réduire l'interdépendance entre le développement front et back pour appliquer certaines déclarations CSS
 - Ex : Appliquer une classe une fois sur deux
- Pallier aux limites de certains sélecteurs CSS
 - Ex : sélecteur de descendants

Pourquoi les sélecteurs CSS avancés ?

- Améliorer le référencement (SEO) du site web en réordonnant le site pour que le HTML soit “SEO compatible” mais la mise en page du site conforme au design

Sélecteur d'enfants directs

- Permet de cibler l'enfant direct d'une balise

A code editor window with a title bar containing three colored circles (red, yellow, green) and a purple tab labeled 'CSS'. The code inside is a CSS selector and rule:

```
div > span {  
    background-color: blue;  
}
```

 The greater-than sign (>) in the selector is highlighted with a red square box.

```
div > span {  
    background-color: blue;  
}
```

Le caractère “>” permet de cibler tout contenu directement dans une balise <div>

Note : il est possible de mettre plusieurs “>” dans le sélecteur

Source(s) :

- https://developer.mozilla.org/fr/docs/Web/CSS/Next-sibling_combinator

Sélecteur de voisin direct

- Cible toute balise située juste à côté d'une autre



```
.paragraphe+ .liste {  
    margin-block: 1.5rem;  
}
```




```
<div class="paragraphe"></div>  
<ul class="liste"></ul>
```

Le caractère “+” permet de cibler tout `.liste` précédé par `.paragraphe`

Note : il est possible de mettre plusieurs “+” dans le sélecteur

Sélecteur de voisin

- Alternative au sélecteur de voisin direct
- Cible tout élément suivant un autre (immédiatement ou non)



```
.paragraphe~.liste {  
    margin-block: 1.5rem;  
}
```



```
<div class="paragraphe"></div>  
<figure>❗ [ ... ] ❗</figure>  
<ul class="liste"></ul>
```

Le caractère “~” permet de cibler tout .liste précédé par .paragraphe~, et ce, même s’il y a un élément (ou plus) entre les deux

Note : il est possible de mettre plusieurs “~” dans le sélecteur

Pseudo-classe - :is(<selecteurs>)

- Alternative “saine” du sélecteur de groupe
 - Ex : .ma-classe1, ma-classe2
- Permet de raccourcir des sélecteurs

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/:is>

Pseudo-classe - :is(<selecteurs>)



```
:valid,  
:unsupported {  
    /* ... */  
}
```

La pseudo-classe “unsupported” n’existant pas, l’ensemble du sélecteur ne sera pas appliqué



```
:is(:valid, :unsupported) {  
    /* ... */  
}
```

Avec la pseudo-classe is(), même si “:unsupported” n’existe pas, les déclarations CSS s’appliqueront pour “:valid”

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/:is>

Pseudo-classe - :is(<selecteurs>)



```
h1.newclass, p.newclass, li.newclass {  
    background-color: yellow;  
}
```



```
.newclass:is(h1, p, li) {  
    background-color: yellow;  
}
```

Les deux codes font la même chose, mais avec la pseudo-classe is(),
le code est plus court

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/:is>

Pseudo-classe - selecteur:has(<condition>)

- Cible un élément en fonction de la condition passée en paramètre
- Sorte de sélecteur de parent
- Peut être combiné avec la pseudo-classe :is()
 - Voir documentation mdn

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/:has>

Pseudo-classe - selecteur:has(<condition>)



On cible tout élément avec la classe .article qui possède une balise img comme enfant

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/:has>

Pseudo-classe - selecteur:not(<condition>)

- Sélectionne tout élément qui ne correspond pas aux conditions passées en paramètres
- Inverse de la pseudo-classe has()

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/:not>

Pseudo-classe - selecteur:not(<condition>)



```
.article :not(p):not(.important) {  
  background-color: darkred;  
}
```

Cible tout élément n'ayant pas la classe "important" ou n'étant pas la balise <p> contenu dans la classe .article



```
.article:not(.important, .archive) {  
  color: rebeccapurple;  
  font-style: italic;  
}
```

Cible tout élément avec la classe .article et ne possédant pas la classe .important ni .archive

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/:not>

Pseudo-classe - :nth-child()

- Cible des éléments en fonction de leur indice dans leur parent
- Accepte en condition un mot-clé ou une fonction sous la forme : $An+B$
 - A : Incrément entier
 - B : Décalage entier
 - n : Position de l'élément dans son parent

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/:nth-child>

Pseudo-classe - :nth-child()



```
li:nth-child(even) {  
  background-color: pink;  
}
```

On cible un sur deux en commençant par deuxième

Note : Le mot-clé “odd” fait l’inverse



```
li:nth-child(-n + 2) {  
  color: darkblue;  
}
```

On cible les deux premiers

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/:nth-child>

Pseudo-classe - :nth-child()

- Possibilité de faire une plage de sélection



```
li:nth-child(n+4):nth-child(-n+10) {  
    /* [ ... ] */  
}
```

On cible le quatrième jusqu'au dixième

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/:nth-child>

Pseudo-classe - :nth-child(of <selector>)

- Alternative à la pseudo-classe:nth-child()
- Sélectionne les n éléments trouvés (quelque soit leur position dans leur parent)

Source(s) :

- https://developer.mozilla.org/fr/docs/Web/CSS/:nth-child#la_syntaxe_of_selector

Pseudo-classe - :nth-child(of <selector>)



```
:nth-child(-n + 3 of li.important) {  
  color: red;  
}
```

On cible les trois premiers avec la classe .important trouvés



```
li.important:nth-child(-n + 3) {  
  color: red;  
}
```

On cible les trois premiers avec la classe .important si et seulement s'ils sont les premiers de leur parent

Source(s) :

- https://developer.mozilla.org/fr/docs/Web/CSS/:nth-child#la_syntaxe_of_selector

Pseudo-classe - :nth-child() - Raccourcis

- Pseudo-classes permettant de cibler directement un index précis
 - Premier élément : :first-child
 - Dernier élément : :last-child
- Calculateur de nth-child()
 - <https://css-tricks.com/examples/nth-child-tester/>

Source(s) :

- https://developer.mozilla.org/fr/docs/Web/CSS/:nth-child#la_syntaxe_of_selector

Pseudo-classe - :nth-last-child()

- Fonctionne comme :nth-child() mais on compte à partir de la fin
- Fonction de forme : $An+B-1$



On cible les trois derniers

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/:nth-last-child>

Pratiquons ! - Découvrons les sélecteurs CSS avancés

Pré-requis :

- Avoir la ressource [ressources/selecteurs-css-avances](#)

A télécharger ici :

Sélecteurs CSS avancés

- Fonctionnent avec tailwindcss avec des modifieurs dédiés :
 - :last, :even, :has()...
- Ou classes :
 - peer, group-*

Source(s) :

- <https://tailwindcss.com/docs/hover-focus-and-other-states#pseudo-classes>
- <https://tailwindcss.com/docs/hover-focus-and-other-states#styling-based-on-the-descendants-of-a-peer>

Questions ?

