

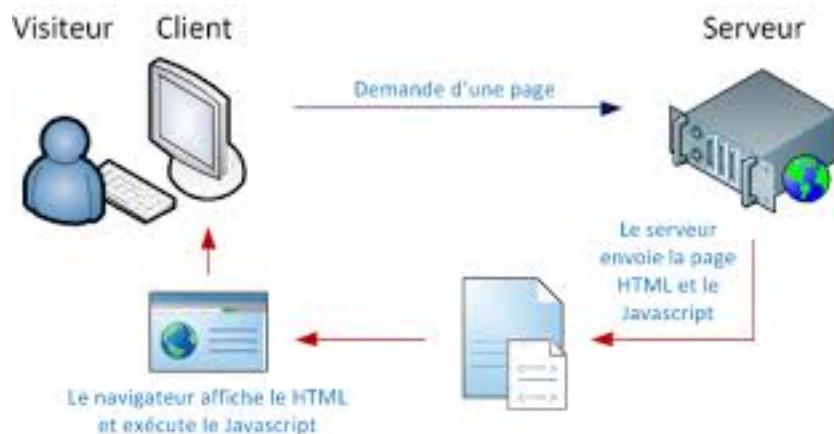
4.3

Créer de l'interaction avec le JavaScript

NSI 1ÈRE - JB DUTHOIT

4.3.1 Introduction

JavaScript est un des langages les plus importants du web. Il est utilisé pour gérer l'interactivité dans un navigateur web (côté client) mais peut aussi être utilisé côté serveur via la plateforme NodeJS.



4.3.2 JavaScript et HTML

Pour utiliser du JavaScript, on peut :

- Soit placer directement le script dans le programme HTML. On place les scripts plutôt à la fin du document pour que la page ait le temps de se charger :

```
<script type="text/javascript">
    alert('Bonjour à tous');
</script>
```

- Soit dans un fichier externe d'extension **.js** avec l'attribut src :

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

4.3.3 Les principales commandes JavaScript

JavaScript est similaire à Python par de nombreux aspects, mais utilise une syntaxe un peu différente.

La table suivante indique les principaux types de données disponibles en JavaScript. La notion de tuple n'existe pas ; on utilise selon les cas des tableaux ou des dictionnaires, mais ceux-ci sont des types mutables.

Nombres	entiers (123) ou réels (123.456)
Chaînes	'texte' ou "texte"
Booléens	true ou false
Tableaux	t = [1, 2, 3], t[1], t.length
Dictionnaires	d = a : 1, b : 2, c : 3, d.a

La table suivante donne la liste des principales instructions. Comparé à Python, une différence importante est que les blocs d'instructions ne sont pas délimités par l'indentation, mais par des accolades.

Déclaration	<code>let var ou let var = expression</code>
Affectation	<code>var = expression</code>
Conditionnelle	<code>if (expression) { instructions }</code> <code>if (expr) { instr } else { instr }</code>
Boucle bornée	<code>for (i = 0; i < n; i+=1) { instructions }</code>
Boucle non bornée	<code>while (expression) { instructions }</code>
Fonction	<code>function f(parametres) { instructions }</code>
Valeur de retour	<code>return expression</code>
Commentaires	<code>// commentaire</code>

⚠ Les expressions sont similaires à celles de Python, avec la différence importante que les opérateurs logiques or, and et not s'écrivent respectivement ||, && et !.

⚠ La division entière n'existe pas (// est utilisé pour les commentaires).

4.3.4 le DOM : modifier une page Web

Le navigateur affiche une page Web à partir du fichier HTML. Il réalise cet affichage avec un type construit complexe appelé **DOM** pour **Document Object Model**.

Pour modifier une page Web, on peut utiliser JavaScript qui va modifier le DOM.

Tout élément HTML peut avoir un identifiant unique, grâce à l'attribut **id**. La fonction JavaScript `document.getElementById` permet de trouver un élément à partir de cet identifiant :

Par exemple, `e = document.getElementById('msg')` en JavaScript va identifier l'élément situé entre les deux balises dans le code html `<p id="msg">...</p>`.

Ensuite, on peut accéder en lecture et en écriture à certains de ces attributs :

`e.hidden` Booléen qui indique si l'élément est caché ou non

`e.value` Valeur d'un champ d'entrée `<input>`

`e.innerHTML` le code HTML du contenu de l'élément

`e.outerHTML` le code HTML de l'élément et de son contenu.

Exercice 4.1

Recopier le script suivant :

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8" />

    <link href="style.css" rel="stylesheet">

    <title>Maths et NSI</title>
  </head>
```

```
<body>  
  
<p id="ex1"> Bonjour à tous </p>  
<p id="ex2"> Nous détestons la NSI </p>  
</body>  
  
<script>  
....  
....  
</script>  
</html>
```

Le navigateur affichera donc :

•

Bonjour à tous
Nous détestons la NSI

1. Créer un script JavaScript pour qu'il soit affiché ceci, sans intervenir sur le langage HTML :

Bonjour à tous
Nous adorons la NSI

2. ajouter du code JavaScript pour que la première ligne ne soit plus affichée :

Nous adorons la NSI

4.3.5 Gestion d'un événement sur JavaScript

De très nombreux attributs événementiels sont associés à des balises HTML et utilisé en lien avec JavaScript. En voici quelques-uns :

- **onclick** : réponse à un clic de la souris
- **onchange** : Changement d'une valeur de champ (lorsqu'on utilise une liste déroulante par exemple ...)
- **oninput** : Champ de formulaire modifié
- **onkeyup** : Touche relachée dans un formulaire.
- **onmouseover** : Le pointeur de la souris est sur l'élément
- **onmouseout** : Le pointeur de la souris n'est plus sur l'événement.

On lie un élément du fichier HTML au déclenchement d'une fonction avec un attribut événementiel (**onclick** par exemple)

Voici 4 exemples de gestion d'événements :

1. Changer de couleur

```
<input type="button" value="Changer couleur" onclick="ChCoul()"/>
<script type="text/javascript">
    function ChCoul(){
        const coul = document.body.style.backgroundColor;
        if (coul == "green"){
            document.body.style.backgroundColor = "red";
        }
        else {document.body.style.backgroundColor = "green";}
    }
</script>
```

2. Entrer des informations et les afficher

```
<script type="text/javascript">
    var msg1 = "Bonjour", msg2 = "vous allez bien ?";
    var nb = 2020;
    alert(msg1);
    confirm(msg2);
    nom = prompt("Quel est votre nom ?", "nom");
    var msg3 = new Array(3);
    msg3[0] = msg1;
    msg3[1] = " "+nom;
    msg3[2] = " nous sommes en " +nb;
    alert(msg3);
</script>
```

3. Création d'un compte à rebours

```
<html>
    <head>
        <title> ma page </title>
        <script type="text/javascript">
            function rebours(x){
                while (x>= 0) {
                    alert(x);
                    x -- ;
                }
                n = prompt("Saisir une valeur de n", "nombre")
            }
        </script>
    </head>
    <body onLoad = "rebours(n)">
        <p> page avec JS </p>
```

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

4. Récupérer la réponse à un formulaire

```
<html>
    <select id='choix' name="lang" onchange ="selection()">
        <option value="fr"> Français </option>
        <option value="zh"> Chinois </option>
        <option value="it"> Italien </option>
    </select>

    <script type = "text/javascript">
        function selection(){
            const choix = document.getElementById('choix').value;
            alert(choix);
        }
    </script>
</html>
```

Exercice 4.2

Le but de l'exercice est de créer un programme JavaScript qui demande de saisir un montant en euro puis une monnaie (dollar, livre, yen) et qui donne le résultat de la conversion, comme le montre les deux screens suivants :

Avant l'appui sur "calculer"

et après !

On peut scinder l'exercice en plusieurs sous-problèmes :

1. Commencez par créer le formulaire seul
2. Lorsque l'on clique sur le bouton "Calculer", un message simple, comme "hello" apparaît
3. Lorsque l'on clique sur le bouton "Calculer", le montant entré en euro apparaît
4. Lorsque l'on clique sur le bouton "Calculer", la devise apparaît ("yen" par exemple)
5. Vous êtes prêts à faire l'exercice !

Remarque

`document.getElementById('name').checked` renvoie un booléen qui vaut `true` si le bouton radio dont l'id est "name" a été coché...

 **Exercice 4.3**

Utiliser le JavaScript pour créer un événement sur un élément HTML (du type le mot change de couleur au passage de la souris)