

R1.02 Développement d'interfaces Web

Marta Soare



BUT Informatique 1A

Tout le matériel pédagogique (diapos, TD, TP, QCM...) utilisé pour ce module a été fait par Céline Adobet.

Module sur Celene :

<https://celene.univ-orleans.fr/course/view.php?id=3214>

Cours 1 : Petit tour d'horizon HTML et CSS

Petit tour d'horizon HTML et CSS

Présentation du module

Courte introduction

HTML

Le DOM (Document Object Model)

CSS

Quelques bonnes pratiques

Être capable de :

- concevoir un site web conforme aux standards d'accessibilité W3C/WAI
- coder un site web (HTML, CSS)
- lire de la documentation
- s'auto-former



- 2 devoirs surveillés
- Notes en TD/TP
- QCM le vendredi
- SAE en deuxième période en lien avec le PPP



Petit tour d'horizon HTML et CSS

Présentation du module

Courte introduction

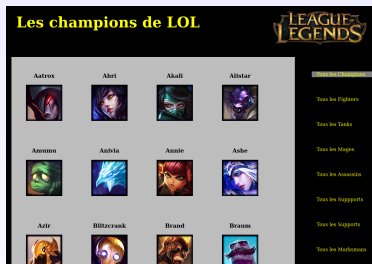
HTML

Le DOM (Document Object Model)

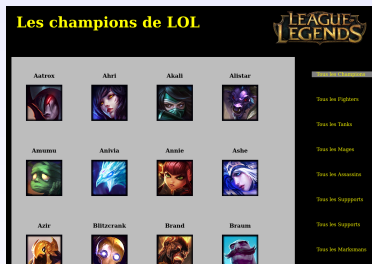
CSS

Quelques bonnes pratiques

Exemple de page web



Exemple de page web



✓ Un navigateur



Exemple de page web



- ✓ Un navigateur
- ✓ Un fichier HTML

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="utf-8"/>
  <title>Les champions de LOL</title>
  <link rel="stylesheet" href="/css/
    lol.css"/>
</head>
<body>
  <header>
    <h1>Les champions de LOL</h1>
    
  </header>
  <main>
    <section>
      <h3>Aatrox</h3>
      <a href="Aatrox.html"></a></td>
    </section>
    <section>
      <h3>Ahri</h3>
      <a href="Ahri.html">
        </a>
    </section>
    ...
```

Exemple de page web



- ✓ Un navigateur
- ✓ Un fichier HTML
- ✓ Au moins un fichier CSS

```
body{  
  display: flex;  
  flex-wrap: wrap;  
}  
  
h1{  
  flex: 1;  
  color: yellow;  
  font-size: 3rem;  
}  
  
section, main{  
  background-color: #BDBDBD;  
}  
  
main{  
  margin: 1rem;  
}  
  
section{  
  padding: 2em;  
}  
...
```

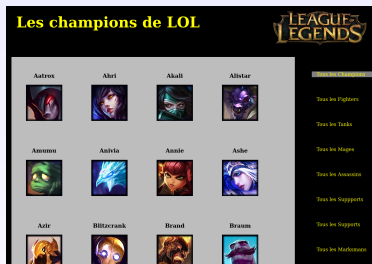
Exemple de page web



- ✓ Un navigateur
- ✓ Un fichier HTML
- ✓ Au moins un fichier CSS
- ✓ Des images

```
images
├── Aatrox.png
├── Ahri.png
├── Akali.png
├── Alistar.png
├── Amumu.png
├── Anivia.png
├── Annie.png
├── Ashe.png
├── Azir.png
└── Blitzcrank.png
```

Exemple de page web



- ✓ Un navigateur
- ✓ Un fichier HTML
- ✓ Au moins un fichier CSS
- ✓ Des images
- ✓ D'autres ressources ...

A retenir

Une page web est donnée par au moins deux fichiers :

- un fichier **HTML** définissant le **contenu** de la page ;
- au moins un fichier **CSS** contenant la **mise en forme** de la page.

☛ HTML = *Hyper Text Markup Language*
= "langage de balises pour l'hypertexte"

☛ CSS = *Cascading Style Sheet*
= "feuille de style en cascade"

Petit tour d'horizon HTML et CSS

Présentation du module

Courte introduction

HTML

Le DOM (Document Object Model)

CSS

Quelques bonnes pratiques

Vocabulaire à connaître

- Les **éléments** HTML sont utilisés pour structurer un document écrit en HTML.
- Une **balise** ouvrante/fermante permet de définir le début/la fin du contenu d'un élément.
Certains éléments sont vides (sans contenu) : dans ce cas on dit que la balise est auto-fermante.
- Certains éléments peuvent posséder des **attributs**, chacun associé à une **valeur**.

Exemple d'un élément HTML :



Exemple de fichier HTML

```
1  <!DOCTYPE html>
2  <html lang="fr">
3    <head>
4      <meta charset="utf-8"/>
5      <title>Ma page de test</title>
6      <link href="./styles/style.css" rel="stylesheet"/>
7    </head>
8    <body>
9      <p>Le HTML c'est super !</p>
10     
11   </body>
12 </html>
```

Exemple de fichier HTML

```
1  <!DOCTYPE html>
2  <html lang="fr">
3    <head>
4      <meta charset="utf-8"/>
5      <title>Ma page de test</title>
6      <link href="./styles/style.css" rel="stylesheet"/>
7    </head>
8    <body>
9      <p>Le HTML c'est super !</p>
10     
11   </body>
12 </html>
```

→ balise ouvrante de l'élément body

Exemple de fichier HTML

```
1  <!DOCTYPE html>
2  <html lang="fr">
3    <head>
4      <meta charset="utf-8"/>
5      <title>Ma page de test</title>
6      <link href="./styles/style.css" rel="stylesheet"/>
7    </head>
8    <body>
9      <p>Le HTML c'est super !</p>
10     
11   </body>
12 </html>
```

→ balise fermante de l'élément body

Exemple de fichier HTML

```
1  <!DOCTYPE html>
2  <html lang="fr">
3    <head>
4      <meta charset="utf-8"/>
5      <title>Ma page de test</title>
6      <link href="./styles/style.css" rel="stylesheet"/>
7    </head>
8    <body>
9      <p>Le HTML c'est super !</p>
10     
11   </body>
12 </html>
```

→ contenu de l'élément body

Exemple de fichier HTML

```
1  <!DOCTYPE html>
2  <html lang="fr">
3    <head>
4      <meta charset="utf-8"/>
5      <title>Ma page de test</title>
6      <link href="./styles/style.css" rel="stylesheet"/>
7    </head>
8    <body>
9      <p>Le HTML c'est super !</p>
10     
11   </body>
12 </html>
```

→ balise autofermante

Exemple de fichier HTML

```
1  <!DOCTYPE html>
2  <html lang="fr">
3    <head>
4      <meta charset="utf-8"/>
5      <title>Ma page de test</title>
6      <link href="./styles/style.css" rel="stylesheet"/>
7    </head>
8    <body>
9      <p>Le HTML c'est super !</p>
10     
11   </body>
12 </html>
```

→ un **attribut** de l'élément meta

Exemple de fichier HTML

```
1  <!DOCTYPE html>
2  <html lang="fr">
3    <head>
4      <meta charset="utf-8"/>
5      <title>Ma page de test</title>
6      <link href="./styles/style.css" rel="stylesheet"/>
7    </head>
8    <body>
9      <p>Le HTML c'est super !</p>
10     
11   </body>
12 </html>
```

→ un **attribut** de l'élément meta
→ dont la **valeur** est utf-8

Exemple de fichier HTML

```
1  <!DOCTYPE html>
2  <html lang="fr">
3    <head>
4      <meta charset="utf-8"/>
5      <title>Ma page de test</title>
6      <link href="./styles/style.css" rel="stylesheet"/>
7    </head>
8    <body>
9      <p>Le HTML c'est super !</p>
10     
11   </body>
12 </html>
```

Remarque : certains éléments peuvent avoir plusieurs attributs.


```
1  <!DOCTYPE html>
2  <html lang="fr">
3    <head>
4      <meta charset="utf-8"/>
5      <title>Ma page de test</title>
6      <link href="./styles/style.css" rel="stylesheet"/>
7    </head>
8    <body>
9      <p>Le HTML c'est super !</p>
10     
11   </body>
12 </html>
```

```
1  <!DOCTYPE html>
2  <html lang="fr">
3    <head>
4      <meta charset="utf-8"/>
5      <title>Ma page de test</title>
6      <link href="./styles/style.css" rel="stylesheet"/>
7    </head>
8    <body>
9      <p>Le HTML c'est super !</p>
10     
11   </body>
12 </html>
```

```
<body><h2>Le HTML c'est super !</body></h2>
<!-- code incorrect -->
```

```
1  <!DOCTYPE html>
2  <html lang="fr">
3    <head>
4      <meta charset="utf-8"/>
5      <title>Ma page de test</title>
6      <link href="./styles/style.css" rel="stylesheet"/>
7    </head>
8    <body>
9      <p>Le HTML c'est super !</p>
10     
11   </body>
12 </html>
```

```
<body><h2>Le HTML c'est super !</body></h2>
<!-- code incorrect -->
```

```
<body><h2>Le HTML c'est super !</h2></body>
<!-- code correct -->
```

Petit tour d'horizon HTML et CSS

Présentation du module

Courte introduction

HTML

Le DOM (Document Object Model)

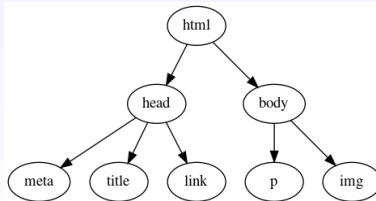
CSS

Quelques bonnes pratiques

```
1  <!DOCTYPE html>
2  <html lang="fr">
3    <head>
4      <meta charset="utf-8"/>
5      <title>Ma page de test</title>
6      <link href="./styles/style.css" rel="stylesheet"/>
7    </head>
8    <body>
9      <p>Le HTML c'est super !</p>
10     
11   </body>
12 </html>
```

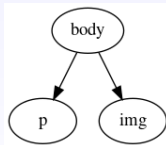
```
1  <!DOCTYPE html>
2  <html lang="fr">
3    <head>
4      <meta charset="utf-8"/>
5      <title>Ma page de test</title>
6      <link href="./styles/style.css" rel="stylesheet"/>
7    </head>
8    <body>
9      <p>Le HTML c'est super !</p>
10     
11   </body>
12 </html>
```

DOM complet



```
1  <!DOCTYPE html>
2  <html lang="fr">
3    <head>
4      <meta charset="utf-8"/>
5      <title>Ma page de test</title>
6      <link href="./styles/style.css" rel="stylesheet"/>
7    </head>
8    <body>
9      <p>Le HTML c'est super !</p>
10     
11   </body>
12 </html>
```

DOM réduit



Petit tour d'horizon HTML et CSS

Présentation du module

Courte introduction

HTML

Le DOM (Document Object Model)

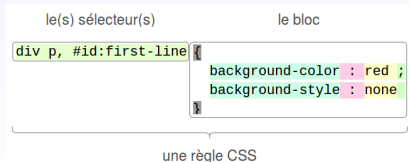
CSS

Quelques bonnes pratiques

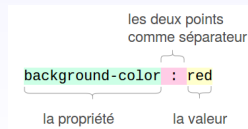
Vocabulaire à connaître

- Un fichier CSS contient une liste de **règles**
- Chaque règle est composée :
 - d'un **sélecteur**
 - d'une ou plusieurs **déclarations**
- Chaque déclaration est composée :
 - d'une **propriété**
 - d'une **valeur**

Une règle CSS



Une déclaration CSS



Un exemple de fichier CSS :

```
1  p, img {  
2    flex-basis: 60%;  
3  }  
4  
5  body {  
6    background-color: lightgray;  
7    display: flex;  
8  }
```

Un exemple de fichier CSS :

```
1  p, img {  
2    flex-basis: 60%;  
3  }  
4  
5  body {  
6    background-color: lightgray;  
7    display: flex;  
8  }
```

2 règles

Un exemple de fichier CSS :

```
1 p, img {  
2   flex-basis: 60%;  
3 }  
4  
5 body {  
6   background-color: lightgray;  
7   display: flex;  
8 }
```

3 déclarations

Un exemple de fichier CSS :

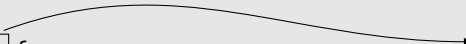
```
1  p, img {  
2    flex-basis: 60%;  
3  }  
4  
5  body {  
6    background-color: lightgray;  
7    display: flex;  
8  }
```

→ un sélecteur

Un exemple de fichier CSS :

```
1  p, img {  
2    flex-basis: 60%;  
3  }  
4  
5  body {  
6    background-color: lightgray;  
7    display: flex;  
8  }
```

un autre sélecteur



Un exemple de fichier CSS :

```
1  p, img {  
2    flex-basis: 60%;  
3  }  
4  
5  body {  
6    background-color: lightgray;  
7    display: flex;  
8  }
```



une **propriété**

Un exemple de fichier CSS :

```
1  p, img {  
2    flex-basis: 60%;  
3  }  
4  
5  body {  
6    background-color: lightgray;  
7    display: flex;  
8  }
```

une **propriété**

la **valeur** associée

Petit tour d'horizon HTML et CSS

Présentation du module

Courte introduction

HTML

Le DOM (Document Object Model)

CSS

Quelques bonnes pratiques

Bonnes pratiques en HTML

- ✓ Indenter les éléments imbriqués de son code HTML en utilisant deux espaces plutôt qu'une tabulation.
Remarque : en python, la règle est d'utiliser 4 espaces au lieu d'une tabulation.
- ✓ Spécifier la langue utilisée dans le document à l'aide de l'attribut `lang` de la balise `html` pour aider les outils de traduction et les outils de synthèse vocale par exemple.

Bonnes pratiques en CSS

- ✓ Indenter les déclarations en utilisant deux espaces plutôt qu'une tabulation.
- ✓ Chaque déclaration doit apparaître sur sa propre ligne pour un rapport d'erreur plus précis.
- ✓ Terminer toutes les déclarations par un point-virgule. Celui de la dernière déclaration de chaque bloc est facultatif, mais votre code est plus sujet aux erreurs sans lui.

**C'est tout pour aujourd'hui.
Merci pour votre attention.**

Questions?