

DESENVOLUPAMENT WEB EN ENTORN CLIENT

Unitat 3 - Operadors

Xuso Ortí Monton





Continguts

1 Assignació	3
2 Increment i decrement	3
3 Lògics	4
Negació	4
AND	5
OR	5
Matemàtics	6
Relacionals	6
Igualtat estricta ===	7
Operador ternari	7



1 Assignació

Utilitza l'operador = per a assignar un valor a una variable.

```
var nom = "Xuso";
```

L'assignació sempre es realitza declarant la variable a l'esquerra i el valor a la dreta. Per tant aquesta sentència seria incorrecta.

```
"Xuso" = var nom; //sentència incorrecta
```

2 Increment i decrement

S'utilitzen per a incrementar o decrementar una variable.

```
var a = 1;  
a++; //el valor de a ara és 2  
a--; //el valor de a ara és 1
```

També seria correcte el següent codi

```
var a = 1;  
++a; //el valor de a ara és 2  
--a; //el valor de a ara és 1
```

Pot semblar que el funcionament és el mateix, però hi ha una subtil diferència. Si l'operador ++ s'indica com a prefix de l'identificador de la variable, el seu valor s'incrementa abans de realitzar qualsevol altra operació. Si l'operador ++ s'indica com sufix de l'identificador de la variable, el seu valor s'incrementa després d'executar la sentència en la qual apareix.

```
var numero1 = 5;  
var numero2 = 2;  
numero3 = numero1++ + numero2;  
// numero3 = 7, numero1 = 6
```

```
var numero1 = 5;  
var numero2 = 2;  
numero3 = ++numero1 + numero2;  
// numero3 = 8, numero1 = 6
```

Per tant, en la instrucció `numero3 = numero1 ++ + numero2;` el valor de `numero1` s'incrementa després de realitzar l'operació (primer es suma i `numero3` val 7, després s'incrementa el valor de `numero1` i val 6). No obstant això, en la instrucció `numero3 = ++ numero1 + numero2;` en primer



lloc s'incrementa el valor de numero1 i després es realitza la suma (primer s'incrementa numero1 i val 6, després es realitza la suma i numero3 val 8).

3 Lògics

S'utilitzen per a operacions de presa de decisions sobre variables amb un valor de tipus booleà.

Negació

Emprem l'operador `!` per a negar el valor d'una variable.

```
var visible = true;
alert(!visible); // Muestra "false" y no "true"
```

El funcionament és el següent.

Variable	!Variable
true	false
false	true

Si la variable original és de tipus booleà, és molt senzill obtenir la seva negació. No obstant això, què passa quan la variable és un nombre o una cadena de text? Per obtenir la negació en aquest tipus de variables, es realitza en primer lloc la seva conversió a un valor booleà.

- Si la variable conté un número, el valor serà `false` en cas que el número siga un 0, i valdrà `true` en cas que la variable tinga altre valor.
- Si la variable és una cadena valdrà `false` en cas que la cadena estiga buida i valdrà `true` en cas que continga qualsevol altre valor.

```
var cantidad = 0;
vacio = !cantidad; // vacio = true
cantidad = 2;
vacio = !cantidad; // vacio = false
var mensaje = "";
mensajeVacio = !mensaje; // mensajeVacio = true
mensaje = "Bienvenido";
mensajeVacio = !mensaje; // mensajeVacio = false
```



AND

Utilitza l'operador `&&` i el funcionament és el següent.

Variable1	Variable2	Valor
false	false	false
false	true	false
true	false	false
true	true	true

```
var valor1 = true;
var valor2 = false;
resultado = valor1 && valor2; // resultado = false
valor1 = true;
valor2 = true;
resultado = valor1 && valor2; // resultado = true
```

OR

Utilitza l'operador `||` i el seu funcionament és el següent.

Variable1	Variable2	Valor
false	false	false
false	true	true
true	false	true
true	true	true

```
var valor1 = true;
var valor2 = false;
resultado = valor1 || valor2; // resultado = true
valor1 = false;
valor2 = false;
resultado = valor1 || valor2; // resultado = false
```



Matemàtics

Utilitzen els operadors suma +, resta -, multiplicació *, divisió / i mòdul %. No confondre en mòdul amb el percentatge, el mòdul calcula el resto d'una divisió entre números sencers. Per exemple el mòdul resultant de dividir 21 entre 7 seria 0 ja que el resultat de la divisió és 3 i el resto 0.

```
var numero1 = 10;
var numero2 = 5;
resultado = numero1 / numero2; // resultado = 2
resultado = 3 + numero1;      // resultado = 13
resultado = numero2 - 4;      // resultado = 1
resultado = numero1 * numero2; // resultado = 50
resultado = numero1 % numero2; //resultado = 0
```

Es poden abreviar les assignacions fent ús de la següent sintaxi.

```
var num = 2;
num += 3; //num = num + 3 = 5
```

Relacionals

Són els operadors menor que <, major que >, igual que == i distint que !=. Cal tindre molta cura ja que és habitual confondre l'operador d'assignació = en el de comparació ==.

```
var numero1 = 3;
var numero2 = 5;
resultado = numero1 > numero2; // resultado = false
resultado = numero1 < numero2; // resultado = true
numero1 = 5;
numero2 = 5;
resultado = numero1 >= numero2; // resultado = true
resultado = numero1 <= numero2; // resultado = true
resultado = numero1 == numero2; // resultado = true
resultado = numero1 != numero2; // resultado = false
//per a cadenes de text
var texto1 = "hola";
var texto2 = "hola";
var texto3 = "adios";
resultado = texto1 == texto3; // resultado = false
resultado = texto1 != texto2; // resultado = false
resultado = texto3 >= texto2; // resultado = false
```

Quan s'utilitzen cadenes de text, els operadors «més gran que» > i «menor que» < segueixen un raonament no intuïtiu: es compara lletra a lletra començant des de l'esquerra fins que es trobi una diferència



entre les dues cadenes de text. Per determinar si una lletra és més o menys que una altra, les majúscules es consideren menors que les minúscules i les primeres lletres de l'alfabet són menors que les últimes (a és menor que b, b és menor que c, A és menor que a, etc.)

Igualtat estricta ===

L'operador d'estructura igualtat === revisa si dos operands són iguals i produeix un resultat Booleà. A diferència de l'operador d'igualtat regular ==, l'operador d'estructura igualtat sempre considera que els operands de diferent tipus de valor són diferents i mai similars.

```
console.log(1 === 1);  
// expected output: true  
console.log('hello' === 'hello');  
// expected output: true  
console.log('1' === 1);  
// expected output: false  
console.log(0 === false);  
// expected output: false
```

Operador ternari

És l'únic operador en javascript que disposa de tres operands, i s'utilitza com a abreviatura de la instrucció if. Serveix per a realitzar una acció si la comparació és vertadera i altra si és falsa.

condició ? expr1 : expr2

```
//si és membre la quota serà de 5 euros i si no serà de 10 euros  
"La quota és de: " + (isMember ? "5.00€" : "10.00€")
```