



Clase 12

JavaScript Parte 1



JavaScript

Temas: Introducción a la programación web, qué es y para qué sirve JavaScript ECMAScript 5 vs ECMAScript 6 - Uso de consola - Variables
- Tipos de Datos - Objeto Math - Number, parseInt y parseFloat



Qué es Javascript?



Javascript es un **lenguaje de programación**, o lo que es lo mismo, un mecanismo con el que podemos decirle a nuestro navegador que tareas debe realizar, en qué orden y cuantas veces por ejemplo.

Programar no es más que decirle a una máquina que cosas debe hacer y cómo debe hacerlas. Eso significa que no podemos pasar por alto nada.



Versiones de ECMAScript

A lo largo de los años, Javascript ha ido sufriendo modificaciones que los navegadores han ido implementando para acomodarse a la última versión de **ECMAScript** cuanto antes. La lista de versiones de ECMAScript aparecidas hasta el momento son las siguientes, donde encontramos las versiones enmarcadas en lo que podemos considerar **el pasado de Javascript**:

Ed.	Fecha	Nombre formal / informal	Cambios significativos
1	JUN/1997	ECMAScript 1997 (ES1)	Primera edición
5	DIC/2009	ECMAScript 2009 (ES5)	Strict mode, JSON, etc...
6	JUN/2015	ECMAScript 2015 (ES6)	Clases, módulos, generadores, hashmaps, sets, for of, proxies...
7	JUN/2016	ECMAScript 2016	Array includes(), Exponenciación **
8	JUN/2017	ECMAScript 2017	Async/await
9	JUN/2018	ECMAScript 2018	Rest/Spread operator, Promise.finally()...
10	JUN/2019	ECMAScript 2019	Flat functions, trimStart(), errores opcionales en catch...
11	JUN/2020	ECMAScript 2020	Dynamic imports, BigInt, Promise.allSettled



Cómo comenzar con Javascript?



Una forma sería agregar la etiqueta `<script>` en la etiqueta `<head>`

```
<html>
  <head>
    <title>Título de la página</title>
    <script>
      console.log("¡Hola!");
    </script>
  </head>
  <body>
    <p>Ejemplo de texto.</p>
  </body>
</html>
```



Cómo comenzar con Javascript?



Para relacionar un **documento Javascript** desde una página web, igual que antes, utilizaremos la etiqueta `<script>`, sólo que en este caso, haremos referencia al archivo **Javascript** con un atributo `src` (*source*), como se ve en el siguiente ejemplo:

Contenido del archivo html

```
<html>
  <head>
    <title>Título de la página</title>
    <script src="js/index.js"></script>
  </head>
  <body>
    <p>Ejemplo de texto.</p>
  </body>
</html>
```

Contenido del archivo index.js ubicado en la carpeta js

```
console.log(" Bienvenidos a Codo a Codo ");
```

Más adelante veremos que la etiqueta `<script>` se suele colocar justo antes de la etiqueta de cierre `</body>`



La Consola



El clásico primer ejemplo cuando se comienza a programar, es crear un programa que muestre por pantalla un texto, generalmente el texto «**Hola Mundo**». También podemos hacer operaciones numéricas

```
console.log("Hola Mundo");  
console.log(2 + 2);
```



La consola

`console.log()` Muestra la información proporcionada en la consola Javascript.

`console.info()` Equivalente al anterior. Se utiliza para mensajes de información.

`console.warn()` Muestra información de advertencia. Aparece en amarillo.

`console.error()` Muestra información de error. Aparece en rojo.

`console.clear()` Limpia la consola. Equivalente a pulsar CTRL+L o escribir `clear()`.

```
console.log(";Hola a todos! Observen este número: ", 5 + 18);
```



Variables



- **Variables:** Es el nombre genérico que se le da a pequeños espacios de memoria donde guardás una información determinada, de forma muy similar a las incógnitas en matemáticas. Un programa puede tener muchas variables, y cada una de ellas tendrá un **nombre**, un **valor** y un **tipo de dato**. El nombre se utiliza para diferenciarlas unas de otras y hacer referencia a ellas, el valor es la información que contienen y el tipo de dato es la naturaleza de ese valor. Se llaman variables porque podemos cambiarle su valor a lo largo del programa, según necesitemos.

```
var num = 5; // nombre: num, valor: 5, tipo de dato: número
```

```
var y = "Hola"; // nombre: y, valor: Hola, tipo de dato: texto
```

```
var nom = "Juan"; // nombre: nom, valor: Juan, tipo de dato: texto
```




Constantes



Constantes: Es el mismo concepto de una variable, salvo que en este caso, la información que contiene es siempre la misma (*no puede variar*).

```
const varname1 = value1;
```

```
const pi=3.141592;
```

```
const iva=21;
```



Tipos de datos

Las variables de JavaScript pueden contener muchos tipos de datos : números, cadenas, objetos y más:

JavaScript tiene tipos dinámicos. Esto significa que la misma variable se puede utilizar para contener diferentes tipos de datos:

```
var x;      // ahora es indefinido
```

```
x = 5;      // ahora es numérico
```

```
x = "Juan"; // ahora es una cadena de caracteres o string
```

JavaScript tiene solo un tipo numérico, no como los demás lenguajes de programación que suelen tener varios tipos de datos numéricos. Los números se pueden escribir con o sin decimales:

```
var x1 = 34.00;    // con decimales
```

```
var x2 = 34;       // sin decimales
```



Objeto Math



Math es un objeto que tiene propiedades y métodos para constantes y funciones matemáticas. Todas las propiedades y métodos de `Math` son estáticos.(no necesito llamar al constructor)

Propiedades:

[Math.E](#) Constante de Euler, la base de los logaritmos naturales, aproximadamente 2.718

[Math.PI](#) Ratio de la circunferencia de un círculo respecto a su diámetro, aproximadamente 3.14159.

[Math.LN2](#) Logaritmo natural de 2, aproximadamente 0.693

[Math.LN10](#) Logaritmo natural de 10, aproximadamente 2.303

[Math.LOG2E](#) Logaritmo de E con base 2, aproximadamente 1.443

[Math.LOG10E](#) Logaritmo de E con base 10, aproximadamente 0.434

[Math.SQRT1_2](#) Raíz cuadrada de 1/2; Equivalentemente, 1 sobre la raíz cuadrada de 2, aproximadamente 0.707.

[Math.SQRT2](#) Raíz cuadrada de 2, aproximadamente 1.414.



Objeto Math



Métodos:

[Math.abs\(x\)](#) Devuelve el valor absoluto de un número.

[Math.trunc\(x\)](#) Devuelve la parte entera del número x, la eliminación de los dígitos fraccionarios

[Math.pow\(x, y\)](#) Las devoluciones de base a la potencia de exponente, que es, baseexponent..

[Math.log\(x\)](#) Devuelve el logaritmo natural (log, también ln) de un número.

[Math.sin\(x\)](#) Devuelve el seno de un número

[Math.cos\(x\)](#) Devuelve el coseno de un número.

[Math.tan\(x\)](#) Devuelve la tangente de un número.

[Math.random\(\)](#) Devuelve un número pseudo-aleatorio entre 0 y 1.

[Math.round\(x\)](#) Devuelve el valor de un número redondeado al número entero más cercano.

[Math.log10\(x\)](#) Devuelve el logaritmo en base 10 de x.

[Math.log2\(x\)](#) Devuelve el logaritmo en base 2 de x.

[Math.sqrt\(x\)](#) Devuelve la raíz cuadrada positiva de un número.

Para seguir investigando: [Clase Math completa](#)



Objeto Number



Number es un objeto primitivo que permite representar y manipular valores numéricos. El constructor Number contiene constantes y métodos para trabajar con números. Valores de otro tipo pueden ser convertidos a números usando la función Number().

Sintaxis:

```
new Number(value);
```

Ejemplos:

```
var a = new Number('123'); // a === 123 es false
```

```
var b = Number('123'); // b === 123 es true
```

```
a instanceof Number; // es true
```

```
b instanceof Number; // es false
```

Para seguir investigando: [Objeto Number](#)



parseInt()



Convierte (parsea) un argumento de tipo cadena y devuelve un entero de la base especificada.

`parseInt()` es una función de alto nivel y no está asociada a ningún objeto.

Sintaxis

`parseInt(string, base);`

Ejemplos:

`parseInt("F", 16);` // F en hexadecimal es igual 15 en decimal

`parseInt("15", 10);` // devuelve 15 en decimal

`parseInt("1111", 2);` // 1111 en binario es igual al 15 decimal

`parseInt("15");` // devuelve 15 en decimal



parseFloat()



Convierte (parsea) un argumento de tipo cadena y devuelve un número de punto flotante.

`parseFloat()` es una función de alto nivel y no está asociada a ningún objeto.

Sintaxis

`parseFloat(cadena)`

Ejemplo

```
parseFloat("3.14"); // devuelve el numero 3.14
```



Ejemplo de Javascript



Para practicar: <https://repl.it/@LeandroGrammati/JS#script.js>



Inspeccionar el html



Inspeccionar el html y ver la consola: <https://adoring-wilson-63f020.netlify.app/>