

CREATION WEB DYNAMIQUE

V °) JavaScript

V-1 °) Introduction

JavaScript est un langage interprété et sensible à la casse. C'est aussi un langage appelé « client », c'est-à-dire exécuté chez l'utilisateur lorsque la page Web est chargée. Il a pour but de dynamiser les sites Internet et de créer une interaction entre l'internaute et les sites internet.

Toutefois, il faut savoir que tout le monde n'a pas JavaScript : Il faut donc, toujours que les informations (données) de votre page soient accessible même sans JavaScript.

JavaScript est là pour apporter un plus (ergonomie, dynamisme), mais on doit pouvoir s'en passer.

V-2 °) Notion de base en JavaScript

JavaScript est un langage objet : chaque objet possède des méthodes (ou fonctions), des propriétés et des objets.

Dans une page Web, l'objet le plus élevé dans la hiérarchie est la fenêtre du navigateur : *window*. Cet objet *window* contient entre autres l'objet *document* qui lui même contient tous les objets contenus dans la page Web (paragraphe, formulaires, etc.). En plus de ces objets, il existe des objets créés par l'utilisateur.

Les méthodes sont des fonctions qui permettent d'agir sur certaines propriétés de l'objet, les propriétés contiennent les paramètres d'un objet.

Exemple d'un objet voiture : nous allons lui attribuer

- des propriétés : la couleur, la marque, le numéro d'immatriculation ;
- des méthodes : tourner(), avancer(), reculer(), changer la couleur() ;
- des objets : le moteur, les phares, les pneus.

La Syntaxe :

Tout script est encadré des balises `<script>` `</script>`, on précise également le type grâce à l'attribut `type` :

```
<script type="text/javascript"> ou bien <script language="javascript">
```

```
// le code JavaScript à mettre ici
```

```
</script>
```

Ces balises script sont généralement insérées dans le head de la page, ou entre les balises body. Dans ce dernier cas les scripts sont exécutés au fur et à mesure du chargement de la page.

Il est possible d'insérer du code JavaScript dans les balises HTML. Cependant, il faut que le code inséré soit bref pour des questions de lisibilité (dans le cas des événements).

On peut enregistrer le script dans un fichier indépendant de la page Web. Dans ce cas, on précise dans le *head* le lien vers ce fichier.

```
<script type="text/javascript" src="MonFichier.js"></script>
```

L'avantage est que l'on peut ainsi réutiliser le script pour une autre page.

Il est fortement recommandé de terminer l'ensemble des instructions JavaScript par un point virgule (même si, en effet, ce n'est pas toujours nécessaire).

Une fonction se définit comme suit :

```
function maFonction()  
{  
.....  
}
```

Les instructions conditionnelles (if et else) et les boucles (for ou while) se font de la même manière qu'en php (voir chapitre précédent).

En JavaScript les variables ne sont pas typées. Il faut néanmoins les déclarer à l'aide du mot clef *var*
Exemple :

```
var maVariable = Une valeur ;
```

Les objets internes JavaScript

Nous avons déjà signalé que JavaScript est sensible à la casse, il est donc important de noter que :

Les objets internes JavaScript commencent par une majuscule : String, Math, Array, Boolean, Date, Number, Function (à ne pas confondre avec le mot-clef function), etc.

Les méthodes ainsi que les propriétés d'un objet commencent par une minuscule.

- exemple de méthodes : toLowerCase() ; getElementById() ; charAt() ; fromCharCode() ; etc.
- exemple de propriétés : id ; type ; innerHTML ; tagName ; style ; etc.

L'accès à une méthode ou à une propriété d'un objet se fait en plaçant un point entre le nom de l'objet et la propriété ou la méthode.

Exemple :

```
var monObjet = document.getElementById("idObjet");  
monObjet.style.display = "none";
```

ou

```
document.getElementById("idObjet").style.display = "none";
```

V-3 °) Programmation événementielle JavaScript

JavaScript est aussi un langage événementiel et tous les événements commencent par *on* : *onclick*, *onload*, *onmouseout*, *onmouseover*... Ils peuvent être insérés dans les balises HTML du document et il est vivement conseillé de les écrire en minuscules.

Exemple :

```
<body onload="maFonction()">  
<input type="button" onclick="maFonction()">
```

NB : maFonction() est préalablement créée dans le script mis dans la balise head de votre page ou appeler depuis un fichier externe toujours dans la balise head

Voici quelques événements courants :

```
onload ----->>> après le chargement de la page  
onunload ----->>> lors de la fermeture de la page  
onbeforeunload ----->>> juste avant la fermeture de la fenêtre  
onclick ----->>> lors d'un clic  
ondblclick ----->>> lors d'un double-clic  
onmousedown ----->>> quand on enfonce le bouton de la souris  
onmouseup ----->>> quand on relâche le bouton de la souris
```

onmousemove ----->>> lorsque la souris se déplace dans un objet du document
onkeydown ----->>> quand on enfonce une touche du clavier
onkeyup ----->>> quand on relâche la touche
onblur ----->>> quand l'élément perd le focus
onfocus ----->>> quand l'élément a le focus
onchange ----->>> quand l'élément perd le focus et que son contenu a changé
onsubmit ----->>> juste avant l'envoi d'un formulaire
onreset ----->>> lors de la réinitialisation du formulaire
onselect ----->>> quand le contenu d'un élément est sélectionné
onscroll ----->>> lors de l'utilisation de la barre de défilement
onbeforeprint ----->>> avant l'impression (après le clic sur le bouton Ok)
onafterprint ----->>> après l'impression
oncopy ----->>> lors du copier vers le presse-papier
onpaste ----->>> lors du coller depuis le presse-papier

Exercice à faire pendant le cours :

Connaitre la position de la souris dans votre page web à tout moment. Cet exercice permet l'utilisation de l'objet *event* (gestionnaire des événements). Nous apprenons comment écrire sur une page avec « innerHTML » et aussi l'événement « onmousemove ».

L'équivalent input box (le prompt) et la méthode writln.

```
<script type="text/JavaScript">
var Prenom=prompt("Qui êtes-vous ? ","Tapez votre nom ici : ")
document.writeln("Bienvenue "+ Prenom)
</script>
```

Comment utiliser les objets et autres propriétés du formulaire.

```
<script type="text/JavaScript">
function DireLongueur(formulaire)
{
var SonTexte = document.formulaire.phrase.value
alert ("Vous avez tapé " + SonTexte + ".\nLa longueur de cette chaîne est de " + SonTexte.length + "
caractère(s)")
}
</script>
<form method="post" action="" name="formulaire">
<input type="text" name="phrase" size="40">
<input type="button" value="Lancer le script" name="Bouton" onClick="DireLongueur(this.form)">
</form>
```

Information sur le navigateur et date du jour

```
<script type="text/JavaScript">
alert("Navigateur : "+navigator.appName+" ; Version : " +navigator.appVersion)

var Aujourdhui = new Date(); // met la date actuelle dans la variable Aujourdhui
document.writeln(Aujourdhui); // écrit le contenu de Aujourd'hui à l'endroit souhaité
LeTitre = document.title;
document.writeln("<br>Le titre de cette page est <b>"+LeTitre+"</b>");
</script>
```

Utilisation des objets du navigateur

L'objet window

Il faut savoir que cet objet représente le navigateur contenant l'objet *document*. Il est créé lors de l'ouverture du navigateur et contient toutes les propriétés et les méthodes de gestion de la fenêtre. Ses propriétés et ses méthodes peuvent être appelées sans devoir préciser l'objet *window*. C'est le cas dans nos exemple précédant ou la méthode `alert()` de cet objet est directement appelé. Le navigateur comprendra que `alert("Bonjour")` et `window.alert("Bonjour")` sont pareils

L'objet `window.document`

L'objet *document* regroupe toutes les méthodes de gestion de la page Web. Ses fonctions permettent de cibler un objet (un paragraphe par exemple) pour modifier ses attributs. Pour modifier les attributs d'un élément (paragraphe, lien, etc...), celui-ci doit être au préalable identifié par un identifiant unique (attribut *id*). L'objet est ensuite ciblé grâce à la méthode `getElementById()` contenue dans l'objet *document*. Si l'élément possédant cet *id* n'existe pas la méthode renvoie *null*.

```
//objet contient toutes les propriétés de idElement
var monElement = document.getElementById("idElement");
//pour modifier la taille de la police
monElement.style.fontSize = "12px";
//pour modifier la police
monElement.style.fontFamily = "Arial";
//pour modifier le contenu du paragraphe (balises div, span, p et body uniquement).
monElement.innerHTML = "Salut tout le monde !!";
```

`innerHTML` est une instruction qui permet de modifier le contenu d'une balise ou d'insérer un objet dans la page.

Exemple d'insertion d'une image :

```
//Ciblage du paragraphe
var MonParagraphe = document.getElementById("idPg");
//Modification de son contenu
MonParagraphe.innerHTML = "<img src='imageInsee.gif' /> Mon nouveau texte";
```

`innerHTML` est une instruction qui permet de modifier le contenu d'une balise ou d'insérer un objet dans la page. Il faudrait que ce paragraphe ait le `id = "idPg"` c'est-à-dire `<p id = "idPg"><p />`

Les étudiants peuvent prendre connaissance d'autres méthodes et propriétés de l'objet *document* dans l'objet javascript de `ScriptEdit` mis a leur possession.

L'objet `window.location`

Cet objet est intéressant de part ses méthodes et propriétés suivantes :

`location.href` , une propriété qui fait référence à l'URL de la page. Elle permet de récupérer ou d'affecter l'URL de la page.

`location.reload()`,une méthode qui permet de recharger la page en cours.

`location.assign()` une méthode qui permet de charger l'URL passée en argument avec inscription dans l'historique.

`location.replace()`une méthode qui permet l'URL passée en argument sans inscription dans l'historique.

DOM et AJAX

Le DOM (Document Object Model) et AJAX (Asynchronous JavaScript and XML) feront l'objet d'un cours apart pour le niveau avancé.

V-4 °) Conclusion

Ces quelques éléments constituent une base à savoir sur l'utilisation de JavaScript dans le Web. Pour plus de détails voir les exemples pendant le cours en classe.