LES CHAÎNES DE CARACTÈRES

Les types primitifs

Nous avons vu, tout au long du cours, que les chaînes de caractères étaient des objets ***String***, les tableaux des ***Array***, etc. C'est toujours vrai mais il convient de nuancer ces propos en introduisant le concept de **type primitif**.

Pour créer une chaîne de caractères, on utilise généralement cette syntaxe :

*var* myString = "Chaîne de caractères primitive";

Cet exemple crée ce que l'on appelle une chaîne de caractères primitive, qui n'est pas un objet String. Pour instancier un objet String, il faut faire comme ceci :

*var* myRealString = new String("Chaîne");

Cela est valable pour les autres objets :

*var* myArray = []; // Tableau primitif

*var* myRealArray = new Array();

*var* myObject = {}; // Objet primitif

*var* myRealObject = new Object();

*var* myBoolean = true; // Booléen primitif

*var* myRealBoolean = new Boolean("true");

*var* myNumber = 42; // Nombre primitif

*var* myRealNumber = new Number("42");

Ce que nous avons utilisé jusqu'à présent était en fait des types primitifs, et non des instances d'objets.

L'objet String

L'objet ***String*** est l'objet que vous manipulez depuis le début du tutoriel : c'est lui qui gère les chaînes de caractères.

**Propriétés**

***String*** ne possède qu'une seule propriété, ***length***, qui retourne le nombre de caractères contenus dans une chaîne. Les espaces, les signes de ponctuation, les chiffres… sont considérés comme des caractères. Ainsi, cette chaîne de caractères contient 21 caractères :

La casse et les caractères

**toLowerCase() et toUpperCase()**

toLowerCase() et toUpperCase() permettent respectivement de convertir une chaîne en minuscules et en majuscules. Elles sont pratiques pour réaliser différents tests ou pour uniformiser une chaîne de caractères. Leur utilisation est simple :

*var* myString = 'tim berners-lee';

myString = myString.toUpperCase(); // Retourne : « TIM BERNERS-LEE »

**Accéder aux caractères**

La méthode charAt() permet de récupérer un caractère en fonction de sa position dans la chaîne de caractères. La méthode reçoit en paramètre la position du caractère :

*var* myString = 'Pauline';

*var* first = myString.charAt(0); // P

*var* last = myString.charAt(myString.length - 1); // e

En fait, les chaînes de caractères peuvent être imaginées comme des tableaux, à la différence qu'il n'est pas possible d'accéder aux caractères en utilisant les crochets : à la place, il faut utiliser charAt().

Certains navigateurs permettent toutefois d'accéder aux caractères d'une chaîne comme s'il s'agissait d'un tableau, c'est-à-dire comme ceci : myString[0]. Ce n'est pas standard, donc il est fortement conseillé d'utiliser myString.charAt(0).

**En résumé**

* Il existe des objets et des types primitifs. Si leur utilisation semble identique, il faut faire attention lors des tests avec les opérateurs ***instanceof*** et ***typeof***. Mais heureusement ***valueOf***() sera d'une aide précieuse.
* Il est préférable d'utiliser les types primitifs, comme nous le faisons depuis le début de ce cours.
* ***String*** fournit des méthodes pour manipuler les caractères : mettre en majuscule ou en minuscule, récupérer un caractère grâce à sa position et supprimer les espaces de part et d'autre de la chaîne.
* ***String*** fournit aussi des méthodes pour rechercher, couper et extraire.
* L'utilisation du caractère tilde est méconnue, mais peut se révéler très utile en couple ***avecindexOf***() ou ***lastIndexOf***().