MODULE XML - JAVASCRIPT

XML – JSON – JAVASCRIPT – JQUERY - ANGULARJS

Partie 2 Javascript

Objectif du cours

But de javascript

Attention compatibilité entre navigateurs (au moins ie 9/chrome/firefox/edge)

Déclaration de variable

Différents types de déclarations :

var : Variable à scope large

let : Variable à scope limité

const : pseudo constante (objet et tableaux non concernés)

Ex :

```
var x=5;

const PI = 3.14;
```

Différences entre var et let

var possède un scope (une existence) plus large que let.

Ex :

```
if(x==5) {
var test = 3;
}
console.log(test); // affiche 3
```

```
if(x==5){
let test = 3;
}
console.log(test); // ReferenceError
```

Attention scope large ne signifie pas infini ! Ex :

```
var x = 3;
function test() {
console.log(x);  // undefined
var x = 7;
}
```

Types de variable

Booléens : var x = true; (true ou false)

Chaines de caractères : var x = ‘test de chaine’;

Entiers : var x = 7;

Flottants : var x = 3.14;

Tableaux : var x = [2,4];

Objets : var x = { a: ’test’ , b: ’essais ’ };

Conversion

var x = true;

x = ‘test de chaine’;

Conversion de chaine en entier :

var x=‘1’;

var y=‘2’;

console.log(x+y); // 12

console.log((+x)+(+y)); // 3

Opérateurs

Arithmétiques (sur des nombres float ou int) : + - / * %

Relationnels (comparaison) : < <= > >= == !=

Logiques : && || !

Unaires (sur une valeur) : ++ -- + -

Assignation : = += -= *= /=

Concaténation (sur des chaines) : +

Conditionnel ternaire : ?: (Ex : rep = (w < x) ? y : z;)

Boucles et condition

Si : if (condition) { Bloc}

Sinon : if (condition) {Bloc} else { Bloc}

Faire … tant que : do {Bloc} while (condition)

Tant que … faire : while (condition) {Bloc}

Pour : for (début; condition; itération) {Bloc}

Pour … dans : for(var in tableau) {Bloc}

Continuer : continue;

Rompre : break;

Selon : switch (expression) { case val1: Bloc case val2: Bloc default: Bloc}

Les fonctions

Déclaration :

```
function nomFonction(param1,param2,paramX)
{
Bloc
return valeur;
}
```

Appel :

```
nomFonction(tot,tata,titi);
```

Les évènements

onLoad : Au chargement de la page

onClick : Au clic sur un élément

onMouseover : Au survol de la souris

onMouseout : A la sortie du survol

Ex : <body onLoad='fonction()'>

Les objets intégrés

Array : méthodes (push, pop, sort, reverse, concat, slice, etc..)

Boolean : méthodes (toString, valueOf)

Date : méthodes (getDate, getFullYear, getMinutes, setHours, toString, etc…)

Global : méthodes (eval, isNaN, parseInt, parseFloat, etc…)

Math : méthodes (abs, acos, exp, max, min, round, random, etc…)

Number : méthodes (toString, valueOf, isInteger, isNan, etc…)

RegExp : méthodes (exec, test, toString)

String : méthodes (toUpperCase, substr, search, replace, size, concat, charAt, etc…)

Ex déclaration d'objet : var x = new Array (2,3);

Ref méthodes : https://www.w3schools.com/jsref/jsref_obj_array.asp

Debugger ces scripts

Pensez à activer les outils de développements sur votre navigateur (touche F12)

Vos erreurs seront présente dans la console.

Si vous souhaitez afficher quelque chose dans la console, utilisez la fonction console.log()

Ex : console.log(‘Coucou’);

Exercice 2 Calculatrice Javascript (noté)

Réaliser une calculatrice type calculatrice Windows en Javascript HTML

La touche CE efface tout, La touche C efface la saisie en cours, La touche <- efface un caractère en cours de saisie, On ne peut saisir qu'une opération, gestion des nombres décimaux, un nombre ne peut avoir qu'une virgule, Gestion du +/-

Notion non abordée mais nécessaire : document.getElementById("id").innerHTML

En Html :

Challenge + : ajouter les touches scientifiques de la calculatrice scientifique Windows

Temps : 3h30

Have Fun ! Work hard !