BUT Science des Données 2 - Semestre 4

COMPETENCE 1 : Traiter des données à des fins décisionnelles

Ressource : **Programmation web (VCOD)**

F.GARNIER

PLAN DE LA RESSOURCE

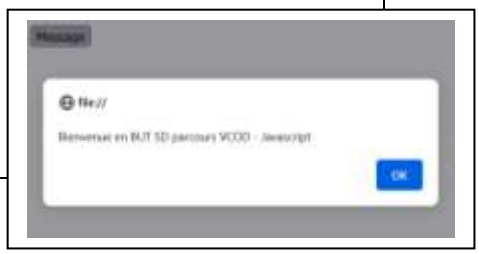
Chapitre 1 : Le langage de programmation JAVASCRIPT

1. Introduction

Javascript est un langage de programmation impératif (*variables assignées plusieurs fois, suite d'instructions, boucles*), fonctionnel (*découpage modulaire et* ***utilisation d'une fonction comme paramètre***) et légèrement orienté objet (prototypes, objets et utilisation de classes)

Différence Java / Javascript : Java est un langage de programmation de type POO (OOP : Object Oriented Programming) tandis que JavaScript est un langage de script POO. Java permet de créer des applications qui sont exécutées sur une machine ou un navigateur virtuel tandis que le code JavaScript est exécuté **uniquement sur un navigateur.**

```
<html>
   <head>
        <script type="text/javascript">
       function ouvrirPopup()
       {
              alert("Bienvenue en BUT SD parcours VCOD - Javascript")
       }
      </script>
   </head>
   <body>
      <button onclick="ouvrirPopup()"> Message </button>
   </body>
</html>
```



Deux versions couramment utilisées :

- ES5 (ECMAScript 5 = normes pour les langages de script) supportée par tous les navigateurs modernes ;
- ES6 (ES2015) apporte de nombreuses améliorations au langage (voir http://es6-features.org), mais est un peu moins bien supportée

2. Type de données

Typage faible dynamique : le type des expressions est non connu à l'écriture mais connu à l'exécution (comme en Python).

Types existants :

- types primitifs : `number, boolean, string, null, undefined`
- types complexes : `object, array`

3. Syntaxe

Elle est similaire à celle du langage C, et donc de PHP ou de JAVA avec des « ; » en fin de ligne et les {}.

Opérateurs :

- **number** : `*  +  -  /`
- **boolean** : `&&  ||`
- **string** : `+`

Pour connaitre le type d'une variable : **typeof**`(nomVariable)`

4. Déclaration de variables et constantes

```
var x       //portée = fonction englobante
let x       // portée = bloc (ES6)
const x     // portée = bloc (ES6)
```

5. Conditions

```
if (booleen)
{
      // code si vrai
}
else
{
      // code si faux
}
```

```
// condition ? <expression si vrai> : <expression si faux>
      age = 20
      alert("Je suis " + (age>=18 ? "majeur" : "mineur"));

// switch
      switch (expression)
      {
            case valeur1 :
                              instructions1 ;
                              break;
            case valeur2 :
                              instructions2 ;
                              break;
            default :
                              instruction3 ;
                              break;
      }
```

6. Itérations (boucles)

```
var i = 0 ;
while (i<3)
{
    console.log(i);   // écrit dans la console F12
    if (i == 1)
    {
          break;
    }
    i++ ;
}
```

```
for (var i=0;i<3;i++)
{
    console.log(i);
}
```

```
var etudiants = ["Pierre","Paul","Jacques"]
for (var i=0;i<etudiants.length;i++)
{
    console.log(etudiants[i]);
}
```

```
etudiants.forEach(function (value)
{
    console.log(value);
});
```

Aller plus loin : https://hacks.mozilla.org/2015/04/es6-in-depth-iterators-and-the-for-of-loop/

7. Découpage modulaire (plusieurs possibilités)

```
function saluer(nom)
{
      return "Salut " + nom
}

alert(saluer("François"));
```

```
var convert_to_min = function(secondes)
{
      return secondes / 60
}

alert(convert_to_min(3600));
```

```
var etudiant =
{
      nom : "François",
      age : 20,
      present : function()
      {
            return "Je suis présent !";
      }
}

alert(etudiant.present());
```

Une procédure en javascript est une fonction qui n'a pas de clause return !

8. Portée des variables

```
var a = function ()     // ici c'est une procédure car pas de return
{
      var b = 3;
}

a();        // Appel de procédure
alert(b);  // ReferenceError, b is not defined
```

```
var exterieur = "Salut";
var maProcedure = function ()
{
      exterieur;
      var interieur = 3;
}

maProcedure();
alert(interieur);  // ReferenceError, interieur is not defined
```

```
var a = "Salut";  // variable globale
var maFonction = function ()
{
      var a = "Bonjour";  // variable locale
      return a;
}

alert(maFonction());   // retourne "Bonjour"
alert(a);              // retourne "Salut"
```

```
for (var i=0;i<10;i++)
{
      var maVar = true;
      console.log(maVar);
}
console.log(i);        // affiche 10
console.log(maVar);    // affiche true
```

```
function test()
{
      if (1 === 1)    // comparaison valeurs strictes
      {
            let maVar = true;
            console.log(maVar);
      }
      console.log(maVar);        // ERREUR
}
```

Remarque : comparaisons de valeurs strictes : http://adrienjoly.com/cours-javascript/tp02.html
Exemple ('1' === 1) => false ! ALORS que ('1' == 1) => true !

9. Débogage

F12 de votre navigateur ! Sinon **`EsLint`** : est un outil d'analyse de code statique pour identifier les modèles problématiques trouvés dans le code JavaScript. Cet outil est configurable et intégrable aux IDE (Eclipse, IntelliJ, VSCode, …). Lien : https://eslint.org/

10. Javascript dans un navigateur

La fenêtre du navigateur est un objet `window` qui contient un ensemble de méthodes et de propriétés :

- Dimension de la fenêtre : largeur et hauteur
- Localisation (url courante, historique)
- Alertes (pop-ups)
- Stockage de variables, de données
- Timers
- Accès aux capteurs, …

11. DOM

Le Document Object Model ou DOM (pour modèle objet de document) fournit une représentation structurée de document sous forme d'arbre pour les documents HTML, XML ou SVG. Le DOM représente le document comme un ensemble de nœuds et d'objets possédant des propriétés et des méthodes. Avec Javascript, il est possible de manipuler le DOM :

```
// Pour sélectionner un élément
document.body // Récupère l'élément body
document.getElementById('demo') // Sélectionne l'élément avec l'id demo
document.querySelector('.demo') // Sélectionne le premier élément correspondant au
sélecteur CSS

// Pour sélectionner plusieurs éléments
document.getElementsByClassName('demo') // Sélectionne les éléments avec la class demo
document.getElementsByTagName('p') // Sélectionne les éléments <p>
var elements = document.querySelectorAll('.demo') // Sélectionne les éléments
correspondant au sélecteur CSS
// Ces méthodes renvoient un objet NodeList enumerable
// On peut parcourir cette liste d'éléments comme un tableau
for (var i = 0; i < elements.length; ++i) {
    var element = elements[i] // objet de type Element
}
```

Pour aller plus loin sur le DOM : https://grafikart.fr/tutoriels/dom-774

Pour aller plus loin en javascript :
https://grafikart.fr/formations/debuter-javascript

- ⇨ TD n°1 Application Météo HTML/CSS/js/GitHub