# Ajouter de l'interactivité : Introduction à JavaScript

# 1. Première approche

## 1.1. Javascript?

Javascript , souvent raccourci en JS , est un langage de programmation interprété, tout comme Python. Il est une des briques angulaire de HTML 5 , avec HTML et CSS .

A chaque fois qu'une page fait plus que simplement afficher du texte et des images - un contenu *statique*, copmme par exemple affiocher des animations, des cartes interactives, des menus déroulants, etc, il y a de bonnes chances pour que ce soit un script JavaScript qui en soit la raison.

Comme souvent, un tutoriel complet sur Javascript est disponible sur MDN Web Docs. Vous y trouverez de nombreuses explications absentes de ce cours.

1.2. Une première approche : modifier le texte d'un paragraphe par un click.



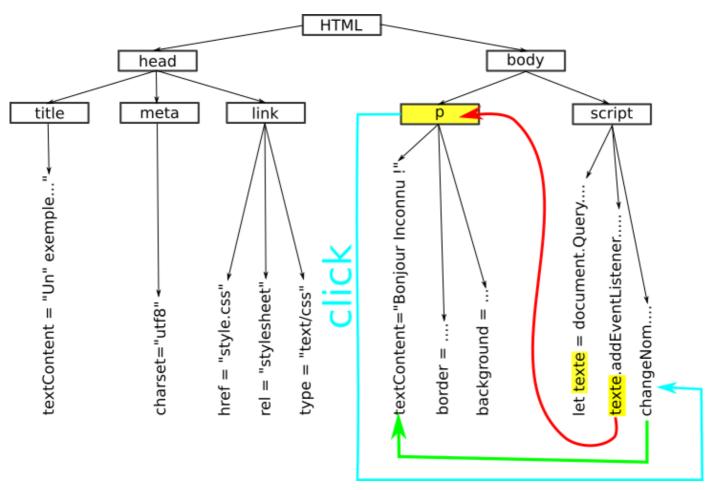
Cette partie est intégralement traitée sous la forme d'une vidéo :

Observons les codes de la page web (corrigé pour la position de la balise <script> 1):

```
<!DOCTYPE html>
<html lang="fr">
   <head>
       <meta charset="utf-8" />
       <title>Un exemple de l'utilité de JS </title>
       <link href="style.css" rel="stylesheet"</pre>
type="text/css" />
   </head>
   <body>
       Bonjour Inconnu !
           let texte = document.querySelector('p');
           texte.addEventListener('click',
changeNom);
            function changeNom(){
               let name = prompt("Entrez votre nom
:");
```

```
p{
    text-align: center;
    border: 2px solid rgba(0,0,200,0.6);
    background: rgba(0,0,200,0.3);
    color: rgba(0,0,200,0.6);
    box-shadow: 1px 1px 2px rgba(0,0,200,0.4);
    border-radius: 10px;
    padding: 3px 10px;
    cursor:pointer;
}
```

Ainsi que l'arbre du DOM correspondant :



La construction d l'arbre du DOM se fait au fur et à mesure de la lecture du fichier .html, et est présentée sur le schéma ci-dessus de gauche à droite. La balise <script> étant la dernière balise présente dans <body>, ell est lue en dernier.

L'exécution du code JS est donc la suivante :

- Création d'une variable texte qui référence la première balise trouvée dans le DOM (surlignage jaune ).
- Ajout à l'élément sélectionné d'un écouteur d'évènement qui attends une action clique-gauche (flèche rouge).
- Fin de la lecture du code, avec création d'une référence vers une fonction changeNom, qui n'est pas exécutée.

Une fois la construction du DOM terminée, le gestionnaire d'évènement général du navigateur écoute toutes les actions de l'utilisateur (mouvement souris, clique gauche ou clique droit, appuis sur le clavier, voire écoute micro si celui-ci a été activé....)

Un *clique-droit* sur l'élément déclenche l'exécution de la fonction changeNom (flèche bleue , qui elle même va modifier l'attribut textContent de l'élément (flèche verte ).

# 2. Bases de Javascript

#### 2.1. Séparer le code

Pour conserver la cohérence de HTML5, on a séparé le code HTML, définissant la nature des élements de la page, du CSS, qui définit la forme que prendront ces éléments.

Il faudra **faire de même** avec le code javascript, en le séparant du code html en renseigant dans la balise <script> l'attribut src avec l'adresse (absolue ou relative) du fichier contenant le code javascript. Ainsi si le fichier javascript s'appelle mon\_fichier.js et est

dans le même dossier que le fichier html, il suffira d'écrire la balise suivante :

```
<script src="mon_fichier.js"> </script>
```

Un fichier javascript est donc simplement un fichier d'extension .js contenant du code javascript.

L'exemple précédent peut donc être classé en trois fichiers :

Fichier 'pagesimple.html'

```
Fichier `style.css`
```

```
p{
    text-align: center;
    border: 2px solid
rgba(0,0,200,0.6);
    background:
rgba(0,0,200,0.3);
    color: rgba(0,0,200,0.6);
    box-shadow: 1px 1px 2px
rgba(0,0,200,0.4);
    border-radius: 10px;
    padding: 3px 10px;
    cursor:pointer;
```

Fichier `code.js`

```
let texte =
document.querySelector('p');
texte.addEventListener('click',
   changeNom);

function changeNom(){
    let name = prompt("Entrez
votre nom :");
    texte.textContent =
"Bonjour "+name+" !";
}
```

# 3. Utilisation de l'inspecteur d'objet

L'inspecteur d'objet permet d'avoir accès à une console javascript, fort utile pour aider au déboggage de code.

Celle-ci est disponible une fois l'inspecteur d'objet ouvert :



# prise en main de la console js

- 1. Ouvrez la console javascript de votre navigateur, sur une page au choix.
- 2. Tapez le code suivant :

```
>>> function plop(){
  console.log("Plop !");
}
```

3. Exécutez la fonction plop:

```
>>> plop();
```

Que se passe-t-il?

4. Tapez maintenant le code suivant :

```
>>> let cible = document.querySelector("h1");
>>> cible.addEventListener('click', plop);
```

- 5. Cherchez la première balise <h1> de la page et cliquez sur cet élément. Que se passe-t-il ?
- 6. Tapez le code suivant :

```
>>> cible.addEventListener('mouseover', plop);
```

Comment faire pour déclencher l'écriture dans la console à partir de cet événement ?

# Utilisation de console.log()

Il est parfois difficile de voir comment s'exécute un code javscript. L'utilisation de console.log(message); dans votre code, avec comme message une chaîne de caractère explicative, est parfois un bon moyen de vérifier qu'un bloc de code est exécuté par javascript.

#### 3.1. Déclaration des variables

Javascript est un langage faiblement typé. Cela signifie que lorsqu'on déclare une variable, il n'est pas nécessaire de déclarer son type. Il suffit de déclarer son **nom**, en ajoutant devant les mots-clés let ou var selon la situation<sup>2</sup>. On peut aussi **déclarer et initialiser** une variable dans le même temps :

```
let x ; // x est déclarée, mais pas initialisée
let y = 10; // y est déclarée et initialisée à la valeur 10
var z ="toto"; // z est déclarée et initialisée avec la chaîne "toto"
```

Sachez seulement que :

- let définit des variables dont la portée (=l'ensemble des lignes où la variable est connue et utilisable) est celle du bloc;
- var définit des variables dont la portée est celle de la fonction.

Pour ceux intéressés les exemples ci-dessous sont parlants :

### **▼** Exemples de portées²

## 3.2. Opérations arithmétiques

Les opérations arithmétiques de base sont les même qu'en Python : +, -, × et ÷ sont représentées par +, -, \* et /.

Ainsi le calcul  $3+\frac{4}{2-7}$  est effectué par :

```
>> 3 + 4/(2-7)
2.2
```

Les opérations de divisions euclidiennes et de reste(modulo) sont utilisables comme en Python, ainsi que les exposants :

```
>> 22%5
2
>> 22//5
4
>> 5**3
125
```

Les fonctions mathématiques

### 3.3. Fonctions

# 3.4. Structures conditionnelles

La structure conditionnelle de base est

### 3.5. Boucles bornées

### 3.6. Boucles non-bornées

<sup>1.</sup> une vidéo détaillant pourquoi la balise <script> doit être placée dans la balise <body> est disponible ici. 🖯

<sup>2.</sup> La question de l'utilisation de 1et ou var dépasse largement le cadre de ce cours. On trouvera ici un début d'explication. 🗸 🗸