

**Plan**

1.	<i>Notion d'objets</i> .....	2
2.	<i>String</i> .....	2
3.	<i>Math</i> .....	4
4.	<i>Array</i> .....	4
5.	<i>Date</i> .....	7
6.	<i>Le type Object</i> .....	8



# Objets prédéfinis

## 1. Notion d'objets

Un objet est une entité caractérisée par :

- Des propriétés : des valeurs qui le caractérise
- Des méthodes : des fonctions qu'il peut accomplir

La syntaxe pour utiliser une propriété ou une méthode d'un objet :

Nom\_objet.Nom\_propriété ou Nom\_objet.Nom\_méthode()

Pour créer une variable de type objet, utiliser le mot clé **new** :

### Exemples

```
var d = new Date();
var T = new Array(4);
var c = String.fromCharCode(65); //c='A'
```

Javascript dispose d'un ensemble d'objets prédéfinis que nous allons découvrir tout au long de ce cours.

## 2. String

On utilise les quotes simples (apostrophes) ou doubles (guillemets) pour définir des valeurs chaînes de caractères. La propriété **length** permet de déterminer la longueur d'une chaîne.

### Exemples

```
var nom = "Ali";
var prenom = 'Mohamed';
alert(prenom.length); // 7
var message = "toto a dit \"je suis malade\".";
```

Il est possible de transtyper une valeur en chaîne avec **String()** ou de manipuler une variable déclarée en tant que type String.

### Exemples

```
var v1 = 10; alert(String(v1)); // "10"
var v2 = true; alert(String(v2)); // "true"
var v3 = null; alert(String(v3)); // "null"
var v4; alert(String(v4)); // "undefined"
```

Il existe de nombreuses fonctions (méthodes) sur les chaînes de caractères :

Méthode	Explication
charAt()	renvoie le caractère se trouvant à la position i (en partant de l'indice 0). <code>var str = "HELLO WORLD";</code> <code>var res = str.charAt(0);</code> → H
charCodeAt()	renvoie le code ASCII du caractère se trouvant à une certaine position. <code>var str = "HELLO WORLD";</code> <code>var n = str.charCodeAt(0);</code> → 72
concat()	permet de concaténer plusieurs chaînes de caractères. <code>var str1 = "Hello ";</code> <code>var str2 = "world!";</code> <code>var res = str1.concat(str2);</code>
fromCharCode()	renvoie le caractère associé au code. <code>var res = String.fromCharCode(65);</code> A
endsWith()	Si une chaîne se termine par une sous-chaîne. <code>var str = "Hello world, welcome to the universe.";</code> <code>var n = str.endsWith("universe.");</code> → true
include()	Si une chaîne intègre une autre. <code>var str = "Hello world, welcome to the universe.";</code> <code>var n = str.includes("world");</code> → true
indexOf()	permet de trouver l'indice d'occurrence d'une valeur dans une chaîne <code>var str = "Hello world, welcome to the universe.";</code> <code>var n = str.indexOf("welcome");</code> → 13
lastIndexOf()	permet de trouver l'indice de la dernière occurrence d'une valeur dans une chaîne <code>var str = "Hello planet earth, you are a great planet.";</code> <code>var n = str.lastIndexOf("planet");</code> → 36
match()	recherche les sous-chaînes correspondantes à une expression régulière. <code>var str = "The rain in SPAIN stays mainly in the plain";</code> <code>var res = str.match(/ain/g);</code> → ain, ain, ain
repeat()	Retourne la chaîne répétée le nombre de fois passée en paramètre <code>var str = "Hello world!";</code> <code>str.repeat(2);</code> → Hello world!Hello world!
search()	Recherche une chaîne de caractères pour une valeur spécifiée ou une expression régulière et renvoie la position de l'occurrence. <code>var str = "Try JavaScript!";</code> <code>var n = str.search("JavaScript");</code> → 4
slice()	permet de retourner la portion de la chaîne qui se trouve à l'indice passé en paramètre <code>var str = "Hello";</code> <code>var res = str.slice(2);</code> → "llo" <code>var res = str.slice(-1);</code> → "o" <code>var str = "Hello world!";</code> <code>var res = str.slice(1, 5);</code> → "ello"
split()	Divise une chaîne dans un tableau de sous-chaînes en spécifiant un séparateur de chaînes. <code>var str = "How are you doing today?";</code> <code>var res = str.split(" ");</code> → How, are, you, doing, today?

substr()	extrait une portion de chaîne en lui passant l'indice du premier et le nombre de caractères à extraire. <code>var str = "Hello world";</code> <code>var res = str.substr(3,5);</code> → "llowo"
substring()	extrait une portion de chaîne entre deux indices passés en paramètres. <code>var str = "Hello world!";</code> <code>var res = str.substring(1, 4);</code> "ell"
toLowerCase()	Convertit une chaîne en minuscule
toUpperCase()	Convertit une chaîne en majuscule
trim()	Supprime les espaces aux extrémités d'une chaîne <code>var str = " Hello World! ";</code> <code>alert(str.trim());</code> → "Hello World!"

### 3. Math

Il s'agit d'un objet définissant de nombreuses constantes et fonctions mathématiques.

#### Exemples

```

alert(Math.E);                // la valeur de e (2.718)
alert(Math.PI);               // la valeur de pi
alert(Math.min(5,12));        // 5
alert(Math.max(23,5,7,130,12)); // 130
alert(Math.ceil(25.3));       // 26
alert(Math.floor(25.8));      // 25
alert(Math.random());         // valeur aléatoire entre 0 et 1
var n = Math.floor(Math.random()*nb + min);
alert(n);                     // valeur aléatoire entre min et min+nb (exclus)

```

D'autres fonctions :

- Math.abs(x)      Math.exp(x)      Math.log(x)
- Math.pow(x,y)    Math.sqrt(x)
- Math.sin(x)      Math.cos(x)      Math.tan(x)

### 4. Array

Les tableaux peuvent contenir des données de nature différente. Un tableau en JavaScript est redimensionné automatiquement en fonction de son contenu.

#### Exemples

```

var colors = new Array();           // tableau vide ou encore = [] ;
var colors2 = new Array(20);        // tableau avec 20 cases
var colors3 = new Array("red","blue","green"); // 3 cases

```

```
var colors4 = ["red","blue","green"]; // notation littérale
```

✎ La propriété **length** indique la taille d'un tableau.

```
var colors = ["red","blue","green"];

alert(colors.length); // 3

colors[colors.length]="black"; // nouvelle couleur

colors[99]="pink";

alert(colors.length); // 100

alert(colors[50]): // undefined

colors.length=10; // plus que 10 cases
```

✎ De nombreuses méthodes existent sur les tableaux, en voici quelques-unes :

Méthode	Explication
concat()	Joint plusieurs tableaux et retourne le tableau résultat. <pre>var info1 = ["e11", "e12"]; var info2 = ["e21", "e22", "e23"]; var info = info1.concat(info2); → e11,e12,e21,e22,e23</pre>
copyWithin()	Copie les éléments de tableau dans le tableau, vers et à partir des positions spécifiées. <pre>var fruits = ["Banana", "Orange", "Apple", "Mango"]; fruits.copyWithin(2, 0); → Banana, Orange, Banana, Orange</pre>
fill()	Remplit toutes les cellules avec une même valeur. <pre>var fruits = ["Banana", "Orange", "Apple", "Mango"]; fruits.fill("Kiwi"); → Kiwi, Kiwi, Kiwi, Kiwi</pre>
find()/findIndex()	Retourne la valeur/indice du 1er élément qui satisfait une condition (fonction). <pre>var ages = [3, 10, 18, 20]; function checkAdult(v) {     return v &gt;= 18; } alert(ages.find(checkAdult)) ; → 18</pre>
indexOf()/lastIndexOf()	Retourne l'indice de la 1 <sup>ère</sup> /dernière valeur <pre>var fruits = ["Banana", "Orange", "Apple", "Mango"]; var a = fruits.indexOf("Apple"); → 2</pre>
isArray()	<pre>var fruits = ["Banana", "Orange", "Apple", "Mango"]; alert(Array.isArray(fruits)); → true</pre>
join()	Joint les éléments d'un tableau dans une chaîne <pre>var fruits=["Banana", "Orange", "Apple", "Mango"]; var energy = fruits.join(); → Banana,Orange,Apple,Mango</pre>

pop()/shift()	Enlève le dernier/premier élément d'un tableau et le retourne. <code>var fruits = ["Banana", "Orange", "Apple", "Mango"];</code> <code>fruits.pop();</code> → Banana, Orange, Apple <code>fruits.shift();</code> → Orange, Apple, Mango
push()	Ajout un nouvel élément à la fin d'un tableau et retourne la nouvelle taille. <code>var fruits = ["Banana", "Orange", "Apple", "Mango"];</code> <code>fruits.push("Kiwi");</code> → Banana, Orange, Apple, Mango, Kiwi
reverse()	Inverse un tableau.
slice()	Sélectionne une partie d'un tableau et retourne un tableau <code>var fruits=["Banana", "Orange", "Lemon", "Apple", "Mango"];</code> <code>var citrus = fruits.slice(1, 3);</code> → Orange, Lemon
sort()	Trie un tableau <code>var fruits = ["Banana", "Orange", "Apple", "Mango"];</code> <code>fruits.sort();</code> → Apple, Banana, Mango, Orange
sort(sortFunction())	Pour spécifier le type de tri, spécifier une fonction de tri en parametre. <code>var points = [40, 100, 1, 5, 25, 10];</code> <code>points.sort(function(a, b){return a-b});</code> → 1, 5, 10, 25, 40, 100
splice()	Ajoute/supprime des éléments d'un tableau. <code>var fruits = ["Banana", "Orange", "Apple", "Mango"];</code> <code>fruits.splice(2, 0, "Lemon", "Kiwi");</code> → Banana, Orange, Lemon, Kiwi, Apple, Mango 2: position d'ajout 0: nombre elements à supprimer
unshift()	Ajoute des éléments au début d'un tableau et retourne la taille. <code>var fruits = ["Banana", "Orange", "Apple", "Mango"];</code> <code>fruits.unshift("Lemon", "Pineapple");</code> → Lemon, Pineapple, Banana, Orange, Apple, Mango
toString()	Convertit un tableau en une chaîne

### Exemples

```

Var colors = ["red", "blue", "green"];

alert(colors);           // red,blue,green

alert(colors.join(";")); // red;blue;green

colors.push("black");    //Ajouter un élément à la fin du tableau

alert(colors);           // red,blue,green,black

var item = colors.pop();  // Enlever le dernier élément du tableau

alert(item + " " + colors); // black red,blue,green

var item2= colors.shift(); // Enlever le premier élément du tableau

alert(item2 + " " + colors); // redblue,green

colors.unshift("yellow"); // Ajouter un élément au début du tableau

alert(colors); // yellow,blue,green

```

#### 4.1. Réordonner les tableaux numériques

On peut utiliser *reverse* et *sort*.

##### Exemples

```
var values = [0, 1, 2, 3, 4];
alert(values.reverse()); // 4,3,2,1,0
values = [0, 1, 15, 10, 5];
alert(values.sort()); // 0, 1, 15, 10, 5
alert(values.sort(function(a, b){return a-b})); // 0,1,5,10,15
alert(values.sort(function(a, b){return b-a})); // 15,10,5,1,0
```

## 5. Date

L'objet Date permet de manipuler les dates.

```
var d = new Date();
var d = new Date(milliseconds);
var d = new Date(dateString);
var d = new Date(year, month, day, hours, minutes, seconds, milliseconds);
```

✎ Les principales méthodes sur les dates :

Méthode	Explication
getDate()	Retourne le jour du mois (1-31) d'une date <code>var d = new Date();</code> <code>var n = d.getDate(); → 27</code>
getDay()	Retourne le jour de la semaine (0-6) d'une date. 0 pour dimanche,.. <code>var d = new Date();</code> <code>var n = d.getDay(); → 1</code>
getFullYear()	Retourne l'année d'une date. <code>var d = new Date();</code> <code>var n = d.getFullYear(); → 2017</code>
getMonth()	Retourne le mois d'une date (0-11)
getHours()	Retourne l'heure d'une date (0-23)
getMinutes()	Retourne la minute d'une date (0-59)
getSeconds()	Retourne la seconde d'une date (0-59)
getMilliseconds()	Retourne la milliseconde d'une date (0-999)
now()	Retourne le nombre de milliseconde depuis 01-01-1970
setDate()	Permettent de spécifier les valeurs des éléments d'une date
setDay()	
setFullYear()	
setMonth()	
setHours()	
setMinutes()	
setSeconds()	
setMilliseconds()	

## 6. Le type Object

On peut créer nos propres objets qui seront utilisés pour stocker des données et exposer des fonctions. On peut alors créer une variable objet avec *new Object()* ou de manière littérale (énumération entre accolades).

### Exemples

```
var personne = new Object();
personne.nom = "Wissam" ;
personne.age = 23;
//ou bien
var personne = { nom : "Wissam", age : 23 }
```

✎ Il est possible d'utiliser les crochets pour accéder à un champ.

```
alert(personne.nom);
alert(personne["nom"]);
var attribut="nom"; alert(personne[attribut]);           //23
for (var pers in personne)
    document.writeln(pers + " : " + personne[pers] );
//nom : Wissam age : 23
```