



Clase 13

JavaScript Parte 2



JavaScript

Temas: Operadores de asignación, aritméticos, de cadena, relacionales, lógicos, bit a bit - Condicionales (if, if else, else if, operador ternario, switch) - Bucles: While y For



Operadores de Javascript



Operadores de Aritméticos: se utilizan para realizar operaciones aritméticas.

Operator	Description
+	Suma
-	Resta
*	Multiplicación
**	Exponenciación
/	División
%	Módulo: resto de dividir
i++	operador de incremento incrementa i en 1
i--	operador de decremento decrementa i en 1



Operadores de Javascript



Operadores posfijo y prefijo: se utilizan para realizar operaciones aritméticas.

Operator	Descripcion	Ejemplo
i++	incremento posfijo	a=i++ primero a=i y despues i=i +1
++ i	incremento prefijo	a=++i primero i=i +1 y despues a=i
i - -	decremento posfijo	a=i - - primero a=i y despues i=i - 1
- - i	decremento prefijo	a=--i primero i=i - 1 y despues a=i

Practica:

<https://repl.it/@LeandroGrammati/PepperySeashellOpentracker#script.js>



Operadores de Javascript



Operadores de Asignación

Operator	Ejemplo	Igual que
=	<code>x = 3</code>	<code>x = 3</code>
+=	<code>x += y</code>	<code>x = x + y</code>
-=	<code>x -= y</code>	<code>x = x - y</code>
*=	<code>x *= y</code>	<code>x = x * y</code>
/=	<code>x /= y</code>	<code>x = x / y</code>
%=	<code>x %= y</code>	<code>x = x % y</code>
**=	<code>x **= y</code>	<code>x = x ** y</code>



Operadores de Javascript



Operadores de cadena

El operador + también se puede usar para agregar (concatenar) cadenas, se llama operador de concatenación.

```
var txt1 = "John";
```

```
var txt2 = "Doe";
```

```
var txt3 = txt1 + " " + txt2; // El resultado de txt3 será: John Doe
```

El += operador de asignación también se puede usar para agregar (concatenar) cadenas:

```
var txt1 = "Hola que tal ";
```

```
txt1 += "como estás?"; El resultado de txt1 será: Hola que tal como estás"
```

Ejemplo

```
var x = 5 + 5; //x valdrá 10
```

```
var y = "Hola" + 5; // y valdrá "Hola5"
```

Si agrega un número y una cadena, el resultado será una cadena!



Operadores de Javascript



Operadores de comparación:

Operator	Description
==	equal to
===	equal value and equal type
!=	not equal
!==	not equal value or not equal type
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
?	ternary operator



Operadores de Javascript



Operadores Lógicos:

Operador	Descripción
&&	operador lógico and
	operador lógico or
!	operador lógico not

p	q	p && q
V	V	V
V	F	F
F	V	F
F	F	F

p	q	p q
V	V	V
V	F	V
F	V	V
F	F	F

p	q	p XOR q
V	V	F
V	F	V
F	V	V
F	F	F

p	! p
V	F
F	V



Operadores de Javascript



Operadores de bit a bit

Los operadores de bits funcionan con números de 32 bits. Cualquier operando numérico de la operación se convierte en un número de 32 bits. El resultado se convierte de nuevo a un número de JavaScript.

Nro Decimal	Numero Binario
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

Operador	Descripcion	Ejemplo decimal	Ejemplo binario	Resultado binario	Resultado Decimal
&	AND	5 & 1	0101 & 0001	0001	1
	OR	5 1	0101 0001	0101	5
~	NOT	~ 5	~0101	1010	10
^	XOR	5 ^ 1	0101 ^ 0001	0100	4
<<	Desplazamiento a la izquierda de llenado cero	5 << 1	0101 << 1	1010	10
>>	Desplazamiento a la derecha	5 >> 1	0101 >> 1	0010	2



Estructuras de control



Condicionales: Al hacer un programa necesitaremos establecer **condiciones** o **decisiones**, donde buscamos que el navegador realice una **acción A** si se **cumple** una condición o una **acción B** si **no se cumple**.

Estructura de control	Descripción
<pre>if (condition) { // bloque de codigo que se ejecuta si la condicion es verdadera }</pre>	Condición simple: Si ocurre algo, haz lo siguiente...
<pre>if (condition) { // bloque de codigo que se ejecuta si la condicion es verdadera } else { // bloque de codigo que se ejecuta si la condicion es falso }</pre>	Condición con alternativa: Si ocurre algo, haz esto, sino, haz lo esto otro...
<pre>if (condition1) { // bloque de codigo que se ejecuta si la condicion es verdadera } else if (condition2) { // bloque de codigo que se ejecuta si la condicion1 es falsa y la condicion2 es verdadera } else { // bloque de codigo que se ejecuta si la condicion1 es falsa y la condicion2 es falsa }</pre>	La declaración else if para especificar una nueva condición si la primera condición es falsa.



Estructuras de control



Condicionales: Al hacer un programa necesitaremos establecer **condiciones** o **decisiones**, donde buscamos que el navegador realice una **acción A** si se **cumple** una condición o una **acción B** si **no se cumple**.

Estructura de control	Descripción
<pre>condición ? expr1 : expr2</pre>	Operador ternario: Equivalente a If/else, método abreviado.
<pre>switch(expression) { case x: // bloque de codigo break; case y: // bloque de codigo break; default: // bloque de codigo }</pre>	<p>Switch</p> <ul style="list-style-type: none">• La expresión de cambio se evalúa una vez.• El valor de la expresión se compara con los valores de cada caso.• Si hay una coincidencia, se ejecuta el bloque de código asociado.• Si no hay coincidencia, se ejecuta el bloque de código predeterminado.



Estructuras de control



Práctica:

<https://repl.it/@LeandroGrammati/PettyHastyServer#script.js>



Estructuras de control



Estructura de control repetitivas

El bucle `for` se suele usar cuando se sabe cuantas repeticiones se tienen que hacer

Tiene la siguiente sintaxis:

```
for (declaracion 1; declaracion 2; declaracion 3) {  
    // bloque de código a ejecutar  
}
```

Ejemplo: mostrar por pantalla los números enteros del 1 a 10

```
for (var i=1; i<=10; i++) {  
    console.log(i);  
}
```

Ejemplo: mostrar por pantalla los números múltiplos del 2 del 2 hasta 100

```
for (var i=2; i<=100; i+=2) {  
    console.log(i);  
}
```



Estructuras de control



Estructura de control repetitivas

El bucle while

El bucle `while` se usa cuando el fin de la repetición depende de una condición.

Tiene la siguiente sintaxis:

```
while (condition) {  
    // bloque de codigo  
}
```

Ejemplo: Leer notas hasta que ingrese -1

```
var nota=0;  
nota=Number(prompt("Ingrese una nota (para finalizar ingrese -1):"));  
while ( nota != -1){  
    nota=Number(prompt("Ingrese una nota (para finalizar ingrese -1):" ));  
}
```