

## L'essentiel de js

### Commentaire

Une ligne //

Multiple line /\* \*/

### Les type de données

En JavaScript, il existe **six types primitifs**:

Définition : Les types primitifs sont les types de données de base fournis par un langage de programmation. Ils permettent de représenter les données fondamentales.

- Booléen
- Nombre
- String
- Null
- Undefined
- Symbol

### Non primitifs:

- Object
- List (array)
- Function

**Typeof** : affiche le type d'une variable ou constante.

## Opérateurs arithmétiques

- Addition: +
- Soustraction: -
- Multiplication: \*
- Division: /
- Modulo: %
- Exponentiel: \*\*
- Incrementation : ++
- Decrementation: –

les opérateurs d'incrément et de décrémentation ne peuvent être utilisés que sur des variables ; tenter de les utiliser sur un nombre brut entraînera une erreur. En d'autres termes, vous ne pouvez pas faire ceci => 5++, ou ceci => 7–

Opération d'affectation ( = ): Elle consiste à attribuer une valeur à une variable ou une constante.

Il est possible de combiner les opérateurs arithmétiques avec l'opérateur d'affectation "=" afin de créer des annotations abrégées.

Expression	Abbreviation	Description
$x = x + y$	$x += y$	additionne x et y et affecte le résultat a x.
$x = x - y$	$x -= y$	soustrait x et y et affecte le résultat a x.
$x = x * y$	$x *= y$	multipli x et y et affecte le résultat a x.
$x = x / y$	$x /= y$	divise x et y et affecte le résultat a x.
$x = x \% y$	$x \% = y$	calcule le modulo de x et y et affecte le résultat a x.

## Opérateurs de comparaison

Les opérateurs de comparaison évaluent les variables comme vraies (true) ou fausses (false). Voici une liste des opérateurs de comparaison.

Opérateur	Comparaison	Description
==	$x == y$	Retourne vrai si x et y sont égaux.
===	$x === y$	Retourne vrai si x et y sont égaux et sont du même type.
!=	$x != y$	Retourne vrai si x et y ne sont égaux.
>	$x > y$	Retourne vrai si x est supérieur a y .
>=	$x >= y$	Retourne vrai si x est supérieur ou égal à y.
<	$x < y$	Retourne vrai si x est inférieur a y .
<=	$x <= y$	Retourne vrai si x est inferieur ou égal à y.

## Truthy & Falsy:

Truthy : Quelque chose qui s'évalue à vrai.

Falsy : Quelque chose qui s'évalue à Faux.

C'est essentiellement logique. Le nombre Un (1) est évalué à vrai, zéro (0) évaluer à faux. Un objet quel qu'il soit (y compris les fonctions, les tableaux, les objets RegExp, etc.) est toujours vrai.

Il n'y a que six valeurs évaluées comme fausses (Falsy) en JavaScript : undefined, null, NaN(Not A Number), 0, "" (chaîne vide), et false.

Ex : console.log(Boolean(undefined))

## Opérateurs logiques et booléens

Il y a un grand malentendu autour des opérateurs logiques AND et OR en JavaScript. Ils **ne renvoient pas** le vrai ou le faux ! Plus précisément :

Opérateur	Usage	Description
et (&&)	x && y	Renvoie x si x est évaluée à <b>fausse</b> ; sinon, renvoie y.
ou (  )	x    y	Renvoie x si x est évaluée à <b>vraie</b> ou y si y est évaluée à <b>vraie</b>

## Inversion de logique

Pour inverser une logique en javascript , on utilise le symbole << ! >> , true devient false, et false devient true.

Et bien nous pouvons doubler ce symbole << !! >>. **convertir une valeur en booléen.**

## les fonctions

Une fonction est un morceau de code réutilisable sans avoir à l'écrire encore et encore.

- Sans paramètre
- Avec paramètre
- Affectation d'une fonction à une variable
- Fonction anonyme

## Structure de controle

- If, else if ,else
- switch

## Structure de repetition

- For , for in, for of
- While
- Do while

