

Introduction au langage CSS

Informatique et Statistique 2A 3^{ème} année

Olivier Caron¹

¹École d'ingénieurs Polytech Lille
Université de Lille

16 novembre 2023



Objectifs de ce cours

- Ce cours est une **introduction** au langage CSS

Objectifs de ce cours

- Ce cours est une **introduction** au langage CSS
- Inutile d'en savoir plus pour un développeur Web - Bases de données

Objectifs de ce cours

- Ce cours est une **introduction** au langage CSS
- Inutile d'en savoir plus pour un développeur Web - Bases de données
- Recours aux **frameworks** CSS

La technologie CSS

- Acronyme pour "Cascading Style Sheet"

La technologie CSS

- Acronyme pour "Cascading Style Sheet"
- Découpage propre charte graphique / les données à afficher

La technologie CSS

- Acronyme pour "Cascading Style Sheet"
- Découpage propre charte graphique / les données à afficher
- Styles visuels **réutilisables**

La technologie CSS

- Acronyme pour "Cascading Style Sheet"
- Découpage propre charte graphique / les données à afficher
- Styles visuels **réutilisables**
- Qualités évolutives (un seul fichier à évoluer)

Syntaxe Générale CSS

- Syntaxe ultra simple :

```
sélecteur {  
  nomPropriété: valeur ;  
  nomPropriété: valeur ;  
  ...  
}  
sélecteur ...
```

Syntaxe Générale CSS

- Syntaxe ultra simple :

```
sélecteur {  
  nomPropriété: valeur ;  
  nomPropriété: valeur ;  
  ...  
}  
sélecteur ...
```

- La complexité réside dans la richesse des propriétés et des effets de bord des propriétés entre elles (placement des objets)

Petite illustration de règles CSS

- `h1 { color : red }` → les balises `<h1>` sont en rouge

Petite illustration de règles CSS

- `h1 { color : red }` → les balises `<h1>` sont en rouge
- `div h1 { color : red ; }` → les balises `<h1>` contenues dans une balise `<div>` sont en rouge

Petite illustration de règles CSS

- `h1 { color : red }` → les balises `<h1>` sont en rouge
- `div h1 { color : red ; }` → les balises `<h1>` contenues dans une balise `<div>` sont en rouge
- `.attention { border: solid blue 1px; background-color: cyan }`
→ style applicable à toute balise html (exemple `<div class="attention"> ...`)

Petite illustration de règles CSS

- `h1 { color : red }` → les balises `<h1>` sont en rouge
- `div h1 { color : red ; }` → les balises `<h1>` contenues dans une balise `<div>` sont en rouge
- `.attention { border: solid blue 1px; background-color: cyan}`
→ style applicable à toute balise html (exemple `<div class="attention"> ...`)
- `#id121 { color:blue }` → style applicable à l'**unique** élément `<XXX id="id121"> ...`

Les sélecteurs

- Les sélecteurs de base :
Nom de la balise, Nom de l'id ('#'), Nom de la classe ('.').

Les sélecteurs

- Les sélecteurs de base :
Nom de la balise, Nom de l'id ('#'), Nom de la classe ('.').
- Les sélecteurs avancés :
 - * : désigne toutes les balises
 - sél1 sél2 : les sélecteurs sél2 situés à l'intérieur de sél1
 - A[B] : une balise A qui possède un attribut B
 - A[B="value"] : une balise A qui possède un attribut B de valeur value

Les sélecteurs

- Les sélecteurs de base :
Nom de la balise, Nom de l'id ('#'), Nom de la classe ('.').
- Les sélecteurs avancés :
 - * : désigne toutes les balises
 - sél1 sél2 : les sélecteurs sél2 situés à l'intérieur de sél1
 - A[B] : une balise A qui possède un attribut B
 - A[B="value"] : une balise A qui possède un attribut B de valeur value
- Et bien plus encore

Comment appliquer un style ?

- Au niveau de la balise, attribut `style` :

```
<p style="color:blue;">
```

Comment appliquer un style ?

- Au niveau de la balise, attribut `style` :

```
<p style="color:blue;">
```

- Au niveau du fichier html :

```
<head> ...  
  <style>  
    p {  
      color:blue;  
    }  
  </style>  
  ...
```

Comment appliquer un style ?

- Au niveau de la balise, attribut `style` :

```
<p style="color:blue;">
```

- Au niveau du fichier html :

```
<head> ...  
  <style>  
    p {  
      color:blue;  
    }  
  </style>  
  ...
```

- Dans une feuille de style (recommandé) :

```
<head> ...  
  <link rel="stylesheet" href="./css/style.css" />
```

Qui s'occupe des feuilles de style ?

- Absolument pas le développeur !

Qui s'occupe des feuilles de style ?

- Absolument pas le développeur !
- Réservé au concepteur graphique multimédia, pour s'en convaincre, voici une démo :

<http://www.csszengarden.com/tr/francais/>

Qui s'occupe des feuilles de style ?

- Absolument pas le développeur !
- Réservé au concepteur graphique multimédia, pour s'en convaincre, voici une démo :

<http://www.csszengarden.com/tr/francais/>

- Complexe, mais des frameworks CSS, ex :
framework bootstrap (twitter) <http://getbootstrap.com/>

Le framework Twitter Bootstrap (1/3)

- Une charte graphique épurée

Le framework Twitter Bootstrap (1/3)

- Une charte graphique épurée
- Beaucoup de règles pour de multiples composants graphiques

Le framework Twitter Bootstrap (1/3)

- Une charte graphique épurée
- Beaucoup de règles pour de multiples composants graphiques
- S'appuie sur la notion de classe pour utiliser la charte

```
<button>un bouton</button> <!-- pas de style appliqué -->  
<button class="btn btn-primary">un joli bouton</button>
```

Le framework Twitter Bootstrap (1/3)

- Une charte graphique épurée
- Beaucoup de règles pour de multiples composants graphiques
- S'appuie sur la notion de classe pour utiliser la charte

```
<button>un bouton</button> <!-- pas de style appliqué -->  
<button class="btn btn-primary">un joli bouton</button>
```

- Du code javascript permet d'associer du comportement aux composants

Le framework Twitter Bootstrap (1/3)

- Une charte graphique épurée
- Beaucoup de règles pour de multiples composants graphiques
- S'appuie sur la notion de classe pour utiliser la charte

```
<button>un bouton</button> <!-- pas de style appliqué -->  
<button class="btn btn-primary">un joli bouton</button>
```

- Du code javascript permet d'associer du comportement aux composants
- Le placement des composants est simplifié grâce au concept de grille

Le framework Twitter Bootstrap (1/3)

- Une charte graphique épurée
- Beaucoup de règles pour de multiples composants graphiques
- S'appuie sur la notion de classe pour utiliser la charte

```
<button>un bouton</button> <!-- pas de style appliqué -->  
<button class="btn btn-primary">un joli bouton</button>
```

- Du code javascript permet d'associer du comportement aux composants
- Le placement des composants est simplifié grâce au concept de grille
- Règles dédiées aux mobiles/smartphones (responsive design).

Le framework Twitter Bootstrap (2/3)

- 2 modes d'utilisation :

Le framework Twitter Bootstrap (2/3)

- 2 modes d'utilisation :
 - 1 Téléchargement d'une version bootstrap sur <http://www.getbootstrap.com>, puis installation dans un projet PHP

Le framework Twitter Bootstrap (2/3)

- 2 modes d'utilisation :
 - ① Téléchargement d'une version bootstrap sur <http://www.getbootstrap.com>, puis installation dans un projet PHP
 - ② Référencement à distance d'une version bootstrap (coût écologique)

Le framework Twitter Bootstrap (3/3)

- Exemple utilisation Bootstrap (lignes 8-11) :

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>mon projet web</title>
6
7 <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
8 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
9     rel="stylesheet">
10 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js">
11 </script>
12 </head>
13 <body> ... </body>
14 </html>
```

Le framework Twitter Bootstrap (3/3)

- Exemple utilisation Bootstrap (lignes 8-11) :

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>mon projet web</title>
6
7 <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
8 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
9       rel="stylesheet">
10 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js">
11 </script>
12 </head>
13 <body> ... </body>
14 </html>
```

- Ligne 7 : permet le mode pour les mobiles (responsive design)

Les Grilles CSS

- Le placement des objets dans une page se fait dans une grille virtuelle de **12** colonnes

Les Grilles CSS

- Le placement des objets dans une page se fait dans une grille virtuelle de **12** colonnes
- La grille est définie à l'aide de la classe `container`

Les Grilles CSS

- Le placement des objets dans une page se fait dans une grille virtuelle de **12** colonnes
- La grille est définie à l'aide de la classe `container`
- Une ligne de la grille est définie par la classe `row`

Les Grilles CSS

- Le placement des objets dans une page se fait dans une grille virtuelle de **12** colonnes
- La grille est définie à l'aide de la classe `container`
- Une ligne de la grille est définie par la classe `row`
- Chaque colonne est spécifiée à l'aide des classes `col-xx-nb` ou `col-nb` `xx` précise l'équipement, `nb` le nombre de colonnes.

```
<div class="container">
  <div class="row">
    <div class="col-6">texte sur 6 colonnes</div>
    <div class="col-3">texte sur 3 colonnes</div>
  </div>
  ...
</div>
```

Les tailles des grilles CSS

- Plusieurs formats selon l'équipement :

Equipement	Largeur	classe CSS	Taille max col
Extra small-devices	<768px	<i>col-nb</i>	auto
Small devices	>=768px	<i>col-sm-nb</i>	60px
Medium devices	>=992px	<i>col-md-nb</i>	78px
Large devices	>=1200px	<i>col-lg-nb</i>	95px

Les tailles des grilles CSS

- Plusieurs formats selon l'équipement :

Equipement	Largeur	classe CSS	Taille max col
Extra small-devices	<768px	col- <i>nb</i>	auto
Small devices	>=768px	col-sm- <i>nb</i>	60px
Medium devices	>=992px	col-md- <i>nb</i>	78px
Large devices	>=1200px	col-lg- <i>nb</i>	95px

- nb* indique le nombre de colonnes (entre 1 et 12)

Exemples grilles (1/2)

- 2 colonnes de même taille quelque soit l'équipement (ne pas oublier "container" !):

```
<div class="row">  
  <div class="col-6">du texte</div>  
  <div class="col-6">autre texte</div>  
</div>
```

Exemples grilles (1/2)

- 2 colonnes de même taille quelque soit l'équipement (ne pas oublier "container" !) :

```
<div class="row">  
  <div class="col-6">du texte</div>  
  <div class="col-6">autre texte</div>  
</div>
```

- 1ère colonne : moitié de la grille pour mobiles, 33% pour desktop :

```
<div class="row">  
  <div class="col-6 col-md-4">. col-6 . col-md-4</div>  
  <div class="col-6 col-md-4">. col-6 . col-md-4</div>  
  <div class="col-6 col-md-4">. col-6 . col-md-4</div>  
</div>
```

Exemples grilles (2/2)

- colonnes inutilisées, notion d'offset :

```
<div class="row">  
  <div class="col-md-4">col-md-4</div>  
  <div class="col-md-4 offset-md-4">col-md-4 offset-md-4</div>  
</div>
```

Exemples grilles (2/2)

- colonnes inutilisées, notion d'offset :

```
<div class="row">  
  <div class="col-md-4">col-md-4</div>  
  <div class="col-md-4 offset-md-4">col-md-4 offset-md-4</div>  
</div>
```

- Autre exemple :

```
<div class="row">  
  <div class="col-md-3 offset-md-3">col-md-3 offset-md-3</div>  
  <div class="col-md-3 offset-md-3">col-md-3 offset-md-3</div>  
</div>
```