Javascript : les opérateurs

Contenu

[Les opérateurs de calcul 2](#_Toc510080884)

[Les opérateurs de comparaison 3](#_Toc510080885)

[Les opérateurs associatifs 4](#_Toc510080886)

[Les opérateurs logiques 5](#_Toc510080887)

[Les opérateurs d'incrémentation 6](#_Toc510080888)

[La priorité des opérateurs JavaScript 7](#_Toc510080889)

# Les opérateurs de calcul

Dans les exemples, la valeur initiale de x sera toujours égale à 11.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signe** | **Nom** | **Signification** | **Exemple** | **Résultat** |
| + | Plus | addition | x + 3 | 14 |
| - | Moins | soustraction | x - 3 | 8 |
| \* | Multiplié par | multiplication | x\*2 | 22 |
| / | Divisé | par division | x /2 | 5.5 |
| % | Modulo | reste de la division par | x%5 | 1 |
| = | à la valeur | affectation | x=5 | 5 |

**L’opérateur + sert aussi à concaténer des chaînes de caractères.**

# Les opérateurs de comparaison

|  |  |  |  |
| --- | --- | --- | --- |
| **Signe** | **Nom** | **Exemple** | **Résultat** |
| == | Egal | x==11 | true |
| < | Inférieur | x<11 | false |
| <= | Inférieur ou égal | x<=11 | true |
| > | Supérieur | x>11 | false |
| >= | Supérieur ou égal | x>=11 | true |
| != | Différent | x!=11 | false |

## 

**On confond souvent le = et le == (deux signes =). Le = est un opérateur de valeur tandis que le == est un opérateur de comparaison. Cette confusion est une source classique d’erreur de programmation.**

# Les opérateurs associatifs

On appelle ainsi les opérateurs qui réalisent un calcul dans lequel une variable intervient des deux côtés du signe = (ce sont donc en quelque sorte des opérateurs d'attribution).

Dans les exemples suivants x vaut toujours 11 et y aura comme valeur 5.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signe** | **Description** | **Exemple** | **Signification** | **Résultat** |
| += | plus égal | x += y | x = x + y | 16 |
| -= | moins égal | x -= y | x = x - y | 6 |
| \*= | multiplié égal | x \*= y | x = x \* y | 55 |
| /= | divisé égal | x /= y | x = x / y | 2.2 |

# Les opérateurs logiques

Aussi appelés opérateurs booléens, ils servent à vérifier deux ou plusieurs conditions.

|  |  |  |  |
| --- | --- | --- | --- |
| **Signe** | **Nom** | **Exemple** | **Signification** |
| && | et | (condition1) && (condition2) | condition1 et condition2 |
| || | ou | (condition1) || (condition2) | condition1 ou condition2 |

# Les opérateurs d'incrémentation

Ces opérateurs vont augmenter ou diminuer la valeur de la variable d'une unité. Ce qui sera fort utile, par exemple, pour mettre en place des boucles.

Dans les exemples x vaut 3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signe** | **Description** | **Exemple** | **Signification** | **Résultat** |
| x++ | incrémentation  (x++ est le même que x=x+1) | y = x++ | 3 puis plus 1 | 4 |
| x-- | décrémentation  (x-- est le même que x=x-1) | y= x-- | 3 puis moins 1 | 2 |

# La priorité des opérateurs JavaScript

Les opérateurs s'effectuent dans l'ordre suivant de priorité (du degré de priorité le plus faible ou degré de priorité le plus élevé).

Dans le cas d'opérateurs de priorité égale, de gauche à droite.

|  |  |
| --- | --- |
| **Opération** | **Opérateur** |
| ,  = += -= \*= /= %=  ? :  ||  &&  == !=  < <= >= >  + -  \* /  ! - ++ --  ( ) | virgule ou séparateur de liste  affectation  opérateur conditionnel  ou logique  et logique  égalité  relationnel  addition soustraction  multiplier diviser  unaire  parenthèses |