



Tipos de Datos, y Operadores



Tipos de datos

- En JavaScript, a diferencia de otros lenguajes, a la hora de declarar variables, no debemos indicar su tipo.
- En JavaScript las variables adquieren un tipo de dato, cuando se le asigna un valor específico.
- Por defecto, las variables que declaramos sin asignar un valor, toman el valor primitivo **undefined**



Tipos de datos mas basicos:

- Los tipos de datos mas básicos en Javascript son:
- number
- string
- boolean
- null
- undefined
- Se les llaman tipos de datos primitivos.

number

- Representa números
- `let` edad = 20;
- `let` vidasRestantes = 4;

```
> Number.MAX_VALUE  
< 1.7976931348623157e+308  
_____  
> Number.MIN_VALUE  
< 5e-324
```

String

- Se utiliza para representar datos textuales, o cadenas de texto.
- La cadena de texto debe estar entre comillas simples o dobles.
- `let` nombre = "Juancito";
- `let` apellido = 'Rodríguez';

Boolean

- Representa datos del tipo booleano (verdadero o falso)
- `let tieneVidas = true;`
- `let debePagarIVA = false;`

Null

- Null es bastante similar a undefined, pero tiene un comportamiento diferente.
- Mas adelante veremos sus diferencias, cuando veamos el estándar EcmaScript.
- Por ahora, usaremos null para cuando queramos “dejar vacía” una variable.
- `var apellido = “Olivera”;`
- `apellido = null`



typeof()

- Es un operador que nos dice el tipo de dato de una expresión.
- Se puede usar de dos maneras:
- `typeof` expresion
- `typeof`(expresion)
- Nos devuelve una cadena con el tipo de valor

typeof() - Ejemplos:

- `typeof("Hola")` // "string"
- `typeof(1)` // "number"
- `typeof(false)` // "boolean"
- `typeof([])` // "object"

Operadores

- Nos sirven para manipular el valor de las variables, realizar operaciones matemáticas con sus valores y comparar diferentes variables.
- Con los operadores podemos realizar cálculos y tomar decisiones lógicas en función de comparaciones y otros tipos de condiciones.
- Hasta el momento vimos dos operadores:
- +
- `typeof`

Operadores aritméticos:

Operador	Uso	Ejemplo
+	Sumar	1 + 2
-	Restar	1 - 2
/	Dividir	10 / 2
*	Multiplicar	5 * 4
%	Obtener resto de división	10 % 3
++	Incremento	numero++;
--	Decremento	numero--;

Veamos ejemplos en la consola...

Operadores de asignación

Operador	Nombre	Ejemplo	Equivalencia
=	Asignación	$x = y$	$x = y$
+=	Asignación de adición	$x += y$	$x = x + y$
-=	Asignación de resta	$x -= y$	$x = x - y$
*=	Asignación de producto	$x *= y$	$x = x * y$
/=	Asignación de división	$x /= y$	$x = x / y$
%=	Asignación de residuo	$x \% = y$	$x = x \% y$
**=	Asignación de exponenciación	$x ** = y$	$x = x ** y$

Estos son algunos de los tantos operadores que hay ...

Operadores de Comparación

Estos operadores retornan true o false, según si la comparación lógica sea cierta o no

Operador		Ejemplo	Valor
==	Igual	1 == 1	true
!=	Distinto	"Hola" != "Adios"	false
===	Estrictamente igual	1 === "1"	false
!==	Estrictamente distinto	"true" !== true	true
>	Mayor	10 > 10	false
>=	Mayor o igual	10 >= 10	true
<	Menor	5 < 6	true
<=	Menor o igual	5 <= 10	true

Veamos ejemplos en la consola...

Comportamientos de los operadores

- Notemos que el operador + lo podemos usar de dos formas:
- Concatenar cadenas
- Sumas valores numéricos (operación aritmética)

- `var suma = 1 + 1;`

- `var mensaje = "Hola " + "mundo";`

```
> var suma = 1 + 1;
< undefined
> console.log(suma);
2
< undefined
> var mensaje = "Hola " + "mundo";
< undefined
> console.log(mensaje);
Hola mundo
```

Comportamiento de los operadores

- El operador `+=` también lo podemos usar para tratar con strings.

```
> let texto = "Me llamo "  
< undefined  
  
> texto += "Joaquin"  
< "Me llamo Joaquin"  
  
> console.log(texto);  
Me llamo Joaquin  
  
< undefined
```



Próxima clase:

- Comenzamos a trabajar con las estructuras de control:
- If
- If - else



Ejercicio de deberes:

- Programar una pequeña calculadora en JavaScript.
- Solicitar dos números con la función `prompt()`
- Convertirlos a valores del tipo `number`.
- Mostrar usando alertas, la suma, resta, multiplicación, y división entre los números ingresados.
- Por consultas: joaquin.p.olivera@gmail.com



Bibliografía

- https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Expressions_and_Operators
- https://developer.mozilla.org/es/docs/Web/JavaScript/Data_structures