**P211 - Le langage JavaScript**

1. **Introduction :**

**Application Web :**

****

**Client :** s'entend dans une architecture client-serveur.

**Client lourd :** un logiciel qui propose des fonctionnalités complexes avec un traitement autonome. Le client lourd ne dépend du serveur que pour l'échange des données dont il prend généralement en charge l'intégralité du traitement.

**Client léger :** (aussi appelé par un anglicisme thin) désigne parfois des éléments matériels et parfois des éléments logiciels.

**Java Script :**

* extension du HTML : permet d’exécuter les commandes
* interpréter par les navigateurs
* comportements différents selon les navigateurs

ex :

<SCRIPT language="Javascript">

Placez ici le code de votre script

</SCRIPT>

1. **Les fonctionalités des noyau :**

* Affichage d’une boîte de dialogue :

ex :

<HTML>

<HEAD>

<TITLE>Page contenant du Javascript</TITLE>

</HEAD>

<BODY>

<SCRIPT language="Javascript">

**window.alert("Un message d alerte!");**

</SCRIPT>

</BODY>

</HTML>

* Concept de variable :
  + variable = objet pouvant contenir des données qui pourront être modifiées lors d’exécution de programme
  + Nom de variable :
    - doit commencer par une lettre ou par “\_”
    - peut comporter les lettres, des chiffres et “\_” ou “&”
    - pas d’espace
    - ne peut pas utiliser des mots réservés dans le language
    - sensible à la casse
  + peut se faire dans 2 façons :
    - explicitement
      * global si elle définit à l'extérieur d’une fonction
      * non global si au sein d’une fonction
      * ex : var chaîne = “bonjour”
    - implicitement
      * global : La variable sera accessible partout dans le script.
      * ex : chaîne = “bonjour”
* Type de données :
  + pas besoin de déclarer le type d’une variable
  + String :
    - les caractères sont encadré entre des “ ou des ‘
    - caractère de banalisation : \
    - caractères spéciaux :
      * \n : retour à la ligne
      * \r : appui sur la touche ENTRÉE
      * \t : tabulation
      * \b : touche de suppression
    - une seule priorité : length
    - longueur de caractère max : 80
    - syntax :
      * str = (‘machaine’).length ;
      * str = nomchaine.length ;
  + Tableau :
    - type : Array
    - 2 types de tableaux :
      * indicé :
        + ex :

var tab = new Array;

tab[0] = “lundi”;

tab[1] = “mardi”;

* + - * + length pour la taille : nomdeTab.length
      * associatif :
        + ex :

var tab = new Array;

tab[“Lundi”] = “premier jour”;

tab[“Mardi”] = “deuxieme jour”;

* + - * fonctions pour le tableau :
        + tab.length : contient le nombre d’élément d’un tableau
        + tab.push( ) : ajouter des éléments à la fin du tableau
        + tab.unshift( ) : ajouter des éléments au début du tableau
        + tab.shift( ) : supprime le premier élément du tableau
        + tab.pop( ) : supprimer le dernier élément du tableau
        + tab.reverse( ) : inverse l’ordre des éléments du tableau
        + tab.sort( ) : tri les éléments dans l’ordre croissant ou dans l’ordre de l’alphabet croissant
        + tab.slice( ) : retourne une tranche d’un tableau composé des éléments ayant un indice supérieur ou égal à début et strictement inférieur à fin
        + tab.splice( ) : ecraser une tranche d’un tableau à partir de l’indice début et sur un nombre d’éléments
        + tab.concat( ) : retourne la concaténation de 2 tableaux.
        + tab.join( ) : convertir un tableau en chaîne de caractères composés de tous les éléments de tableau séparé par la chaîne séparateur
  + Objet :
    - fonctionnalités :
      * ex :

var obj = { un : 1, deux : 2, trois : 3, verite : true }

test> obj.deux

retourne> 2

test> obj[deux]

retourne> 2

test> obj.[‘tab’] = [true,1];

retourne> var obj = { un : 1, deux : 2, trois : 3, verite : true, tab : [true,1] }

test> “trois” in obj

retourne> true

test> obj.tab[0]

retourne> true

* + Fonction :
    - déclaration :
      * ex :
      * function nomMethode(params) {

//Traitement de la fonction

}

1. **Les opérateurs et structures de contrôle :**

* **Les opérateur de calculs :**

|  |  |
| --- | --- |
| **a+b** | **addition** |
| **a-b** | **soustraction** |
| **a\*b** | **multiplication** |
| **a/b** | **division** |
| **a=b** | **affectation** |

* **Les opérateur d’assignations :**

|  |  |
| --- | --- |
| **a+=b** | **additionne a et b stocker le résultat dans a** |
| **a-=b** | **soustrait a par b, stocker le résultat dans a** |
| **a\*=b** | **multiplier a par b, stocker le résultat dans a** |
| **a/=b** | **diviser a et b, stocker le résultat dans a** |

* **Les opérateur d’incrémentassions :**

|  |  |
| --- | --- |
| **a++** | **a+1** |
| **a--** | **a-1** |

* **Les opérateur de comparaisons :**

|  |  |
| --- | --- |
| **a==b** | **egal** |
| **a!=b** | **différent** |
| **a<b** | **inférieur** |
| **a>b** | **supérieur** |
| **a<=b** | **inférieur égal** |
| **a>=b** | **supérieur égal** |

* **Les opérateur de logique :**

|  |  |
| --- | --- |
| **a&&b** | **et** |
| **a||b** | **ou** |
| **!a** | **négation si a et faut** |

* **Les structures conditionnelles :** 
  + **if...else…**
    - **ex :**

i =100;

if((i >=0) && (i <200)) {

window.alert (i +' est compris entre 0 et 199');

}else if((i >=200) && (i <500)) {

window.alert (i +' est compris entre 200 et 499');

}else{

window.alert (i +' est supérieur à 499');

}

* + switch :
    - ex :

i =1;

switch(i) {

case 0 :

window.alert('La variable i vaut 0');

break;

case 1 :

window.alert('La variable i vaut 1');

break;

default :

window.alert('La variable i est différente de 0 et 1 ');

break;

}

* + **for :** 
    - **ex :**

for (i =0; i <5; i++){

window.alert(i);

}

* + while :
    - ex :

i =0;

while(i <5){

window.alert(i);

i++;

}

1. **JS et HTML :**

* Code JS et Code HTML
  + Clic sur un lien<a href=“javascript:...”>ici</a>
* <SCRIPT> :
  + Cette balise peut être insérée n’importe où
  + Mais existence debonnes pratiques
    - Un maximum de code à l’intérieur des balises
    - <head> et </head> (pour être sûr que tout le code JS soit chargé avant que l’utilisateur n’interagisse avec le site)
    - Evènements placés n’importe où dans le corps de la page entre les balises <body> et </body>
    - Code JS dans un fichier annexe

<SCRIPT LANGUAGE="Javascript" SRC="url/fichier.js"> </SCRIPT>

Où url/fichier.js correspond au chemin d’accès au fichier contenant le code JS

* A travers des évènement :
  + Syntaxe onEvenement = "Code\_JS";
  + fonctionnalités :

|  |  |
| --- | --- |
| onAbort | en cas d'interruption |
| onBlur | en quittant |
| onChange | après modification réussie |
| onClick | en cliquant |
| onDbClick | en double-cliquant |
| onErreur | en cas d’erreur |
| onFocus | en activant |
| onKeydown | en appuyant sur une touche |
| onKeypress | en maintenant une touche appuyée |
| onKeyup | en relâchant la touche |
| onLoad | en chargeant le document |
| onMousedown | en maintenant la touche de souris appuyé |
| onMousemove | en bougeant la souris |
| onMouseUp | en relâchant la touche souris |
| onMouseout | en quittant élément avec la souris |
| onMouseover | en passant sur l’élément avec la souris |
| onReset | en initialisant le formulaire |
| onSelect | en selectionnant du texte |
| onSubmit | en envoyant le formulaire |
| onUnload | en quittant le fichier |
| javascript : | pour les liens |

* Les objets du navigateur
  + les objets de base
    - navigator : contient des informations sur le navigateur (une seule instance)
    - window : une instance par fenêtre (et frame du document HTML), accès à tous les objets crées par les balises HTML
    - location : informations relatives à l’adresse de la page
    - history : liste de liens qui ont été visités précédemment
    - document : propriétés sur le contenu du document (couleur de l’arrière-plan, titre, etc)
* Accès aux objets (DHML) :
  + Pour accéder à un objet, il faut parcourrir la hiérarchie du navigateur en partant du sommet (window)
  + syntax : window.objet1.objet2.objet3
* L’objet window :
  + alert : permet d’afficher dans une boîte de dialogue le texte passé en paramètre (simple boutton “OK”)
  + comfirm : similaire à alert, mais permet un choix entre “OK” et “Annuler”
  + prompt : permet de récupérer une information provenant de l’utilisateur
  + open :
    - permet l’ouverture d’une nouvelle fenêtre
    - syntaxe : window.open(“URL”, “nom\_fenetre”, “options\_fenêtre”);
  + close : pour fermer une fenêtre à l’aide d’un boutton, d’un hyperlien, etc.
* L’objet document :
  + accès à tous les éléments affichés sur la plage, contrôle des saisies, modification de l’apparence et du contenu
  + location : URL de la plage
  + title : titre de la plage
  + filesize : taille de la page en octets
  + domain : nom de domaine de l’adresse de la page
  + lastModified : date de dernière modification de la page
* Accès aux objets (DOM) :
  + getElementById( ) : retourne un objet HTML à partir de son identifiant
  + getElementsByName( ) : retourne un objet HTML à partir de son nom
  + getElementsByTagName( ) : retourne un tableau d’objets à partir de leur balise
  + write( ) : écrit du texte dans le document HTML
* L’objet forms :
  + Permet d’accéder à tous les formulaires du document
    - window.document.forms[“nomform”];
    - window.document.forms[numform];
  + Les propriétés :
    - length : nb de formulaires dans le document
    - name : nom du formulaire
    - action : l’attribut action du formulaire
    - method : méthode de transmission des données du formulaire
  + Les méthodes :
    - reset( ) et submit( )