Les bases en Javascript

Table des matières

1) Le Javascript

1.1) Introduction

Le langage HTML (ou Hypertext Markup Language) permet de créer la structure d'une page internet. Grâce à un jeu de balises, il permet de décomposer la page comme un traitement de texte : titre, sous-titre, section,...On peut par exemple indiquer que l'on utilise un titre avec les balises ouvrante et fermante<h1></h1>ou indiquer l'utilisation d'une énumération avec
et
pour chaque item. CSS (ou Cascade Style Sheets) permet la mise en forme des différents éléments HTML. On peut ainsi modifier la couleur ou la police des éléments précédents. Les pages ainsi créées sont statiques c'est-à-dire que l'interaction avec l'utilisateur est réduite à la possibilité de cliquer sur un lien hypertexte présent dans la page. On peut dynamiser la page internet de deux manières :

- soit du côté serveur avec PHP (ou Hypertext Preprocessor) qui peut, par exemple, ajouter le résultat d'une requête à une base de données dans la page qui sera fournie au navigateur ;

- soit du côté client avec JavaScript qui peut, par exemple, faire apparaître des info-bulles contextuelles ou réaliser des animations.

1.2) Présentation

JavaScript est un langage créé en 1995 par Brendan Eich qui travaille chez Netscape Communication Corporation principal concurrent à l'époque d'Internet Explorer. JavaScript dont le nom est une référence à Java de Sun Microsystems, ne doit pas être confondu avec Java. En effet, Java est fortement typé et précompilé alors que JavaScript est à typage dynamique et interprété. JavaScript est un langage orienté objet.

Brendan Eich

2) Premier programme

2.1) Code

Ne dérogeons pas à la règle traditionnelle qui veut que tous les tutoriels de programmation commencent par afficher le texte « Hello World ! » (« Bonjour le monde ! » en français) à l'utilisateur. Voici un code HTML simple contenant une instruction (nous allons y revenir) JavaScript, placée au sein d'un élément `<script>` :

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8" />
<title>Hello World!</title>
</head>

<body>

<script>
alert('Hello world!'); // affichage
instruction_3;
</script>

</body>
```

2.2) Observations et changements par rapport à Python

- Un élément `<script>` est présent dans la page HTML : c'est cet élément qui contient le code Javascript ;
- Le code Javascript ne contient qu'une seule instruction : l'instruction `alert()`. Cette instruction permet d'afficher une boîte de dialogue contenant un message. Le message en question est placé entre apostrophes.
- La sortie avec la fonction `alert()` peut être remplacé par une sortie sur la console grâce à l'instruction `console.log('Hello world!')`. Il faut se rendre sur la console du navigateur(en général : menu puis développement web puis console web) ou encore sur le corps de la page web avec l'instruction `document.write('Hello world!')`
- Chaque instruction doit se terminer par un point virgule ;

- Le Javascript, contrairement au Python n'est sensible ni à l'indentation, ni aux espaces ;
- Les commentaires monolignes commencent par deux slashs (tandis qu'en Python ils commencent par #) et les commentaires multilignes se font par /* et se termine par */ .
  Remarque : on peut aussi placer un commentaire multiligne sur une seule ligne
  Exemple :

```
/* Ce script comporte 3 instructions :
- Instruction 1 qui fait telle chose
- Instruction 2 qui fait autre chose
- Instruction 3 qui termine le script
*/
instruction_1;
instruction_2;    /*Commentaire */
instruction_3; // Fin du script
```

Exercice 1 : afficher son Prénom sur une boite de dialogue puis dans la console.

3) Où placer son code ?

Il est possible, et même conseillé, d'écrire le code JavaScript dans un fichier externe, portant l'extension.js. Ce fichier est ensuite appelé depuis la page Web au moyen de l'élément `<script>` et de son attribut `src` qui contient l'URL du fichier.js. Ce code **fichier.js** s'insère dans l'HTML de la manière suivante :

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

Il est conseillé de placer les éléments `<script>` juste avant la fermeture de l'élément `<body>`

Exemple :

Code HTML : hello.html

```
<!DOCTYPE html>
<html>
<head>
        <title>Hello World!</title>
</head>
<body>
        <p> Lorem ipsum dolor sit amet.<p>
        <script src="hello.js"></script>
</body>
</html>
```

Code Javascript : hello.js

```
alert('Hello world!');
```

4) Les variables

4.1) Déclaration de variables et type

Nous déclarerons une variable avec le mot clé `let`.

Exemple : avec la variable monAge

```
let monAge ;
```

Remarque : il existe un autre mot clé pour déclarer une variable : `var`. La différence entre `let` et `var` est une différence de portée. `let` a une portée de bloc, c'est-à dire entre deux accolades tandis que `var` a une portée plus globale.

Nous déclarerons une constante *Pi* avec :

```
const Pi=3.14 ;
```

En JavaScript, il existe 7 types de valeurs différents. Chaque valeur qu'on va pouvoir créer et manipuler en JavaScript va obligatoirement appartenir à l'un de ces types. Ces types sont les suivants :

- String ou « chaine de caractères » en français ;
- Number ou « nombre » en français ;
- Boolean ou « booléen » en français ;
- Null ou « nul / vide » en français ;
- Undefined ou « indéfini » en français ;
- Object ou « objet » en français ;

4.2) Les conversion de types

Comme en Python, on peut avoir besoin de convertir une variable d'un type vers un autre. En Javascript, il existe de nombreuses manière de réaliser cette tâche mais les plus simples sont ces fonctions :

- `parseInt(variableAConvertir, base)}` ; Cette fonction se charge de convertir en entier (et dans une certaine *base*) la *variableAConvertir* ;
- `parseFloat(variableAConvertir)` ; Cette fonction se charge de convertir en flottant la *variableAConvertir* ;
- `parseString(variableAConvertir)}` ; Cette fonction se charge de convertir en chaîne de caractère la *variableAConvertir* ;

4.3) Un intermède avec la fonction prompt

La fonction `prompt("Question")` crée une fenêtre où apparaît la *Question* et où l'utilisateur dispose d'un champ pour y répondre. La réponse tapée est considérée par Javascript comme un chaîne de caractère.

Exemples

```
let name=prompt("Quel est votre prénom");
alert(x) ;
```

```
let x=parseInt(prompt("Tapez un valeur x?"),10);
alert(x) ;
```

5) Les opérations arithmétiques

Les opérateurs arithmétiques vont nous permettre d'effectuer toutes sortes d'opérations mathématiques entre les valeurs de type **Number** contenues dans différentes variables.

Nous allons pouvoir utiliser les opérateurs arithmétiques du tableau 1 en JavaScript :

Opérateur	Nom de l'opération associée
+	Addition
-	Soustraction
*	Multiplication
/	Division
%	Modulo (reste d'une division)
**	Puissance

Tableau 1 – Opérateurs

Exercice 2 :

- écrire un script qui demande une somme en dollars, qui la convertit en euros et qui affiche la somme en euros à l'aide la fonction alert ;
- écrire un script dans lequel vous afficherez la valeur de $x^2 + 3y \times z$. Il faudra déclarer quatre variables et afficher le résultat à l'aide de la fonction alert.

6) Les instructions conditionnelles et les boucles

6.1) if

Modèle

```
if(condition){
        instruction1;
}else{
        instruction2;
}
```

Exemple :

```
let x=parseInt(prompt("x?"),10);
if(x > 1){
 alert('x est strictement supérieure à 1');
}else{
 alert('x est inférieure ou égale à 1');
}
```

6.2) Les opérateurs de conditions

Opérateur	Définition
==	Permet de tester l'égalité sur les valeurs
===	Permet de tester l'égalité en termes de valeurs et de types
!=	Permet de tester la différence en valeurs
<>	Permet également de tester la différence en valeurs
!==	Permet de tester la différence en valeurs et en types
<	Permet de tester si une valeur est strictement inférieure à une autre
>	Permet de tester si une valeur est strictement supérieure à une autre
<=	Permet de tester si une valeur est inférieure ou égale à une autre
>=	Permet de tester si une valeur est supérieure ou égale à une autre

Tableau 2 – Les opérateurs de comparaison

Exercice 3 : reprendre l'exercice avec la conversion euro/dollars et ajouter une condition. Si la somme excède 15 euros, afficher "trop cher" et si elle est en dessous ou égale, afficher "on prend" .

Opérateur	Opérateur logique
&&	et
||	ou
!	non

Tableau 3 – Les opérateurs logiques

6.3) While et le for

La boucle while s'écrit ainsi en Javascript :

```
while (condition) {
        instruction_1;
        instruction_2;
}
```

Et la boucle for s'écrit ainsi en Javascript :

Modèle

```
for (initialisation; condition; incrémentation) {
        instruction_1;
        instruction_2;
}
```

Exemple

```
for (var iter = 0; iter < 5; iter++) {
        alert('Itération n°' + iter);
}
```

Exercice 4 : Écrivez une boucle while qui se répète 10 fois. Vous ferez le même exercice avec une boucle for.

7) Les fonctions

La définition d'une fonction est introduite par le mot clé **function** suivi de ses arguments, puis du code de la fonction entre accolades Le résultat de la fonction est indiqué par le mot-clé return. Exemple :

Modèle

```
        function maFonction(arguments) {
        // Le code à executer
        }
```

Exemple

```
        function carre(i){
        return(i*i);
        }
```

Cette procédure permet de déclarer une fonction, il faut maintenant afficher le résultat.

Dans le fichier.js

```
document.write(carre(3));
```

ou bien dans le fichier HTML

```
<script>
 document.write("Le résultat : "+carre(3)+".");
</script>
```

Exercice 5 :

- Reproduire les deux exemples précédents et observer le résultat.

- Écrire une fonction f en Javascript qui à une valeur x renvoie $x^2 - 3x + 1$. On affichera l'image de 3.