Les bases de Javascript

- Généralités
- Les types de données
- Les variables
- Les opérateurs
- Les structures de contrôle
- Fonctions
- Objets
- Tableaux

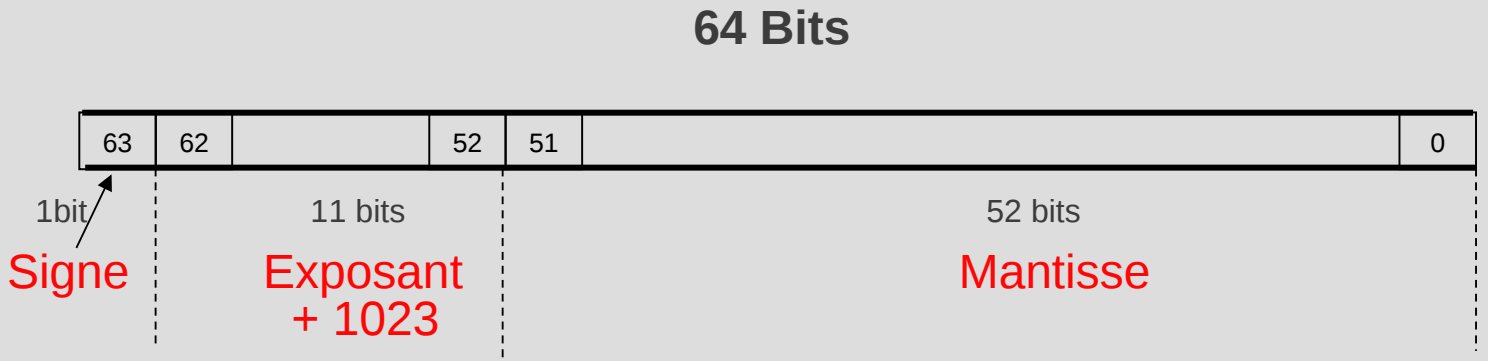
Syntaxe

- Majuscules et minuscules ne sont pas équivalentes
- Séparateur d'instructions ***;*** ou ***retour à la ligne***
- Commentaires ***/* ...*/*** ou ***//....***
- Nom de variable, fonction, objet...
  - Premier caractère : *lettre* ou *_ ou $*
  - Pas de mots réservés : *function , new, if, for, var...*

Les types de données primaires

- Nombre (number)
  - Codage en virgule flottante (IEE754) sur 64 bits
  - *NaN , Infinity*
- Texte (string)
  - Chaînes de caractères Unicode (16bits)
  - Délimiteurs : " ou '
  - Echappement \
- Valeurs booléennes (boolean)
  - Vrai (*true*)
  - Faux *(false*)
- Types *null* et *undefined*

Codage des nombres IEE 754



1 ≤ Mantisse normalisée <2 (1 bit non codé)

Précision : 2^{-53} (mantisse normalisée)

Les plus proches de 0: +/- 2^{-1076} environ 5.10^{-324} (dénormalisé)

Max/Min : +/- 2^{1024} environ $1,8.10^{308}$

Infinity E 1111..1 avec la mantisse nulle

NaN E 1111..1 avec la mantisse non nulle

Nombre et chaines (à tester)

```
document.write(1+2) ;
document.write(0.1+0.2) ;
document.write('1+2') ;
document.write('1'+'2') ;
document.write(1+'2') ;
document.write(3/0) ;
document.write('0/0  = ' + 0/0) ;
document.write(Number.MAX_VALUE) ;
```

Echappement \ , guillemets " et '

```
document.write('J \'utilise \\ pour échapper l\'apostrophe ') ;
document.write("J 'utilise  \\ pour échapper les \"guillemets\" ");

document.write(' \u00A9 Unicode du copyright \\u00A9  ') ;

document.write('<a href= "http://metiersduweb.free.fr">Lien</a>') ;

window.alert( 'Texte \n sur \n plusieurs \n lignes') ;
window.alert( 'Texte \t avec \t des  \t tabulations') ;
```

Les types primaires comme pseudo-objets

```
window.alert( 'Mon texte'.length) ;
window.alert( 'Mon texte'.toUpperCase()) ;
window.alert( (18).toString(2)) ;
window.alert( (18).toString(16)) ;
```

Les types de données composites
objets

- Object

  Function : objet associé à du code exécutable

  Array : collection ordonnée de données primaires ou d'objets

  Objet : collection de données, d'objets et de méthodes

- Objets prédéfinis spécialisés

  Date

  Error

  RegExp

Les variables

- *Nom* associé à une *valeur*

```
var   x = 1.25e3 ;
var   message = "Hello world " ;
var   doubleX = 2*x ;
```

- Les variables ne sont pas typées, exemple :

```
var x ;                  // le type de x n'est pas défini
alert(typeof(x)) ;  // pour afficher le type
x= 7 ;                   //  x devient un nombre
alert(typeof(x)) ;
x =  'Il y a ' + x + 'jours ' ; //  x est maintenant une chaine !
alert(typeof(x)) ;
```

Variable globale, variable locale

- *<script>*

```
var x = "globale" ;
function affiche(){
    var x = "locale";
    alert(x) ;
}
affiche() ;
alert(x) ;
```

</script>

- Supprimer les « var » et réessayer...

Affectation par valeur, affectation par adresse

- *Par valeur pour les types primaires*

```
var  a = 1 ;
var  b = a ;  // la valeur est copiée
a = 100 ;
alert(b) ;
```

- *Par référence pour les types composés*

```
var a = [ 1 , 2, 3, 4] ;
var  b = a ;   // l'adresse est copiée
a[0] = 100 ;
alert(b) ;
```

Opérateurs *(à tester sur help.dottoro.com)*

- *Arithmétiques*

  + , - , * , / , % , ++ , --
- *Comparaison*
- == , === , != , !== , < , <= , > , >= ,
- *Affectation*

  = , += , -= , *= , /= , %= , ...
- *Logiques binaire (sur entiers)* *sur expressions booléennes*

  & , | , ^ , ~ , >> ,<< , >>> &&, || , !
- *Autres opérateurs*

  new , in, ( ) , . ,[ ] , ? : , typeof ,instanceof , delete

Opérateurs *(à tester sur help.dottoro.com)*

- *Arithmétiques*

  + , - , * , / , % , ++ , --
- *Comparaison*
- == , === , != , !== , < , <= , > , >= ,
- *Affectation*

  = , += , -= , *= , /= , %= , ...
- *Logiques binaire (sur entiers)* *sur expressions booléennes*

  & , | , ^ , ~ , >> ,<< , >>> &&, || , !
- *Autres opérateurs*

  new , in, ( ) , . ,[ ] , ? : , typeof ,instanceof , delete

Les structures de contrôle

- Les boucles
  - Pour répéter un bloc d'instructions
  - Quatre sortes de boucles
- Les instructions conditionnelles
  - Le choix simple
  - Alternative
  - Les choix multiples

Boucle (1)

```
do
   {instruction1;
    ...
     }
while (expression);
```

FAIRE la séquence d'instructions TANT QUE l'expression est vraie

- Exemple :

```
var i = 0 ;
do{
      document.write( i + ' , ') ;
       i++ ;
} while (i<5) ;
```

Boucle (2)

```
while(expression)
  {instruction1;
    …
   }
```

TANT QUE l'expression est vraie faire la séquence d'instructions.

- Exemple :

```
var i = 0 ;
while (i<5) {
      document.write( i + ' , ') ;
      i++ ;
}
```

Boucle (3)

```
for (variable= valeur initiale; test ; modification de la variable)
    {instruction1;
     …
     }
```

Les instructions sont exécutées tant que le test est vrai.

- Exemple :

```
for (i = 0 ;  i<5 ; i++){
      document.write( i +' , ') ;
}
```

Boucle (4)

```
for (variable in objet)
    {instruction1;
     …
     }
```

La variable parcoure les propriétés de l'objet .

- Exemple :

```
var i ;
for (  i   in   window.navigator){
          document.write( i  + ' = ' + window.navigator[i] );
          document.write('<br>') ;
}
```

Le choix simple

```
if (expression)
    {instructions;
      …
     }
```

Les instructions sont exécutées si l'expression est vraie.

Exemple :

```
var d= new Date() ;
if (d.getHours()< 17){
     window.alert('Bonjour') ;
}
```

L'alternative

```
if (expression) {
    instructions1;
}
else  {
    instructions2;
}
```

Si l'expression est vraie les instructions1 sont exécutées sinon ce sont les instructions2

- Exemple :

```
var d= new Date() ;
if (d.getHours()< 17) {
    window.alert('Bonjour') ;
}
else {
    window.alert('Bonsoir') ;
}
```

Choix multiples

```
switch(variable) {
    case valeur1   : instruction1(s); break;
    case valeur2   : instruction2(s) ;break;
    case valeur3   : instruction3(s); break;
    default : instruction4s; break;
}
```

L'(es) instruction(s) correspondant à la valeur de la variable est (sont) exécutée(s).

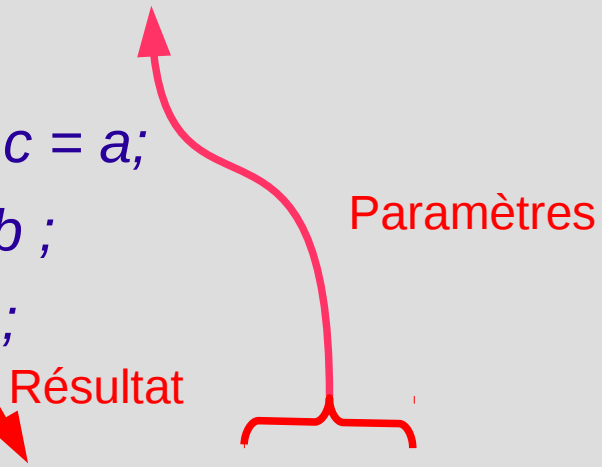
La ligne default est facultative.

Les fonctions

```
function nomDeLaFonction(argument1, argument2, ...) {
        liste d'instructions ;
        }
```

Exemple :

```
        function  plusGrand(a,b) {
                var c ;
                if (a>b)  c = a;
                else  c=b ;
                return c ;
        }
        window.alert('plusGrand(3 , 10)') ; //pour tester la fonction
```



Travail à faire et à mettre en ligne

- Jeu : Faire deviner un nombre choisi par l'ordinateur.

  Utiliser *Math.random()* et *Math.round(...)* pour obtenir un nombre aléatoire compris entre *0* et *100*.

  Faire une boucle qui demande à l'utilisateur un nombre entre 0 et 100 et qui affiche « trop grand » , « trop petit » ou « vous avez trouvé en ...x.. essais

  Utiliser prompt(..) pour la saisie

  Un bouton doit permettre de démarrer un nouveau jeu

  Donner la réponse si la bonne réponse n'a pas été trouvée en 5 essais.

- (en option) un bouton pour abandonner, le nombre à trouver doit alors être affiché.

Travail à faire et à mettre en ligne

- Construire une application WEB d'aide à l'apprentissage des tables de multiplications proposant à l'utilisateur:
  - 1) l'affichage d'une table à choisir de 1 à 9
  - 2) Un mode entraînement
  - 3) Une évaluation notée de 10 questions avec des nombres choisis au hasard (+1 pour les bonnes réponses et -1 pour les mauvaises)
- Personnaliser la page WEB (facultatif)

Travail à faire et à mettre en ligne

- Etudier et mettre en œuvre les balises pour construire des tableaux:
  - <table> < /table><th></th><tr></tr> <td></td> …
- Utiliser ces balises pour mettre en page une facture avec 3 produits différents sur 5 colonnes :
  - Utiliser du code Javascript pour automatiser les calculs
  - Prix unitaire HT, quantité,TVA(19.6%),Prix TTC, Total TTC