

HTML & CSS

Cours du 12 novembre 2018

PLAN DU COURS

- Introduction
- Le HTML
- **Le CSS**
- Bootstrap
- Les bonnes pratiques
- Mise en production
- Initiation à la modularisation ?

OBJECTIFS DU JOUR

- L'intégration et la syntaxe des CSS
- Les sélecteurs
- Les pseudo-classes et les pseudo-éléments
- La notion de cascade et d'héritage
- Les unités de mesure
- La notion des couleurs
- Le concept du modèle de boîte

L'INTÉGRATION DU CSS

Dans le header via la balise
<link>

Dans le <body> via une balise
<style>

Sur un élément via l'attribut
style

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Titre de la page</title>
    <link rel="stylesheet" href="style.css">
    <script src="script.js"></script>
  </head>
  <body>
    <style>
      header{
        color : blue;
      }
    </style>
    <header style="color:blue;">...</header>
  </body>
</html>
```

LES SÉLECTEURS

X

Permet de cibler un élément par type.

/!\ Le sélecteur de type étant très large, veuillez à ne pas être trop spécifique dans les règles qui lui seront associées.

```
//html
```

```
<p>
```

```
...
```

```
</p>
```

```
//css
```

```
p{
```

```
    color: grey;
```

```
}
```

```
//DON'T
```

```
p{
```

```
    color: grey;
```

```
    margin-left: 20px;
```

```
    font-weight: bold;
```

```
    font-style: italic;
```

```
...
```

```
}
```

LES SÉLECTEURS

#value

Permet de cibler un élément par son ID.

⚠ Un ID doit être unique dans la page !

```
//html
```

```
<div id="main-wrapper">  
    ...  
</div>
```

```
//css
```

```
#main-wrapper{  
    color: blue;  
}
```

LES SÉLECTEURS

.value

Permet de cibler un (des) élément(s) par la Class.

⚠ Une class peut être réutilisée dans la page !

```
//html
```

```
<h3 class="section-title">  
    ...  
</h3>
```

```
//css
```

```
h3{  
    font : normal 18px/125% Arial;  
}  
  
.section-title{  
    margin-bottom: 20px;  
    font-weight: bold;  
    color: grey;  
}    ...
```

LES SÉLECTEURS

XY

Permet de cibler un élément descendant.

⚠ Vous pouvez l'associer avec n'importe quel autre sélecteur.

```
//html
```

```
<div>
  <p class="main-title">
    Ciblé
  </p>
</div>

<p>non ciblé</p>
```

```
//css
```

```
div p{
  font : normal 18px/125% Arial;
}

div .main-title{
  font : bold 24px/125% Arial;
}
```


LES SÉLECTEURS

X[attribut]

Permet de cibler un élément par son attribut.

```
//html
```

```
<a href="https://google.fr">lien</a>
```

```
<div data-state="closed">...</div>
```

```
//css
```

```
a[href="https://google.fr"]{  
    color : green;  
}
```

```
div[data-state="closed"] {  
    color : red;  
}
```

LES SÉLECTEURS

X:after

X:before

Ce sont des pseudo-éléments. Ils permettent de générer du contenu autour de l'élément X.

⚠ Ne marche pas sur les balises auto-fermantes.

⚠ Le « content » est obligatoire, sans cela le pseudo-élément ne s'affichera pas.

```
//html
```

```
<q>...<q>
```

```
//css
```

```
q:before{  
    content : '“”';  
}
```

```
q:before{  
    content : '“”';  
}
```

LES SÉLECTEURS

X:hover

Permet d'appliquer un style au survol de l'élément X.

```
//html
```

```
<a href="/projets/mon-projet">  
    
  <span class="title">Mon projet</span>  
</a>
```

```
//css
```

```
a:hover{  
  opacity : 0.8;  
}  
  
a:hover img{  
  transform : scale(1.05);  
}
```

LES SÉLECTEURS

X:not(selecteur)

Permet de sélectionner un élément, ne répondant pas aux critères dans le :not().

```
//html
```

```
<p>ciblé</p>
```

```
<p class="introduction">non ciblé</p>
```

```
//css
```

```
p{
```

```
    color : grey;
```

```
}
```

```
p:not(.introduction){
```

```
    font-weight : bold;
```

```
}
```

LES SÉLECTEURS

X:nth-child(n)

Permet de sélectionner le nième élément.

/!\ Ne fonctionne qu'avec les sélecteurs de type.

<https://css-tricks.com/examples/nth-child-tester/>

```
//html
```

```
<ul>
  <li>● ●</li>
  <li>● ● ●</li>
  <li>● ●</li>
  <li>● ●</li>
  <li>● ●</li>
  <li>● ●</li>
</ul>
```

```
//css
```

```
li:nth-child(1n){}
li:nth-child(odd){} //paire
li:nth-child(even){} //impaire
li:nth-child(2){}
```

LES NOTIONS DE CASCADE

C'est la priorité d'une règle par rapport à une autre.

La pondération est calculée selon les 3 points suivants :

- Importance
- Spécificité
- Ordre de déclaration

LES NOTIONS DE CASCADE - IMPORTANCE

Pour rendre une déclaration prioritaire par rapport à une autre, il est possible d'utiliser la syntaxe « !important ».

Par exemple :

```
//css

p{
    color : grey!important;
}
#red{
    color : red;
}
```

LES NOTIONS DE CASCADE - SPÉCIFICITÉ

C'est la mesure de la spécificité d'un sélecteur.
Pourquoi un sélecteur est prioritaire à un autre ?

Sélecteur	Milliers	Centaines	Dizaines	Unités	Spécificité
H1	0	0	0	1	1
#main-header	0	1	0	0	100
.introduction	0	0	1	0	10
p.introduction	0	0	1	1	11
Css inline	1	0	0	0	1000

LES NOTIONS DE CASCADE - SPÉCIFICITÉ

Sélecteur	Milliers	Centaines	Dizaines	Unités
X	Mettre un 1 si le style est inline	Mettre un 1 pour chaque ID dans le sélecteur	Mettre un 1 pour chaque class dans le sélecteur	Mettre un 1 pour chaque élément ou pseudo-élément dans le sélecteur

/! Les sélecteurs *, +, >, ~, :not n'ont pas d'effet sur la spécificité

LES NOTIONS DE CASCADE - SPÉCIFICITÉ

Sélecteur	Milliers	Centaines	Dizaines	Unités	Spécificité
p > a					
#main-header p					
div p a span					
.introduction #price					
Css inline					

LES NOTIONS DE CASCADE — ORDRE DE DÉCLARATION

```
//css
```

```
p{  
    color : blue;  
}
```

```
// Cette règle l'emporte sur celle du dessus  
p{  
    color : purple;  
}
```

L'HÉRITAGE

Certaines propriétés CSS peuvent être héritées d'un élément parent à un élément enfant.

Par exemple les propriétés comme font ou color sont héritées naturellement du parent vers l'enfant.

Par exemple les propriétés margin, padding, background ne sont pas héritées naturellement. Cependant il est possible de forcer cet héritage avec la valeur de propriété : inherit.

LES UNITÉS DE MESURE

```
//css
```

```
100px; // 100 pixels
```

```
100% // 100% de la taille de son parent
```

```
100vw // 100% de la largeur la fenêtre, qu'importe la largeur du parent
```

```
100vh // Même règle que le VW mais avec la hauteur
```

```
1em // Identique à la taille de police de l'élément actuel
```

```
1rem // identique à la taille de police défini par défaut (16px)
```

```
1.2em // 1.2X Identique à la taille de police de l'élément actuel
```

```
1.5rem // 1.5X la taille de police défini par défaut
```

Pour l'unité em, attention à l'héritage.

LES COULEURS

```
//css
```

`#ffffff` -> Code hexadécimal

`rgb([r], [g], [b])` -> Définie par trois nombre de 0 à 255;

`rgba([r], [g], [b], [a])` -> Identique à RGB mais avec un paramètre alpha;

Yellow, green, blue, aqua... Il existe également des couleurs normalisées

LE CONCEPT DU MODÈLE DE BOÎTE

Il existe 3 types pour les éléments HTML :

- Bloc
- En ligne
- Invisible

Certaines propriétés ne sont conçues que pour s'appliquer aux éléments blocs.

LE CONCEPT DU MODÈLE DE BOÎTE - BLOC

Ils prennent par défaut toute la largeur disponible.

Ils se positionnent par défaut l'un sous l'autre.



LE CONCEPT DU MODÈLE DE BOÎTE - EN LIGNE

Ils n'occupent que la largeur indispensable à l'affichage du contenu.

Ils sont positionnés côte à côte.



LE CONCEPT DU MODÈLE DE BOÎTE - INVISIBLE

Ils ne servent qu'à apporter des informations invisibles pour l'internaute.



LE CONCEPT DU MODÈLE DE BOÎTE - RÈGLE

Il est possible d'imbriquer des éléments en ligne dans un élément bloc, mais l'inverse n'est pas possible.

Il est possible d'imbriquer des éléments en ligne dans d'autres éléments en ligne et des éléments bloc dans d'autres éléments bloc.

Il est possible de changer le comportement d'un élément avec la propriété « display ».



PROJET /

Intégrer une maquette HTML

LEARN MORE

Timeshift 165

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce mattis, felis id ullamcorper consectetur, nunc mauris tincidunt mauris, egestas congue purus nunc non nibh. Etiam tincidunt, massa eget ullamcorper volutpat, neque arcu venenatis ligula, a consequat purus est vitae mi.

