**TP JavaScript**

\_ Les programmes Javascript permettent de rendre dynamique un site Internet développé par HTML.

\_ Le Javascript est un langage de script simplifié orienté objet.

\_ Les scripts écrits en Javascript sont interprétés.

\_ Le JavaScript est écrit directement au sein du document HTML sous forme d'un script encadré par des balises HTML spéciales **(<script></script>)** ou appelé à partir d’un fichier externe (**<script type="text/javascript" src="exemple2.js"></script>**.

<html>

<head>

<title>une page simple </title>

</head>

<body>

Bonjour

**<script language='javascript'>**

**alert('bonjour');**

**document.write (**

**prompt('quel est votre nom ?','Entrez votre nom');**

**confirm(Veuillez choisir un boutton ?');**

**</script>**

</body>

</html>

**Exemple 1**

1. **Exemples introductif**

* Expliquez le résultat de l’exemple 1.
* Entre **confirm**  et **write**?

1. **Règles générales**

La syntaxe du langage Javascript s'appuie sur le modèle de Java et C :

* 1. Chaque commande doit être terminée par un point-virgule ;
  2. Un nombre à virgule est séparé par un point (.) et non par (,)
  3. Le langage Javascript y est **sensible à la casse.**
  4. Il existe deux méthodes permettant d'intégrer des commentaires à vos scripts : (//) au début ou (/\*) debut et (\*/) à la fin.
  5. JavaScript est un langage objet: chaque objet possède des méthodes (ou fonctions), des propriétés et même des objets : **objet window** ; **objet document ou** **objets créés par l'utilisateur**.
  6. Comme en C et en Java : Des points-virgules, Des accolades, Les opérateurs :

**+ - \* / % ++ -- = += -= \*= /= %= == != <><= >= && || ! ?:**

* 1. Comme en Java : Les structure de contrôle, Opérateur + pour la concaténation des chaines de caractères.

**x = 5 + 5;//10; pour x = "5"+"5";// "55"; pour x =5+"5";// "55"; pour x="5"+5;// "55"**

* 1. Variables: Ex. **var x=10**;
  + Pas de type de variable à la déclaration.
  + Les variables non déclarées sont créés à la 1ère affectation.
  + 5 types de variables en interne (**boolean, number, string, fonction, objet**).
  + Opérateur **typeof;** par exemple: typeof (3.14) // Retourne number.
  + Opérateur **===** pour tester la valeur et le type ? EX.**10 == "10"; // true. 10 === "10"; // false**

**3. innerHTML**

<p> un texte </p>

*innerHTML* est une propriété de tout élément HTML qui désigne le contenu qui se trouve entre la balise entrante et la balise fermante.

Dans l'exemple ci-dessous, innerHTML a pour valeur "un texte":

On l'utilise pour lire pour insérer dynamiquement un contenu dans une page.

**4. Utilisation d'innerHTML**

<p id="**pid**"> un texte </p>

Soit l'élément suivant:

On peut obtenir le contenu du paragraphe (dans ce cas "un texte" ) avec le code suivant, soit dans un fichier externe ou dans le code html:

var x = document.getElementById("**pid**");

alert(x.innerHTML);

On peut remplacer dynamiquement le contenu du paragraphe avec ce code:

var x = document.getElementById("**pid**");

x.innerHTML = "un autre texte";

**Exercice 1**:

**<html>**

**<body>**

**<div id="texte" onclick="this.style.color='red'">**

**<h1>Si je clique sur ce texte, il deviendra rouge</h1>**

**</div>**

**</body>**

**</html>**

Réalisez l’exemple suivant:

Quand vous cliquez sur le texte, la couleur et la font changent.

On utilise l'élément div.

Refaire le même exo avec appel à un ficher externe.

**<html>**

**<head>**

**<script type="text/javascript" src="exemple2.js">**

**</script>**

**</head>**

**<body>**

**<input type="button" value = "Cliquer ici" onclick="changeTexte();"/>**

**<div id="texte">**

**Premier texte ..**

**</div>**

**</body>**

**</html>**

**Exercice 2**:

**1.** Faire l’exemple 2 puis le modifier pour changer deux images : image 1 et image 2 à chaque fois on clique sur le bouton.

**2.** On veut changer trois images.

**3.** Si on met la souris dessus (**onmouseover)**, l’image 1 change et devient image2. Si on enlève la souris (**onmouseout)** l’image2 redevient image1.

**exemple2.js**

**function** changeTexte**()** **{**

elem **=** **document.**getElementById**(**"texte"**);**

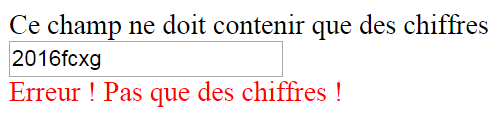
elem**.**innerHTML **=** "Ceci est un nouveau texte ..."**;**

**}**

1. **Les structures de contrôle**

* **if(**x**){...}elseif(**y**){...}else{...}**
* **Switch(**x**){case** 1**:...;break;default:...}**
* **for(**x **=** A**;** x **<**B**;** x**++){...}** et **break;continue;**
* **while(**x**){...}** et **do{...}while(**x**);**
* **try{...}catch(**err**){ alert(**err**.**description**);}**
* **function** name**(**arg1**,** arg2**}{ ...;return** x**;}**

**Exercice3.** Reprendre l’exo 2 et cette fois limitez le nombre de déplacement de la souris en 5 dans ce cas afficher un message: « vous ne pouvez plus changer d’image ! »

**Exercice4.** Nous voulons réaliser l’exemple suivant :

1. Le principe est de contrôler la saisie de l’utilisateur, en vérifiant si les caractères sont numériques ou pas (dans ce cas afficher un message d’erreur).On propose deux solutions possibles :

1. Mettre un bouton « **Vérifier** » pour vérifier tout le champ.

2. Vérifier caractère par caractère.

1. Utiliser cette fois un texte pour le mot de passe : le texte saisi doit être : « pass2018 »,
2. S’il est juste, affichez : Votre mot de passe est correcte et charger une page html,
3. Si le texte est faux : effacer le et afficher : « Votre password est incorrecte, réessayez »
4. Limitez les essais à 5, dans ce cas afficher le message « Vous ne pouvez plus essayez ! » puis fermer la page.

**Exercice Supplémentaire**

1. Vous avez une page avec un formulaire et un champ. Écrivez un script qui va valider le formulaire. Si le champ est vide, un message d'erreur est affiché à côté du champ et le formulaire n'est pas envoyé. Si le champ contient des données, affichez le message: le formulaire peut être envoyé.

|  |  |
| --- | --- |
| **Evènements pour la souris** | |
| **Onclick** | Clic sur l’élément |
| **onbdclick** | Double-clic sur l’élément |
| **onmousedown** | Appui de la souris sur l’élément |
| **onmousemove** | Déplacement de la souris au dessus de l’élément |
| **onmouseover** | Arrivée de la souris sur l’élément |
| **onmouseout** | Sortie de la souris de l’élément |
| **onmouseup** | Relâchement de la souris sur l’élément |
| **Evènements pour le clavier** | |
| **onkeydown** | Appui sur une touche |
| **onkeypressed** | Appui puis relâchement de la touche |
| **onkeyup** | relâchement de la touche |
| **Evènements pour les formulaires** | |
| **onblur** | Un élément de formulaire perd le focus |
| **onchange** | Le contenu d’un élément change |
| **onfocus** | Un élément de formulaire prend le focus |
| **onreset** | Le formulaire est réinitialisé |
| **onselect** | Sélection de texte dans un formulaire |
| **onsubmit** | Soumission du formulaire |