



Partie 1

Objectifs :

Comprendre à quoi sert le JavaScript
Exemples "basiques"



Pré-requis : connaître un langage de programmation



Pour la pratique

- Sur un moteur de recherche, entrez les mots clé suivants:
- Javascript youtube
- En première position, vous aurez la vidéo suivante:
- <https://www.youtube.com/watch?v=wXOxtKLN5tk>
- A visualiser!!

JavaScript est un **langage**

JavaScript n'est pas Java

https://developer.mozilla.org/fr/Guide_JavaScript_1.5/Aper%C3%A7u_de_JavaScript#JavaScript_et_Java

JavaScript et Java ne sont en aucune manière apparentés.
Il s'agit de deux langages différents.

(Historiquement ce langage devait s'appeler LiveScript)


http://fr.wikipedia.org/wiki/Langage_de_script

JavaScript est un langage de script orienté objet qui est exécuté
dans le contexte d'un navigateur internet.

*(Peut aussi être utilisé pour faire des « batch » dans le cadre
d'administration de serveur au même titre que VBScript)*

Exemple N°1

```
<script language="javascript">
alert('Ooops');
</script>
```

Voir l'exécution de
cet exemple 



Attention : Javascript est un langage 'sensible à la casse'

A quoi sert le Javascript ?

Comme il est exécuté « coté client » il peut soulager le serveur de certaines tâches.
En particulier pour les contrôles de saisie dans les formulaires

The screenshot shows a web form titled "1/3 Vos informations personnelles" with a note "* Tous les champs sont obligatoires". The form fields are:

- Civilité*: Radio buttons for M., Mme., and Mlle. (M. is selected). A green checkmark is visible.
- Nom*: Text input containing "DAMM". A green checkmark is visible.
- Prénom*: Text input containing "Lionel". A green checkmark is visible.
- Date de naissance*: Date picker showing "07" for the day and "1990" for the year. A red 'X' is visible.
- Adresse mail*: Empty text input.
- Confirmation de l'e-mail*: Empty text input.
- Mot de passe*: Empty text input.
- Téléphone*: Text input with a placeholder "(ex: 0123456789)".

 A green "Valider" button is at the bottom. A blue starburst callout points to the date field with the text "L'erreur est détectée sans soumission au serveur !".

A quoi sert le Javascript ?

Comme il est exécuté « coté client » il « fluidifie » et facilite « l'expérience utilisateur ».

Exemple d'utilisation :

Affichage d'aide au passage de la souris (popup)

Affichage/Masquage de certains éléments en fonction des actions

Dans certains cas on pourra s'en servir pour proposer aux utilisateurs d'utiliser le « drag and drop »

<http://tool-man.org/ToolManDHTML/sorting.html>

Javascript est-il dangereux ?

NON 

JavaScript a été conçu pour être sans risques pour le visiteur.

JavaScript ne permet pas d'écrire ou de lire un fichier sur le disque dur.
(à l'exception des cookies)


Il est donc impossible de détruire des fichiers indispensables ou d'introduire un virus sur le poste d'un visiteur.

Le Javascript est donc exécuté dans un navigateur internet et est étroitement lié au source HTML

Comment intégrer du JavaScript dans une page HTML ?

3 façons :

Avec la balise HTML : `Cliquer ici`

*Voir l'exécution de
cet exemple* 

Dans les balises scripts : Vu plus haut (premier exemple)

Dans un fichier externe :

`<script type="text/javascript" src="exemple-3.js"></script>`

Contenu du fichier externe (noter l'absence de balise 'script') :

`alert('Je suis dans un autre fichier');` *Voir l'exécution de
cet exemple* 

Exemples

```
<script language="javascript">
premier="Mireille";
nom="Delarue";
age=29;
accueil="Bonjour " + prenom + " " + nom;
alert(accueil);
</script>
```

```
...
document.write('Exemple d\'utilisation de "document.write"');
</script>
```

```
...
alert('Attention au changement de couleur ...');
document.bgColor='lightyellow';document.fgColor='red';
</script>
```


Point abordés dans ces exemples :
"non " déclaration des variables.

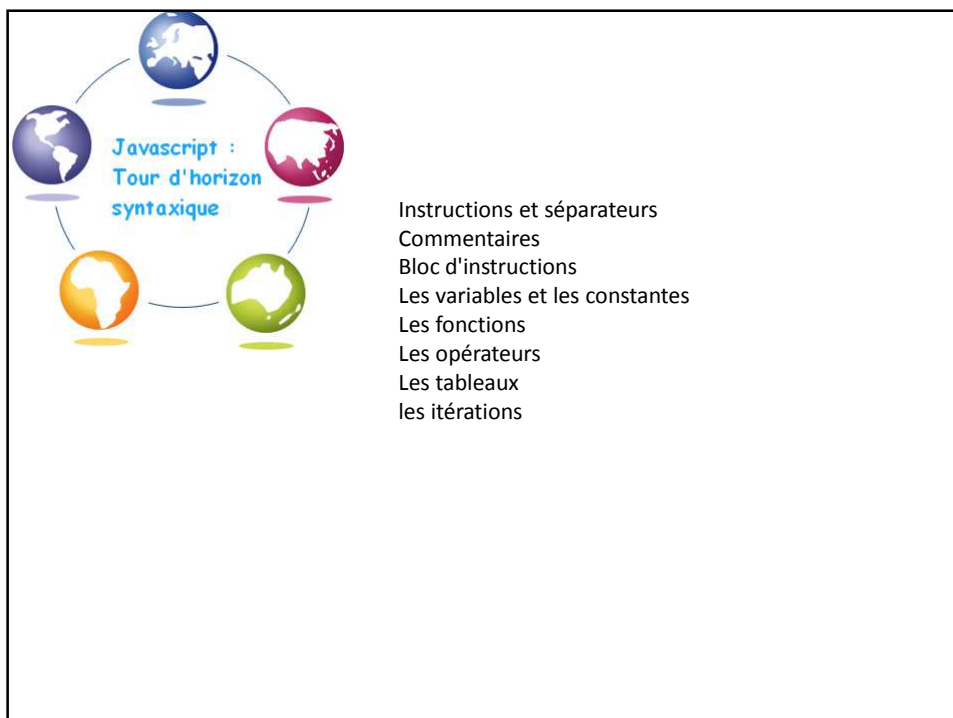
Concaténation avec le signe **+**

« **L'échappement** » de caractères

Boite « **modale** »

« **objet** » : ici **document** avec une méthode et une propriété

Voir l'exécution de
cet exemple 



Instructions et séparateurs

Commentaires

Les variables et les constantes

Les fonctions

Les opérateurs

Les tableaux

les itérations

Les instructions ne sont pas limitées par la fin de ligne (sauf le commentaire simple) et ont pour séparateur le point-virgule.

La virgule est un séparateur interne aux instructions, par exemple dans l'en-tête d'une fonction elle sépare les arguments.

Les crochets [] servent à indiquer un tableau.

Instructions et séparateurs**Commentaires**

Les variables et les constantes

Les fonctions

Les opérateurs

Les tableaux

les itérations



Un commentaire simple débute par une double barre inclinée // et se poursuit jusqu'en fin de ligne.

Un commentaire de plusieurs lignes commence par le couple de symboles /* et se termine par */

Instructions et séparateurs
 Commentaires
Les variables et les constantes
 Les fonctions
 Les opérateurs
 Les tableaux
 les itérations

Les variables sont *dynamiques* en JavaScript, on peut les réaffecter à des valeurs de type différent: nombre, chaîne de caractères, tableau, etc.

```
age=29; alert(age); age='Ooops?'; alert(age);
```

Les noms sont sensibles à la casse :

Age est donc différent de **age**.

Si la variable n'a pas de valeur assignée, son contenu est undefined, ce que l'on peut tester par une instruction de comparaison.

```
var y;  
if(y == undefined) { alert('non defini'); y = 0; }  
if(y != undefined) { alert('defini'); }
```

Instructions et séparateurs
 Commentaires
 Les variables et les constantes
Les fonctions
 Les opérateurs
 Les tableaux
 les itérations

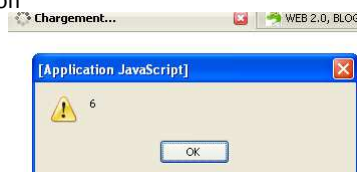
Les fonctions permettent d'isoler, structurer du code.
 Elles permettent également de factoriser du code.

Elles sont définies avec le mot-clé function, suivi du nom, d'une liste d'arguments entre parenthèses et du corps entre crochets:

```
function maFonction(a, b)  
{  
  var x = a + b;  
  return x;  
}
```

A l'intérieur du corps d'une fonction, le mot-clé var est obligatoire si on souhaite que la variable soit locale

// Appel de la fonction
 maFonction(4,2);



Voir <http://www.commentcamarche.net/contents/javascript/jsvar.php3>

Instructions et séparateurs
Commentaires
Les variables et les constantes

Les fonctions (suite)

Les opérateurs
Les tableaux
les itérations

Les paramètres sont passés par valeurs pour toutes les variables mais pas lorsqu'il s'agit d'objets.

Les arguments d'une fonction JavaScript sont tous optionnels : l'interpréteur place les paramètres dans un tableau nommé « arguments ».

```
function maConcatenation(separateur) {
  var resultat = ""; // initialisation de la liste
  // parcourt les paramètres
  for (var i = 1; i < arguments.length; i++)
  {
    resultat += arguments[i] + separateur;
  }
  return resultat;
}

// renvoie "rouge, orange, bleu, "
maConcatenation(" ", "rouge", "orange", "bleu");
// renvoie "éléphant ; girafe ; lion ; singe ; "
maConcatenation(" ; ", "éléphant", "girafe", "lion", "singe");
// renvoie "sauge. basilic. origan. poivre. persil. "
maConcatenation(". ", "sauge", "basilic", "origan", "poivre",
"persil");
```

Voir <http://www.toutjavascript.com/reference/reference.php?iref=189>

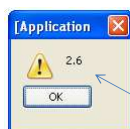
Instructions et séparateurs
Commentaires
Les variables et les constantes

Les fonctions (suite)

Les opérateurs
Les tableaux
les itérations

JavaScript offre aussi des fonctions très pratiques

```
// eval() évalue une chaîne de code JavaScript
document.write(eval("2+2"));document.write("<br />");
eval("x=10;y=20;document.write( (x*y) + '<br />')");
var x=10;document.write(eval(x+17));
```



```
// parseFloat() essaie de renvoyer un nombre en virgule flottante
essai = parseFloat('2.6 euro');
alert(essai);
```

```
// indexOf()
//Syntaxe : Integer chaîne.indexOf(String souschaîne, Integer debut)
// Retourne la position d'une sous-chaîne dans une chaîne, à partir de la
// position debut
// Si la sous-chaîne n'est pas trouvée, indexOf() retourne -1
var str="Hello world!";
document.write(str.indexOf("Hello") + "<br />");
document.write(str.indexOf("World") + "<br />");
document.write(str.indexOf("world"));
```

0
-1
6

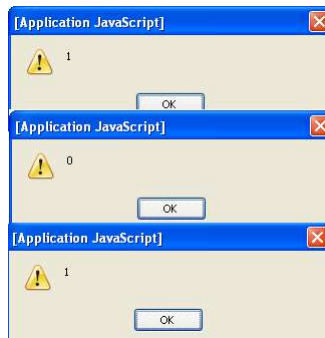
Etc.

Pour en savoir plus : <http://www.w3schools.com/jsref/>

Instructions et séparateurs
 Commentaires
 Les variables et les constantes
 Les fonctions
Les opérateurs
 Les tableaux
 les itérations

Opérateurs arithmétiques

+ addition
 - soustraction
 * multiplication
 / division
 % modulo. Retourne la reste de la division de deux nombres.
 ++ incrémentation, simplifie l'addition de 1.
 -- décrémentement, simplifie la soustraction de 1.



```
<script language="javascript">
i=0;alert(++i);
j=0;alert(j++);
alert(j);
</script>
```

Instructions et séparateurs
 Commentaires
 Les variables et les constantes
 Les fonctions
Les opérateurs (suite)
 Les tableaux
 les itérations

Opérateurs de comparaison

Ce sont des opérateurs qui s'emploient dans les tests de comparaison:

if(x == 5) ...
 == égal
 < inférieur
 > supérieur
 <= inférieur ou égal
 >= supérieur ou égal
 != différent

```
<script language="javascript">
uneVariable=true;
if ( uneVariable ) alert('1 : VRAI'); else alert('1 : FAUX');
uneVariable=false;
if ( ! uneVariable ) alert('2 : FAUX'); else alert('2 : VRAI');
</script>
```



true et false sont des « booleens »
 Le signe ! Signifie « not »

Instructions et séparateurs
 Commentaires
 Les variables et les constantes
 Les fonctions
 Les opérateurs
Les tableaux
 les itérations

Les tableaux permettent de **ranger** un ensemble de « valeurs » dans un même « panier ».
 Pour retrouver les valeurs on utilise un **numéro d'index**.

```
x = [ "a", , , , "e"];
alert(x[0] + "\n" + x[4]);
```



```
x = new Array(3);
alert(x[0] + "\n" + x[4]);
```



```
x[0]='toto';
alert(x[0] + "\n" + x[4]);
```



\n permet d'aller à la ligne

Instructions et séparateurs
 Commentaires
 Les variables et les constantes
 Les fonctions
 Les opérateurs
Les tableaux (suite)
 les itérations

Amusons nous un peu avec les tableaux

```
x = [ "a", , , , "e"];
y = new Array(x,["aa","bb","cc"]);
alert(y[0][0] + "\n" + y[0][4] + "\n" + y[1][2]);
```



```
var a = new Array( new Array("a", "b", "c"), new
Array(4, 5, 6));
document.write(a);
document.write('<br>');
document.write(a[0][0]);
document.write('<br>');
document.write(a[1][0]);
```



document.write permet d'écrire dans le « document HTML »

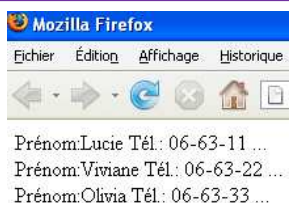
Instructions et séparateurs
 Commentaires
 Les variables et les constantes
 Les fonctions
 Les opérateurs
 Les tableaux
les itérations (ou boucles)

for
 for in
 while
 break et continue
 do while
 switch case

```
prenom=["Lucie", "Viviane", "Olivia"];
telephone=["06-63-11 ...", "06-63-22 ...", "06-63-33 ..."];

var annuaire = new Array( prenom,telephone);

for(var i = 0; i < prenom.length; i++)
  document.write( 'Prénom: ' + annuaire[0][i] + " Tél.: " + annuaire[1][i] + "<br>");
```



...
 Les tableaux
les itérations (suite)


for in (et for each ...)

```
a= new Array();
for( var i = 0; i < 20; i+=4 )    a[i]=i*i;

document.write(a + '<br>');

for(x in a) document.write( a[x] + " - ");
document.write('<br>');
-----for each(x in a) document.write(x + " * " + a[x]);
```

La structure **for in** assigne les indices du tableau à une variable et l'on utilise celle-ci pour indexer le tableau

Plus directe la structure **for each** assigne le contenu du tableau à la variable mais ne fonctionne pas avec IE 

(Cette structure a été ajoutée à JavaScript 1.6 et IE est en 1.5)



0,,,16,,,64,,,144,,,256
 0 - 16 - 64 - 144 - 256 -
 0 * 16 * 64 * 144 * 256 *


Pour aller plus loin http://fr.wikibooks.org/wiki/Programmation_JavaScript
 rubrique « Structures de contrôle »

Questions



Exercice 1

Coder en javascript un programme qui demande une phrase et qui affiche le nombre de lettres 'e' trouvées

*Voir l'exécution de
la solution* 



Exercice 2

A quoi sert le programme suivant ?

```
saisie = prompt('Entrer une phrase' , 'engage le jeu que je le gagne');
var tableau = new Array();

for(var i = 0; i < saisie.length; i++)
    if (saisie.charAt(i) != ' ') tableau[i] = saisie.charAt(i)
saisie = tableau.join("");
saisie2=tableau.reverse().join("");
if (saisie2 == saisie ) document.write('Ok !!' + '<br>');
else document.write('... KO' + '<br>');
```

Ce qu'il faut savoir avant de faire l'exercice pratique N° 3

```
saisie_du_visiteur = prompt('Vous êtes superstitieux ? ( O/N )' , 'Non, ça porte malheur');
alert(saisie_du_visiteur);
```

```
var result = parseInt( prompt('Entrez un nombre entre 1 et 100') );
while( isNaN(result) || result < 0 || result > 100 )
{
    alert('Saisie incorrecte');
    result = parseInt( prompt('Entrez un nombre entre 1 et 100') );
}
```

```
for( var i=1;i<50;i++) // Ecrire des nombres entre 0 et 10
    document.write( Math.round( Math.random() * 10 ) + ' - ');
```

```
if( confirm('Ok ou annuler') )
    alert('Vous avez cliqué sur ok !');
else
    alert('Vous avez cliqué sur annuler !');
```




Exercice 3

Coder en JavaScript un programme qui choisit un nombre secret entre 0 et 100

A charge de l'utilisateur de trouver le nombre secret en faisant des propositions.

Le programme lui répond alors « plus haut » ou « plus bas »

Lorsque l'utilisateur a gagné le programme affiche le message
« Gagné en *n* essai(s) »

Voir l'exécution de
la solution 

Plus loin :

Proposer de faire une nouvelle partie et conserver le meilleur score

Avant ou après les exercices, en fonction de votre état d'avancement et/ou de compréhension du cours, consulter très attentivement <http://www.w3schools.com/js/default.asp>

Aller faire un tour sur :

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>