Programmation Web

Les bases du JavaScript

Chaker BEN MAHMOUD

Introduction

- Le JavaScript est un langage "de script" simplifié "orienté objet" :
  - Initialement élaboré par Netscape en association avec Sun.
  - Standardisé par un comité spécialisé, l'ECMA (European Computer Manufactures Association).
- JavaScript permet :
  - De rendre dynamique un site internet développé en HTML :
    - Validation de formulaires, calculs, messages,
    - Modification de la page web,
    - Communication avec un serveur directement (AJAX)
  - De développer de véritables applications fonctionnant exclusivement dans le cadre d'Internet.

Caractéristiques principales

- Le JavaScript est :
  - Ecrit directement dans le document HTML
  - Un script encadré par des balises HTML
  - Exécuté chez le client (pas d'appel réseau)
  - Interprété (pas compilé)
- Supporté par la plupart des navigateurs web
- Syntaxe proche du C

HTML et JavaScript

- Deux types d'insertion (comme CSS)
  - Directement dans le fichier HTML
  - Dans un fichier externe et inclus en HTML

- Utilisation de balises spécifiques :
  - <script type="text/javascript">…</script>

Insertion dans une page HTML

- Dans le corps de la page HTML
  - Le code s'exécute lors du chargement de la page

```
<html>
<body>
<script type="text/javascript">
  alert('bonjour');
</script>
</body>
</html>
```

Insertion dans une page HTML

- Dans l'entête de la page
  - Le code s'exécute lors d'un événement venant de l'utilisateur
  - Le code correspondant à cet événement se trouve dans le corps du document.
- En pratique les deux sont similaires sur la plupart des navigateurs

```
<html>
<head>
<script type="text/javascript">
   function f () { alert('Au revoir'); }
</script>
</head>
<body onUnload="f()">
</body>
</html>
```

Insertion dans une page HTML

- A placer dans le **<head>** ou le **<body>**
  - Fichier en format texte
  - Permet de réutiliser les scripts dans plusieurs pages
  - Inconvénient : requête supplémentaire vers le serveur

```
<html>
<head>
  <script type="text/javascript" src="fichier.js"></script>
</head>

<body>
  <input type="button" onclick="popup()" value="Clic">
</body>
</html>
```

Structure d’un script

- Similaire à Java ou C
- Règles générales
  - On peut mettre des espaces n'importe où
  - On sépare les commandes par des point-virgule ";"
  - Les réels sont notés avec un "." et pas une virgule ","
  - Commentaires : //… ou /*…*/
  - // ceci est un commentaire
  - /* ceci est aussi un commentaire */

Variables

- **Déclaration**

```
<script language="JavaScript">
    var  date; // Déclaration sans affectation
    var  compteur = 0; // Déclaration avec affectation
    toto = 'coucou'; // Déclaration implicite par affectation
    var  prem, second; // variables séparées par des virgules
    monNombre = new Number(); //Déclaration typée sans affectation
    e = new Number(2.71828); // Déclaration typée avec affectation
    var  maChaine = new String();  //Déclaration de chaîne
    var  toto = new Boolean(true);  //Déclaration de booléen
</script>
```

Principaux types :

- **Chaînes**
- **Nombre** (entier ou décimaux) : 10^-308 >nombre < 10^308
- 3 valeurs spéciales :
  - **Positive** Infinity ou +Infinity
  - **Negative** Infinity ou –Infinity
  - **NaN** (Not a Number) en général le résultat d'une opération incorrecte
- **Boolean** : true (vrai) et false (faux)

- Type **Undefined** => Une seule valeur, undefined
  - Type d'une variable avant qu'elle soit affectée
- Type **Null** => Une seule valeur, null
  - Signale l'absence de données dans une variable

Les Boucles

- Boucle **for** :
  - **for** (i=0 ; i<5 ; i++)

    { … }
- Boucle **while** :
  - **while** (test)

    { … }
  - **do**

    { … } **while** (test)

Les Conditions

- **if (test) { } else { }**
- **Tests possibles :**
  - **Egalité** : ==, !=
  - **Inférieur**, **supérieur** : =< , >= , > , <
  - **Opérations** bit à bit : & , |
  - **Identique** à : ===, !== (teste valeur et type)
    - ('1' == 1) // true
    - ('1' === 1) // false
  - **Opérations logiques** : &&, ||

Les tableaux

- **Méthode 1 :**
  - **var** table = **new Array ( ) ;**
  - **var** noms = **new Array** ("Pierre", "Paul", "Jacques");
  - **var** best_scores = **new Array** (2.5, 4.7, 3.14);
- **Méthode 2 :**
  - **var** MonTableau = **[ ];** **//// recommandé**
  - **var** tab1 = **[42, 12, 6, 3];**
  - **var** tab2 = **[42, 'Sébastien', 12, 'Laurence'];**
- **Les tableaux associatifs:**
  - **var** MonTab1 = { **"valeur1"** : 10 , **"valeur2"** : 55, **"valeur3"** : 30 };
- **Accès :**
  - tab [1] ou bien MonTab1 [**"valeur1"**]
- **Proprieté length =>** retourne le nombre d'éléments du tableau

  => **tab1.length**

Opérations sur les tableaux

- **push(val)** : ajouter à la fin du tableau

  => tab1.**push** ('Ahmed');

- **unshift(val)** : ajouter au début du tableau

  => tab1.**unshift** (‘salah');

- **shift()** : retirer le premier élément du tableau

  => tab1.**shift**();

- **pop()** : retirer le dernier élément du tableau

  => tab1.**pop**();

- **split ()** : découper une chaîne de caractères en tableau avec

  => var s1 = 'Jérôme Guillaume Paul',

  => tab1 = **s1. split**(' '); *// Avec les espaces, on a trois items*

Opérations sur les tableaux

- **sort()**: permet le classement des éléments du tableau

=> tab1var fruits = ["Banana", "Orange", "Apple", "Mango"];
=> **fruits.sort();** //// Apple,Banana,Mango,Orange

- **slice ( )** : retourne une section du tableau

=> var fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"];
=> var citrus = fruits.slice(1, 3); /// Orange,Lemon

```
for ( var i = 0 ;  i < tab.length ; i++) {
      alert ( tab [i] );
}
```

Entrées/sorties

- Trois types de boites de messages peuvent être affichés en utilisant Javascript
  - **Méthode alert()** : sert à afficher à l'utilisateur des informations simples de type texte. Une fois que ce dernier a lu le message, il doit cliquer sur OK pour faire disparaître la boîte
  - **Méthode confirm()** : permet à l'utilisateur de choisir entre les boutons OK et Annuler.
  - **Méthode prompt()** : La méthode prompt() permet à l'utilisateur de taper son propre message en réponse à la question posée
- La méthode **document.write** permet d 'écrire du code HTML dans la page WEB

Entrées/sorties

```
3  <script type="text/javascript">
4      /// afficher dans le body
5      document.write("Bonjour  à l'ENIG ");
6      /// message de confirmation "true/false"
7      var x1=confirm("test de confirmation");
8      if(x1)
9          alert(x1+" :juste");
10       else
11         alert(x1+" :fault");
12
13     ///// lecture des données de l'utilisateur
14     var x=prompt("Donner votre departement");
15     alert(x);
16 </script>
```

Fonctions

- Valeur de retour non typée
- Arguments non typés

```
function ma_fonction(arguments)
  {
    instructions ;
    return quelque_chose; // ou ne pas faire…
  }

ma_fonction(12) ; // Appel
```

Chaînes : Propriétés & Méthodes

- Méthodes
  - **charAt(*index*)**
  - **charCodeAt(*index*)**
  - **concat(*chaine2, chaine3, …*)**
  - **fromCharCode(*code1, code2, …*)**
  - **indexOf(*aiguille*[, *index*])**
  - **lastIndexOf(*aiguille*[, *index*])**
  - **match(*expr_reg*)**

Chaînes : Exemples

```
var s = "Bon anniversaire Benjamin" ;
s.charAt(2) ;   → n
s.charCodeAt(2) ; → 110
s.concat(" du groupe C12") ;
      → Bon anniversaire Benjamin du groupe C12
s.fromCharCode(49, 50) ; → 12
s.indexOf("Benjamin") ; → 17
s.lastIndexOf("a") ; → 21
s.match(/Benjamin$/) ;
      → Benjamin      (null si non trouvé)
```

Chaînes : Méthodes

- Méthodes
  - **replace**(*expr_reg, nouvelle_chaine*)
  - **search**(*expr_reg*)
  - **slice**(*debut*[, *fin*])
  - **split**(*separateur*[, *limite*])
  - **substr**(*debut*[, *taille*])
  - **substring**(*debut, fin*)
  - **toLowerCase**()
  - **toUpperCase**()
- Opérateurs
  - **+**

Chaînes : Exemples

```
var s = "Bon anniversaire Benjamin" ;
```

```
s.replace(/i/g, 'I'); → Bon annIversaIre BenjamIn
s.search(/n{2}/i) ;   → 5
s.slice(17) ;         → Benjamin
s.split(" ") ;        → Bon,anniversaire,Benjamin
s.substr(4, 12) ;     → anniversaire
s.substring(4, 16) ;  → anniversaire
s.toUpperCase()+s.toLowerCase() ;
   → BON ANNIVERSAIRE BENJAMINbon anniversaire benjamin
```

Objet Math

- Propriétés
  - `E, LN10, LN2, LOG10E, LOG2E, PI, SQRT1_2, SQRT2`
- Méthodes
  - `abs(val)`
  - `acos(val), cos(val), …`
  - `exp(val), log(val)`
  - `floor(val), round(val), ceil(val)`
  - `max(val1, val2), min(val1, val2)`
  - `pow(val, puiss), sqrt(val)`
  - `random() // 0 → 1`

Objet `Math` : Exemples

```
document.write(115.04+15) ;
```

➔ 130.04000000000002 (*Euh ?...*)

```
document.write(Math.PI) ;
```

➔ 3.141592653589793

```
document.write(Math.abs(-12.34)) ;
```

➔ 12.34

```
document.write(Math.floor(12.54)) ;
```

➔ 12

```
document.write(Math.round(12.54)) ;
```

➔ 13

```
document.write(Math.ceil(12.54)) ;
```

➔ 13

```
document.write(Math.random()) ;
```

➔ 0.394555831655689

Propriétés & Fonctions supérieures

- Propriétés
  - **`Infinity, NaN, undefined`**
- Fonctions
  - **`eval(`*`chaine`*`)`**
  - **`isFinite(`*`nombre`*`)`**
  - **`isNaN(`*`objet`*`)`**
  - **`parseFloat(`*`chaine`*`)`**
  - **`parseInt(`*`chaine`*`)`**
  - **`escape(`*`chaine`*`)`**
  - **`unescape(`*`chaine`*`)`**

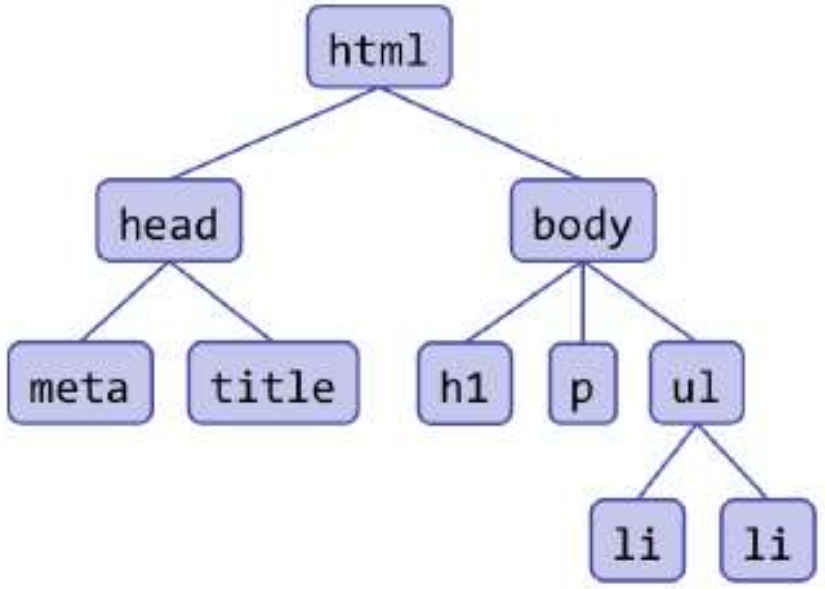
Propriétés & Fonctions supérieures

```
eval("Math.pow(3+2, 2)") ;  → 25
isFinite(Math.log(0)) ;     → false
isNaN("abcd") ;             → true
"12.34"+  2 ;               → 12.342
parseFloat("12.34")+2 ;     → 14.34
escape("Bon anniversaire");
                    → Bon%20anniversaire
unescape("Bon%20anniversaire") ;
                    → Bon anniversaire
```

Javascript et DOM

Document Object Model

- Le DOM (Document Object Model) est une API orientée objet pour la lecture/manipulation des documents html et xml...
- Fournit une représentation structurelle du document et des méthodes pour accéder à son contenu et à sa présentation.
  - Ensemble d'interfaces (classes+propriétés).
  - Types de base (chaines de caractères,entiers, etc).
- La représentation du document est celle d'un arbre.



Interface Node

- Représente chaque nœud de l'arbre

```
.ParentNode //Le noeud parent du noeuf courant
.childNodes[n] //Retourne le n-ieme fils du noeud courant
.firstChild //Le premier fils du noeud courant
.lastChild //Le dernier fils du noeud courant
.nextSibling //Permet de récupérer noeud enfant suivant
//d'un noeud.
.prevSibling //Permet de récupérer noeud enfant precedent
//d'un noeud.
.nodeType //Retourne le type du noeud "TR", "IMG",  etc.
.attributes //Retourne les attributs du noeud
```

Plusieurs sous-interfaces de Node dont les principales sont :

- Document : interface du nœud racine
- Element : interface des nœuds correspondant à des balises
- Attr : interface des nœuds correspondant à des attributs
- Text : interface des nœuds correspondant à des textes

Javascript et DOM

+ Attraper le contenu des différentes balises de la page HTML grâce au méthodes de l'interface Document

`.getElementById("foo")` : récupère l'élément HTML correspondant à la balise d'identifiant "foo"

`.getElementsByTagName("div")` : récupère sous la forme d'un tableau tout les éléments de type div

+ Récupérer et modifier les attributs de balises HTML

`.getAttribute(att)` : récupère l'attribut att de l'élément courant

`.setAttribute(att,nval)`:modifie l'attribut ***att*** de l'élément courant en ***nval***

+ Récupérer et modifier le contenu texte des balise HTML

`.innerHTML` : récupère le contenu HTML de l'élément courant

`.appendChild` : ajoute un élément enfant à l'élément courant

+ Beaucoup d'autres méthodes disséminés dans les interfaces Document, Element, Noeud...etc

Javascript et DOM : Exemple

Supposons que la page HTML contienne la balise paragraphe suivante :

```
<p id="intro"> Ceci est l'ancien paragraphe </p>
```

1. On récupère cet élement par son Id

```
var monElement = document.getElementById("Intro");
```

2. On change la propriété de cet élement

```
monElement.style.fontSize="12pt";
```

Ou

```
monElement.innerHTML="Ceci est mon nouveau paragraphe" ;
```

Les évènements

onclick	Se déclenche lorsqu'un clic est fait sur l'élément
ondblclick	Se déclenche lorsqu'un double clic est fait sur l'élément
onblur	Se déclenche lorsque l'élément perd le focus
onfocus	Se déclenche lorsque l'élément à le focus
onchange	Se produit lorsqu'une modification a lieu sur un champ de saisie
onselect	Se déclenche lorsque l'utilisateur sélectionne du texte sur la page (si l'évènement est associé au <body> ou dans une zone de texte)
onsubmit	Permet de soumettre le formulaire

Les évènements

<table>
<tr><td>onkeydown</td><td rowspan="3">Ces trois évènements constituent ensemble la saisie d'une touche du clavier :
• "onkeydown" correspond à l'action d'appuyer sur une touche
• "onkeypress" correspond à l'action de maintenir enfoncer la touche
• "onkeyup" correspond à l'action de relâcher la touche</td></tr>
<tr><td>onkeypress</td></tr>
<tr><td>onkeyup</td></tr>
<tr><td>onmouseover</td><td rowspan="2">Ces évènements se déclenchent lorsque le curseur de la souris survole l'élément (onmouseover) ou bien quitte l'élément (onmouseout)</td></tr>
<tr><td>onmouseout</td></tr>
<tr><td>onload</td><td>Cet évènement se déclenche lorsque le document a terminé son chargement complet</td></tr>
<tr><td>onunload</td><td>Se déclenche lorsque le navigateur va quitter la page courante</td></tr>
</table>

Les évènements

Il nous semble utile dans cette partie "avancée" de présenter la liste des objets auxquels correspondent des gestionnaires d'événement bien déterminé.

Objets	Gestionnaires d'événement disponibles
Fenêtre	onLoad, onUnload
Lien hypertexte	onClick, onmouseOver, on mouseOut
Elément de texte	onBlur, onChange, onFocus, onSelect
Elément de zone de texte	onBlur, onChange, onFocus, onSelect
Elément bouton	onClick
Case à cocher	onClick
Bouton Radio	onClick
Liste de sélection	onBlur, onChange, onFocus
Bouton Submit	onClick
Bouton Reset	onClick

Exercice


nom
Nom obligatoire
prenom
Prenom Obligatoire
cin
Cin oblig
Valider

Propriétés spécifiques aux éléments de formulaire

Valables pour certains des éléments (pas forcément tous, ex. selected).

- ***value :*** Pour les champs de formulaire uniquement. Contient la valeur *actuelle* du champ. *L'attribut* value, accessible par getAttribute(), quant à lui, contient la valeur par défaut.
- ***name :*** pour les éléments qui ont un attribut name, et en particulier les champs de formulaire, la valeur de l'attribut en question.
- ***disabled :*** un booléen. Permet de désactiver/activer un champ
- ***readOnly :*** un booléen. Permet de placer un champ en lecture seule
- ***checked :*** pour les cases à cocher (CheckBox). Permet de les cocher/décocher
- ***selectedIndex :*** pour des listes de type Select, permet de fixer ou consulter l'élément sélectionné
- ***Selected :*** S'utilise sur un élément option dans une liste. Permet de le sélectionner/déselectionner.

Des Tests

- Pour récupérer ce qui est sélectionné

```
var select = document.getElementById("op");
var op = select.options[select.selectedIndex].value;
```

- Pour tester si une cas est coché

```
var elt= document.getElementById('checkbox2');
if(elt.checked == true) {   }
```

onclick