

# Cours 1

JavaScript

Mihaela.Scuturici@univ-lyon2.fr

# Objectif et Ressources

- Objectif du cours :
  - Connaître les éléments de base du langage JavaScript
- Ressources :
  - <http://www.editeurjavascript.com/cours/> - Cours complet de JavaScript
  - <http://www.toutjavascript.com/reference/index.php> - référence JavaScript
  - <http://www.w3schools.com/> - tutoriaux JavaScript, HTML, CSS, etc. – en anglais
  - Danny Goodman, *JavaScript – Le guide du développeur*, Editions OEM 2001, ISBN 2-7464-0347-1
  - Flanagan, *JavaScript la référence*, O'Reilly France, 2007, ISBN 2841774155

# Organisez votre travail

- Créez sur votre machine un dossier CoursJavaScript
- Chaque semaine créez dans ce dossier un dossier de travail: Cours1, Cours2, ...

# JavaScript - Historique

- Langage de script coté client créé par Brendan Eich pour Netscape
- Décembre 1995 – sortie de JavaScript annoncée par Netscape et Sun
- Août 1996 – JScript développé par Microsoft est intégré dans Internet Explorer 3.0
- Juin 1997 – nouveau standard ECMAScript (document Standard ECMA-262 - <http://www.ecma-international.org/publications/standards/Ecma-262.htm> )

# JavaScript

- Est un langage de programmation interprété
- Est un langage de script qui permet d'écrire des scripts du côté client
- Permet d'animer les pages Web en utilisant DHTML (Dynamic HTML)
- Permet de répondre aux actions de l'utilisateur
- Est un langage de programmation orienté objet

# JavaScript

- Attention : JavaScript ne marche pas pour tous les utilisateurs !
  - s'ils utilisent un navigateur "rare" avec du support DOM (Document Object Model) incomplet ou inhabituel
  - s'ils utilisent un navigateur sur PDA ou sur téléphone mobile qui n'exécute pas JavaScript,
  - s'ils ont désactivé l'exécution de JavaScript par précaution de sécurité
  - s'ils utilisent un lecteur d'écran (speech browser)

# Les outils nécessaires

- Un éditeur simple pour visualiser ou éditer les sources JavaScript
- Un navigateur pour JavaScript coté client
- Modèle de fichier HTML :
  - `<html>`
  - `<head>`
  - `<title>Ma page de test JavaScript</title>`
  - `<script language = "JavaScript">`
  - `//script JavaScript ici`
  - `</script>`
  - `</head>`
  - `<body>`
  - `//code HTML ici`
  - `</body>`
  - `</html>`

# Notions de base du langage JS

- Java Script contient :
  - Des mots clés
    - Fonctions natives : `escape()`, `eval()`, `isFinite()`, `isNaN()`, `parseFloat()`, `parseInt()`, `unescape()`, ...
    - Instructions : `break`, `if`, `else`, `for`, `in`, `return`, `switch`, `while`, `with`
    - Opérateurs : `function`, `new`, `this`, `var`, `+`, `-`, `*`, `/`, ...
  - Des événements : `onclick`, `onload`, etc.
  - Des objets
    - Propriétés
    - Méthodes



# Notions de base du langage JS

- Commentaires :
  - `//` commentaire sur une ligne
  - `/* ... */` commentaire sur plusieurs lignes
- **alert()** – méthode du JavaScript qui affiche un message dans une boîte de dialogue
- Exemple :
  - `<html>`
  - `<head>`
  - `<script language = "javascript">`
  - `alert (" Bienvenue ! ");`
  - `</script>`
  - `</head>`
  - `<body>`
  - `</body>`
  - `</html>`

# Notions de base du langage JS

- **`prompt("unTexte", "valeurParDefaut")`** - méthode du JavaScript qui affiche un message ("*unTexte*") dans une boîte d'invite et fait apparaître une zone d'édition unique permettant à l'utilisateur d'entrer une réponse ("*valeurParDéfaut*" – s'il existe, affiche une valeur par défaut – que l'utilisateur peut modifier)
- Exemple :
- `<html>`
- `<head>`
- `<script language = "javascript">`
- `couleur = prompt("Entrez votre couleur préférée", "Bleu");`
- `</script>`
- `</head>`
- `<body>`
- `</body>`
- `</html>`

# Notions de base du langage JS

- Sortie de base – cas où JavaScript est coté client, dans un fichier HTML chargé par un navigateur :
  - **document.write**("Chaîne de caractères");
- Exemple :
  - <html>
  - <head>
  - <script language = "javascript">
  - **document.write**('Bonjour le monde !');
  - </script>
  - </head>
  - <body>
  - 
  - </body>
  - </html>

# Générer des balises HTML

- Quel est le contenu affiché par cette page Web ?
- `<html>`
- `<head>`
- `</head>`
- `<body>`
  - `<script language = "javascript">`
  - `document.write('<strong>Bonjour le monde !</strong><br />');`
  - `</script>`
- `test`
- `</body>`
- `</html>`

# Exemple

- Quel est le résultat de cette ligne ?
- `document.write(prompt("Entrez votre couleur préférée :", "bleu"));`

# Types de données

- Nombres : 17, 21,5, etc.
- Chaînes de caractères : " Bonjour "
- Booléen : true (vrai) ou false (faux)
- null : mot clé spécial qui représente la valeur nulle

# Conversion de types de données

- Quel est le résultat de l'évaluation de ces expressions ?
- $2+5$
- $2+"5"$
- $2+3+"5"$

# Conversion de types de données

- Conversion de chaînes de caractères en nombres
- Les fonctions :
  - **parseInt("chaîne")** – transforme la chaîne de caractères en entier
  - **parseFloat("chaîne")** - transforme la chaîne de caractères en réel
- Conversion de nombres en chaînes de caractères ?



# Déclaration des variables

- La commande **var** permet de déclarer des variables
  - **var** exemple = " un texte ";
  - document.write(exemple);
- Noms des variables : doivent commencer par une lettre ou par " \_ ". Les caractères suivants peuvent inclure des nombres

# Affectation de valeurs

- `=` attribue la valeur de l'opérande droit à l'opérande gauche
- `+=` ajoute les opérandes gauche et droit et affecte le résultat à l'opérande gauche
- `-=` (soustraction), `*=` (multiplication), `/=` (division)
- `%=` divise l'opérande gauche par l'opérande droit et affecte le reste à l'opérande gauche

# Opérateurs

- Arithmétiques binaires : +, -, \*, /, %
- Arithmétiques unaires : ++ (incrément), -- (décrément), -(négation unaire)
- Exemple :
  - ++x incrémente x d'une unité et renvoie le résultat
  - x++ renvoie la valeur de x et incrémente ensuite

# Exemple

- Quelles sont les valeurs des variables suivantes après exécution ?
- $x = 5;$
- $y = ++x ;$
- $z = x++;$

# Conversion de types de données

- Le type d'une variable peut changer pendant l'exécution d'un script JavaScript
- `var uneValeur = 2;`
- `uneValeur = uneValeur + 2;`
- `uneValeur = uneValeur * 10;`
- `uneValeur = uneValeur + "20";`
- `uneValeur = "Robert";`

# Opérateurs logiques

- && - And (et) logique renvoie la valeur true lorsque les 2 opérandes ont pour valeur true et false dans les autres cas
- || - Or (ou) logique renvoie true lorsque l'un des opérandes a pour valeur true et false lorsque les deux opérandes ont pour valeur false
- ! – Not logique – renvoie la valeur true si l'opérande a pour valeur false et false si l'opérande a pour valeur true (opérateur unaire: il précède l'opérande)

# Opérateurs de comparaison

- `==` true si les 2 opérandes sont égaux
- `!=` true si les 2 opérandes ne sont pas égaux
- `>`, `<`, `>=`, `<=`
- Exemples :
  - `1==1` renvoie true
  - `3<1` renvoie false

# Tableaux

- Un tableau est une variable qui permet de stocker plusieurs données différentes, rangées en mémoire les unes à la suite des autres.
- Exemples :
  - `var totaux_mensuels = new Array(12);`
  - `totaux_mensuels[2] = 2000.30;`
  - `totaux_mensuels[3]='pas de ventes';`
  - `var jours = new Array("Lun", "Mar", "Mer", "Jeu", "Ven", "Sam", "Dim");`

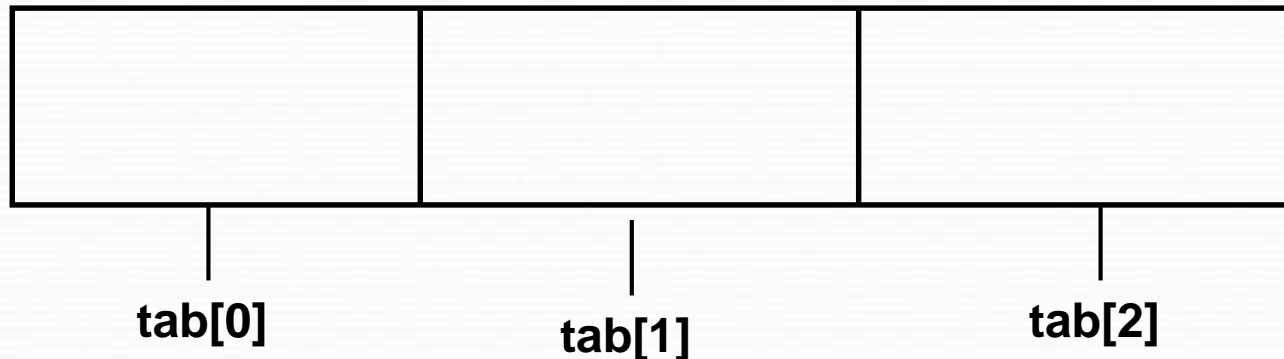


# Tableaux

- `new Array` effectue 2 tâches différentes :
  - Calcule le nombre d'éléments présents dans le tableau et le place dans un espace réservé appelé *length*.
  - Libère suffisamment d'espace de mémoire pour tous les éléments du tableau et associe cet espace à la variable tableau ainsi définie.

# Tableaux unidimensionnels

- Attention : l'indexation des tableaux commence à 0 !
- Les éléments du tableau n'ont pas forcément le même type
- Exemple : `var tab = new Array(3);`



# Tableaux

- Taille d'un tableau :
  - *NomTableau*.length
  - Exemple : tab.length
- Accéder à un élément du tableau :
  - *NomTableau*[index]
  - Exemple : tab[0], tab[2],...

# Tableaux à plusieurs dimensions

- Dans un élément de tableau on peut stocker un autre tableau
- Exemple :
  - `var matrice = new Array(2);`
  - `matrice[0] = new Array(2);`
  - `matrice[1] = new Array(2);`
  - `matrice[0][0] = 1;`
  - `matrice[0][1] = 0;`
  - `matrice[1][0] = 0;`
  - `matrice[1][1] = 1;`

Tab[0][0]	Tab[0][1]
Tab[1][0]	Tab[1][1]

# La structure if ... else

- `if (condition vraie) {  
    instruction;  
}`
- Si la condition est vérifiée, les instructions s'exécutent. Si elle ne l'est pas, les instructions ne s'exécutent pas et le programme passe à la commande suivant l'accolade de fermeture
- `if (condition vraie) {  
    instructions1;  
}  
else {  
    instructions2;  
}`
- Si la condition est vérifiée (true), le bloc d'instructions 1 s'exécute. Si elle ne l'est pas (false), le bloc d'instructions 2 s'exécute.
- Dans le cas où il n'y a qu'une instruction, les accolades sont facultatives.

# Boucles

- while ( *expression* )  
    *instruction*;
- for ( *expression1*; *expression2*; *expression3* )  
    *instruction*;

# while

- Permet de répéter une instruction ou un bloc d'instructions autant de fois qu'on désire, en testant à chaque fois une condition.
- `while ( expression )  
    instruction;`
- Comment ça marche ?
  1. L'*expression* est évaluée
  2. Si elle est non nulle, l'*instruction* s'exécute et retour à 1. Si elle est nulle, l'exécution continue **après** *instruction*.

# while - Exemple

- Afficher les entiers de 1 à 10.

- `<html>`
- `<head>`
- `</head>`
- `<body>`
- `<script language = "JavaScript">`
- `var nPos = 1;`
- `while (nPos <= 10)`
- `{`
- `document.write("nPos = " + nPos + "<br />");`
- `nPos = nPos + 1;`
- `}`
- `</script>`
- `</body>`
- `</html>`



# for

- for (*expression1*; *expression2*; *expression3*)  
    *instruction*;

équivalent à :

- *expression1*;  
  while ( *expression2* ) {  
    *instruction*;  
    *expression3*;  
  }

# for - Example

- Afficher les entiers de 1 à 10.

- `<html>`
- `<head>`
- `</head>`
- `<body>`
- `<script language = "JavaScript">`
- `for(nPos = 1; nPos <=10; nPos++)`
- `document.write("nPos = " + nPos + "<br />");`
- `</script>`
- `</body>`
- `</html>`

# Exemple - for

```
var fruit = new Array(3);
```

```
fruit[0] = 'pomme';
```

```
fruit[1] = 'poire';
```

```
fruit[2] = 'orange';
```

```
for(i=0;i<=2;i++)
```

```
    document.write(fruit[i] + ' ');
```

# Exercice 1

- Affichez la valeur de chacune des expressions suivantes :
- $7+5$
- `"7"+"5"`
- $7==7$
- $7<=7$
- $(7>=5)\&\&(5<5)$
- $(7>=5)|| (5>5)$

# Exercice 2

- Ecrivez un programme en JavaScript qui demande à l'utilisateur une valeur entre 1 et 100 et qui affiche soit la valeur si elle est correcte (entre 1 et 100), soit un message d'erreur si la valeur n'est pas correcte.

# Exercice 3

- Calculez et affichez la factorielle de  $n$  où la valeur de  $n$  est donnée par l'utilisateur :
- $n! = 1 * 2 * \dots * n$
- Exemple :
  - $1! = 1$
  - $2! = 1 * 2 = 2$
  - $3! = 1 * 2 * 3 = 6$