

# HTML-css-javascript

Alice Jacquot, M1 MIAGE

# Rappels HTML

Le HTML (*HyperText Markup Language*) est un langage servant à représenter des pages web structurées.

L'extension d'un tel fichier est .html

Il fonctionne, comme xml, par des balises : <balise> contenu </balise>

Il fait partie des 3 langages traités par les navigateurs : html-css-javascript

```
<!DOCTYPE html>  
<html>
```

*Entête*

```
<head>  
<meta charset="UTF-8">  
<title>Mon Titre</title>  
<link rel="stylesheet" href="general.css">  
<script src="script.js"></script>  
</head>
```

*Head : déclarations*

```
<body>
```

```
<h1>Index</h1>  
<p> Un paragraphe</p>
```

*Body : contenu*

```
</body>
```

```
</html>
```

# Quelques balises du body

- `<h1>` à `<h6>` : un (sous-)titre
- `<p>` : un paragraphe
- `<a>` : un lien par `<a href="monlien"> texte </a>`
- `<img>` : une image par `<img src ="urlImage" alt="description"> </img>`
- `<br>` : un saut de ligne (balise auto-fermante)
- `<div>` : un bloc
- `<span>` un bloc (plutôt à l'intérieur d'une ligne)

# Attributs de balises

Les balises peuvent contenir des *attributs* :

```
<img src ="urlImage" alt="description" id="cetteImage"> </img>
```

src, alt et id sont des attributs de l'élément

Il s'écrivent sous la forme **attribut="valeur"** et sont séparés par des espaces.

L'attribut *id* peut être appliqué à tout élément et permet d'y faire référence

# Utilité du div et span

Les balises `<div>` et `<span>` permettent d'appliquer des styles (css) ou du contenu dynamique (JS) sur une partie d'un paragraphe

# Formulaire html

Les formulaires (<form>) regroupe un ensemble de champs (<input>), pouvant être de divers types et permettent l'envoi d'une requête Http avec les informations contenues on clique du bouton (<boutton>) de soumission (type="submit")

```
<form action="/takeDamage" method="post">
  <label for="nbDamage">Dégats :</label>
  <input type="text" id="nbDamage" name="damage">
  <div>
    <button type="submit">Souffrir</button>
  </div>
</form>
```

---

Dégats :

Le clic sur le bouton “Souffrir” lancera une requête HTTP en POST vers l’url /takeDamage avec le paramètre “damage” valant la valeur indiquée



# CSS

CSS (*Cascading Style Sheets*) est un langage utilisé pour décrire l'affichage de documents html (ou xml).

Il permet un style cohérent à travers un site web entier, comme la personnalisation d'éléments particuliers.

Il applique des styles à des éléments, ou type d'éléments, html

Il peut être utilisé à l'intérieur d'un fichier html, ou comme un fichier externe, son extension est alors .css

*Type d'élément html*

```
body {  
    background-color: lightblue;  
}
```

*caractéristique*                      *valeur*

```
h1 {  
    color: white;  
    text-align: center;  
}
```

```
p {  
    font-family: verdana;  
    font-size: 20px;  
}
```

Bloc de CSS

# Inclure du CSS

On peut inclure du CSS de 3 manières :

- Fichier externe : dans `<link>` de la head html

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

# Inclure du CSS

On peut inclure du CSS de 3 manières :

- Fichier externe : dans <link> de la head html
- Bloc interne : dans <style> de la head html

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

# Inclure du CSS

On peut inclure du CSS de 3 manières :

- Fichier externe : dans `<link>` de la head html
- Bloc interne : dans `<style>` de la head html
- En ligne, pour un élément particulier, avec l'attribut `style` dans une balise

```
<!DOCTYPE html>  
<html>  
<body>
```

```
<h1 style="color:blue;text-align:center;">  
This is a heading  
</h1>  
<p style="color:red;">  
This is a paragraph.  
</p>  
</body>  
</html>
```

# Les sélecteurs : des styles plus précis

On peut appliquer un style à :

- un type d'élément html
- un élément particulier, par son id
- les éléments de classe donnée, par l'attribut classe="ceStyle"
- les éléments d'un type particulier de classe donnée
- Tous les éléments

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

#idElement {
  text-align: center;
  color: red;
}

.ceStyle {
  text-align: center;
  color: red;
}

p.ceStyle {
  text-align: center;
  color: red;
}

* {
  font-family: verdana;
}
```

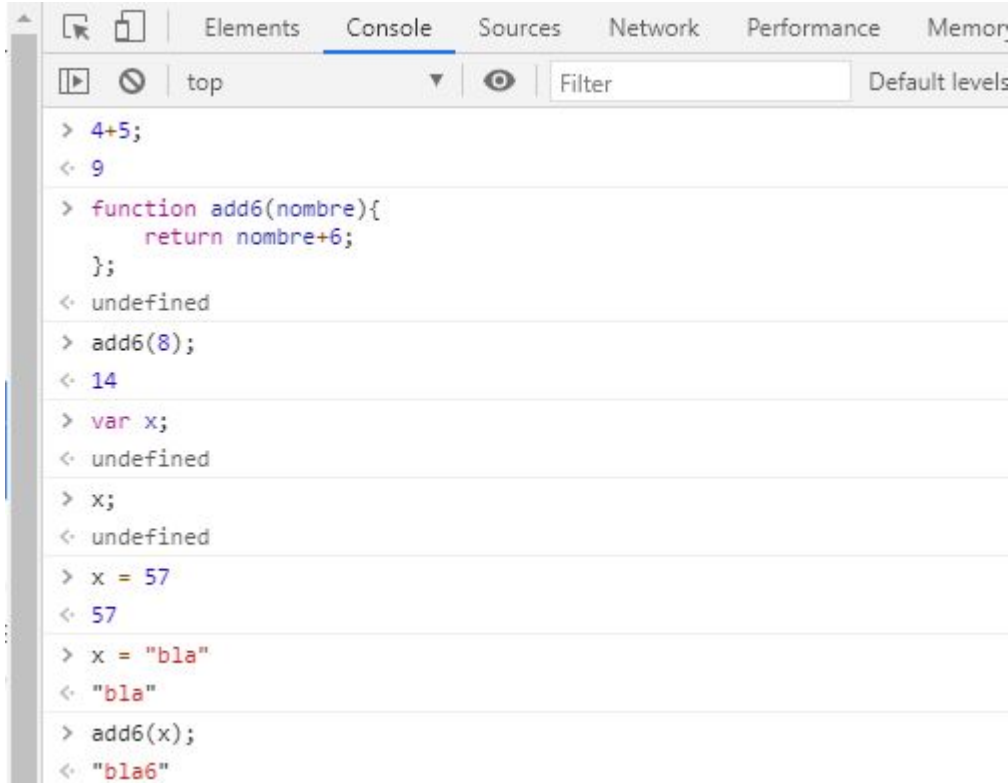
# JavaScript

Le JavaScript (abrégé JS) est un langage de script interprété par les navigateurs (“modernes”).

C'est un langage qui n'est pas fortement typé, orienté objet à prototype : il y a des bases du fonctionnement objet mais on ne définit pas de classe, chaque objet définit ses constructeurs et héritage (d'objet).

Le langage n'a pas de lien direct avec Java, si ce n'est une syntaxe proche.

# Langage interprété, utilisation en console



The screenshot shows a web browser's developer console with the 'Console' tab selected. The console displays the following sequence of commands and their outputs:

```
> 4+5;
< 9

> function add6(nombre){
    return nombre+6;
};
< undefined

> add6(8);
< 14

> var x;
< undefined

> x;
< undefined

> x = 57
< 57

> x = "bla"
< "bla"

> add6(x);
< "bla6"
```



# Inclure du JS à une page web

JavaScript sert à rendre les pages web dynamiques, les instructions sont pour cela incluses dans des **balises** `<script>` du document html, qui peuvent appartenir à la head comme au body

Un script peut également être déclenché par une action signalée dans un attribut de balise : `<p id="autreP" onclick="alert('Bonjour');"> Clic </p>`

# document.getElementById

On peut accéder ou modifier le contenu d'un élément par :

```
document.getElementById("nomElement")
```

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Exemple</title>
<script> alert("coucou");</script>
```

```
</head>
<body>
```

```
<p id="p1">
Hello
</p>
```

```
<script>
document.getElementById("p1").innerHTML = "Bonjour";
</script>
```

```
</body>
</html>
```

```
<body>
```

```
<p id="p1">
```

```
Hello
```

```
</p>
```

```
<p id="autreP" onclick="alert('reBonjour'); document.getElementById('p1').innerHTML ='Au revoir';">
```

```
Clique-moi !
```

```
</p>
```

```
<script>
```

```
p1.innerHTML = "Bonjour";
```

```
</script>
```

```
</body>
```

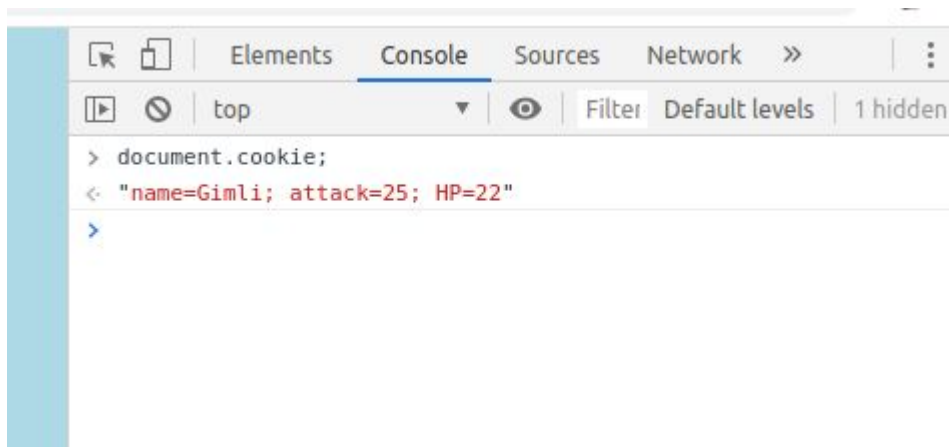
```
</html>
```

# Les cookies

On récupère les cookies associées à la requête par :

`document.cookie`

qui renvoie une chaîne de caractères.



```
> document.cookie;
< "name=Gimli; attack=25; HP=22"
>
```

# Éléments statiques

Les fichiers html, css et js sont des éléments statique de notre application web :

le serveur d'application les rend juste disponible

Conséquence : il n'est pas nécessaire de redémarrer le serveur lorsqu'on les modifie ! Attention par contre à bien réactualiser les pages, et éventuellement à vider votre cache (la navigation privée peut être utile)