

# Javascript : les objets natifs

## Contenu

L'objet String.....	3
L'objet Math .....	4
L'objet Date .....	5
Testez l'exemple ci-dessous .....	5
L'objet Array .....	7
L'objet Image .....	8
Exercice.....	9

Les objets sont des composants essentiels de JavaScript.

Ils sont constitués de propriétés (variables) et de méthodes (fonctions).

JavaScript possède ses propres objets, dits « natifs ».

## L'objet String

L'objet **String** est très utile à la manipulation des caractères.

On crée un objet String avec la syntaxe suivante :

```
var str = new String("toto");  
var str = "toto"; // toute chaîne de caractère est un objet String.
```

La position du premier caractère d'une chaîne est 0.

Instruction	Description
length	Propriété de type entier qui indique la longueur de la chaîne de caractères.
charAt(int) charAt(string, int)	Méthode qui permet d'accéder à un caractère isolé d'une chaîne.
indexOf(string) indexOf(string, int)	Méthode qui retourne la position d'une chaîne partielle à partir d'une position déterminée. (position 0 par défaut)
lastIndexOf(string) lastIndexOf(string, int)	Méthode qui retourne la position d'une chaîne partielle en partant d'une position déterminée puis en reculant. (position par défaut égale à <i>length</i> moins 1).
match(regExp)	Méthode qui retourne un tableau si l'expression régulière est présente dans la chaîne de caractères, <i>null</i> sinon.
replace(regExp,string)	Méthode qui remplace l'expression régulière trouvée par le texte <i>string</i> .
substring(x,y)	Méthode qui retourne une chaîne partielle située entre la position x et la position (y-1).
toLowerCase()	Méthode qui transforme toutes les lettres en minuscules.
toUpperCase()	Méthode qui transforme toutes les lettres en majuscules.

## L'objet Math

L'objet Math sert à manipuler les nombres.

On utilise les Méthodes et propriétés de l'objet Math directement sans créer d'instance (new).

```
var maxVal = Math.max(10,20);
```

Instruction	Description
abs(x)	Méthode qui retourne la valeur absolue de x.
ceil(x)	Méthode qui retourne l'entier supérieur ou égal à x.
floor(x)	Méthode qui retourne l'entier inférieur ou égal à x.
round(x)	Méthode qui arrondit le nombre x à l'entier le plus proche.
max(x, y)	Méthode qui retourne le plus grand des 2 nombres x et y.
min(x, y)	Méthode qui retourne le plus petit des 2 nombres x et y.
pow(x,y)	Méthode qui calcule la valeur du nombre x à la puissance y.
sqrt(x)	Méthode qui calcule la racine carrée du nombre x.
random()	Méthode qui retourne la valeur d'un nombre aléatoire compris entre 0 et 1.

L'objet Math possède également des méthodes trigonométriques et logarithmiques.

JavaScript possède également d'autres fonctions utiles pour les nombres :

Instruction	Description
eval(string)	Méthode qui évalue la chaîne passée en argument comme un script JavaScript. Exemples : <i>var total = eval("10 + 20");</i> <i>eval("x=10;y=20;document.write(x*y)");</i>
isNaN(x)	Méthode qui retourne <i>true</i> si le paramètre x n'est pas un nombre.
parseFloat(string)	Méthode qui convertit la chaîne en nombre à virgule flottante. <i>Si la chaîne ne contient pas que des nombres, JavaScript ne prendra en compte que les nombres en début de chaîne.</i> <i>Si le premier caractère n'est pas un caractère admis, la méthode renverra "NaN" (Not A Number).</i>
parseInt(string)	Méthode qui convertit la chaîne en entier.

## L'objet Date

L'objet Date sert à manipuler les dates.

La création d'un objet Date peut se faire de plein de façons différentes :

```
var myDate = new Date() ;      // Date du jour
var myDate = new Date(millisecondes) ;    // Date depuis le 1er janvier 1970
var myDate = new Date(annee, mois, jour) ;
var myDate = new Date(annee, mois, jour, heure, minute, seconde);
var myDate = new Date(chaîne de caractère représentant une date);
```

Instruction	Description
getFullYear()	Méthode qui retourne les 2 derniers chiffres de l'année.
getMonth()	Méthode qui retourne le mois compris entre 0 et 11 !
getDate()	Méthode qui retourne le jour du mois compris entre 1 et 31.
getDay()	Méthode qui retourne le jour de la semaine compris entre 0 et 6. (0 pour dimanche)
getHours()	Méthode qui retourne les heures comprises entre 0 et 23.
getMinutes()	Méthode qui retourne les minutes comprises entre 0 et 59.
getSeconds()	Méthode qui retourne les secondes comprises entre 0 et 59.
getTime()	Méthode qui retourne l'heure courante sous forme d'un entier représentant le nombre de millisecondes écoulées depuis le 1 <sup>er</sup> janvier 1970 00:00:00.
getTimezoneOffset()	Méthode qui retourne la différence entre l'heure locale et l'heure GMT (Greenwich, UK Mean Time) sous forme d'un entier représentant le nombre de minutes (et pas en heures).
toLocaleString()	Méthode qui retourne la date sous forme de chaîne de caractères.
parse(Date)	Méthode statique qui retourne la date passée en paramètre en millisecondes depuis le 1 <sup>er</sup> janvier 1970 00:00:00.

L'objet Date possède l'équivalent des méthodes *getXXX()* en méthodes *setXXX(x)* pour assigner une nouvelle valeur x à la date. Exemple : *setMonth(10)*.

### Testez l'exemple ci-dessous

```
<!DOCTYPE html>
<html>
<head>
<title>Objet Date - Exemples</title>
```

```
</head>
<body>
<script>
var myDate1 = new Date() ;    // Date du jour
var myDate2 = new Date(3600*24*365*40) ; // Date depuis le 1er janvier 1970
var myDate3 = new Date(2012, 9, 19) ;
var myDate4 = new Date(2012, 9, 19, 10, 33, 12);
var myDate5 = new Date("Jan 1, 2000 00:00:00");

console.log('new Date() : ' + myDate1.toLocaleString());
console.log('new Date(3600*24*365*40) : ' + myDate2.toLocaleString());
console.log('new Date(2012, 9, 19) : ' + myDate3.toLocaleString());
console.log('new Date(2012, 9, 19, 10, 33, 12) : ' + my-
Date4.toLocaleString());
console.log('new Date("Jan 1, 2000 00:00:00") : ' + my-
Date5.toLocaleString());
console.log("");
var annee = 1900 + myDate1.getYear();
var mois  = myDate1.getMonth() + 1;
var jour  = myDate1.getDate();
console.log('date du jour : ' + jour + "/" + mois + "/" + annee);
</script>
</body>
</html>
```

## L'objet Array

L'objet Array (ou tableaux) est une liste d'éléments indexés dans lesquels on pourra ranger (écrire) des données ou aller reprendre ces données (lire).

### L'indexation commence à 0.

Pour faire un tableau, il faut créer la structure du tableau en premier. Les éléments du tableau sont vides. Ensuite, affecter des valeurs dans les cases ainsi définies.

```
var tab = new Array (x); // où x est le nombre d'éléments du tableau.  
tab[i] = "élément";
```

On peut également créer et affecter un tableau en une seule étape :

```
var tab = new Array (10, 5, 4, 20);
```

JavaScript étant un langage peu typé, il n'est pas nécessaire de déclarer le nombre d'éléments du tableau (soit x). JavaScript prendra comme nombre d'éléments, le nombre i le plus élevé lors de "l'alimentation" de la structure (en fait i + 1). Ainsi la formulation suivante serait aussi correcte pour un tableau à 3 éléments :

```
var tCarnet = new Array();  
//ou  
var tCarnet = [];  
  
tCarnet[2] = "Philippe";
```

Instruction	Description
length	Donnée membre qui retourne le nombre d'éléments du tableau.
join() join(string)	Méthode qui regroupe tous les éléments du tableau dans une seule chaîne. Les différents éléments sont séparés par un caractère séparateur spécifié en argument. Par défaut, ce séparateur est une virgule.
reverse()	Méthode qui inverse l'ordre des éléments (ne les trie pas).
sort()	Méthode qui trie les éléments par ordre alphabétique (à condition qu'ils soient de même nature)
push()	Méthode qui ajoute un nouvel élément à la fin du tableau.
pop()	Méthode qui supprime le dernier élément du tableau.
unshift()	Méthode qui ajoute un nouvel élément au début du tableau.
shift()	Méthode qui supprime le premier élément du tableau.
slice(deb, fin)	Méthode qui retourne la partie du tableau commençant à l'indice <i>deb</i> et se terminant à l'indice <i>fin-1</i> .

## L'objet Image

L'objet image permet avec JavaScript de pré charger des images qui pourront être affichées ultérieurement. Ce procédé est utile notamment pour améliorer les effets visuels.

Ces images sont mises en mémoire au chargement de la page.

Une fois chargée, une image est affichée plus rapidement. En revanche, le chargement de la page prend plus de temps.

Pour pré-charger une image, il faut créer un objet image en JavaScript :

```
var img1 = new Image();  
img1.src = "images/logo2.gif";
```

Pour afficher l'image, il faut passer par une balise `<img>`, puis lui affecter l'objet Image pré chargé :

```

```

**Ne pas confondre l'objet image avec la balise `<img>`.**

Autres propriétés de l'objet Image : `name`, `height`, `width`, `hspace` et `vspace`.



## Exercice

Ecrivez un programme permettant de saisir différentes valeurs numérique (avec l'instruction `window.prompt`), ces valeurs seront rangées dans un tableau. La saisie s'arrête quand l'utilisateur entre la valeur 0.

A la fin de la saisie, votre programme doit afficher le nombre de valeurs saisies, la somme et la moyenne.