

SINTAXIS Y CÓDIGO

CODIGO JAVASCRIPT

- JavaScript tiene sus propias reglas para la sintaxis, aunque respeta los estándares de muchos lenguajes de programación lógicos. Existen dos maneras de escribir código en JavaScript.



Figura 1.2

Código JavaScript dentro y fuera de un archivo HTML.

¿CÓMO ESCRIBIR CÓDIGO JS?

- Dentro de un archivo html, entre medio de las etiquetas

```
<script>  
    // Aquí se escribe el código JS  
</script>
```

¿CÓMO ESCRIBIR CÓDIGO JS?

- En un archivo individual con extensión .js Ejemplo: mi-archivo.js Recuerda no utilizar espacios ni mayúsculas en los nombres de archivo

```
<script src="js/main.js"></script>
```

```
<!DOCTYPE html>
<html>
<title>Myfpschool</title>
<head>
<script src="script.js"></script>
</head>
<body>
</body>
</html>
```

La etiqueta `<script src="script.js"></script>` indica que el código JavaScript está contenido en un archivo aparte llamado `script.js`, que se encuentra en la misma carpeta que este archivo `index.html`.

BUENAS PRACTICAS

- Como se puede observar, aunque el resultado es el mismo, la forma de distribuir el código HTML y el JavaScript no es igual. El mejor consejo es utilizar archivos separados para HTML y JavaScript, salvo que las líneas de código de este último sean mínimas y no vayan a ser modificadas prácticamente nunca.
- Las ventajas de tener el JavaScript en un fichero separado son que las páginas cargarán mucho más rápido, se independiza el HTML del código y, como se puede adivinar, el JavaScript será mucho más fácil de mantener.
- Los proyectos suelen tener los archivos script en una carpeta aparte llamada script, para tener el contenido mas ordenado.

SINTAXIS: REGLAS CLARAS

- No se tienen en cuenta los espacios en blanco y las nuevas líneas (al igual que HTML).
- Se distinguen las mayúsculas y minúsculas.
- Se pueden incluir bloques de comentarios:

```
<script>  
    // Comentario simple: una línea  
    /* Comentario de más de una línea I  
       Comentario de más de una línea II */  
</script>
```

SINTAXIS: PALABRAS RESERVADAS

- Palabras reservadas: son las palabras que se utilizan para construir las sentencias de JavaScript y que por tanto no pueden ser utilizadas libremente. Las palabras actualmente reservadas por JavaScript son:

break, case, catch, continue, default, let

delete, do, else, finally, for, function, if, in,

instanceof, new, return, switch, this, throw, try,

typeof, var, void, while, with, etc.

LAS 10 REGLAS BÁSICAS DE LA SINTAXIS DEL LENGUAJE

- Regla 1. Las instrucciones en JavaScript terminan en un punto y coma. Ejemplo:
- Regla 2. Uso de decimales en JavaScript. Los números en JavaScript que tengan decimales utilizarán el punto como separador de las unidades con la parte decimal. Ejemplos de números:
- Regla 3. Los literales se pueden escribir entre comillas dobles o simples. Ejemplo:
- Regla 4. Cuando sea necesario declarar una variable, se utilizará la palabra reservada var
- Regla 5. El operador de asignación, al igual que en la mayoría de lenguajes, es el símbolo igual (=).
- Regla 6. Se pueden utilizar los siguientes operadores aritméticos: (+-*/*). Ejemplo:

```
var s = "hola";
```

```
var x = 4;  
var pi = 3.14;
```

```
var s1 = "hola";  
var s2 = 'hola';
```

```
var x = (5*4)/2+1;
```


LAS 10 REGLAS BÁSICAS DE LA SINTAXIS DEL LENGUAJE

- Regla 7. En las expresiones, también se pueden utilizar variables. Ejemplo:

```
var t = 4;  
var x = (5*t)/2+1;  
var y;  
y = x * 2;
```

- Regla 8. Comentarios en JavaScript. Existen dos opciones para comentar el código:

a) // cuando se desea comentar el resto de la línea a partir de estas dos barras invertidas.
b) /* y */. todo lo contenido entre ambas etiquetas quedará comentado.

- Los identificadores en JavaScript comienzan por una letra o la barra baja (_) o el símbolo del dólar (\$).
- Regla 10. JavaScript es sensible a las mayúsculas y minúsculas (case-sensitive). Ejemplo:

```
var nombre = "Julio";  
var Nombre = "Ramón";
```

VARIABLES Y VALORES

- Una variable es un espacio reservado en la memoria que, como su nombre indica, puede cambiar de contenido a lo largo de la ejecución de un programa. Podemos almacenar un número, un texto, un listado de números, etcétera.

VALORES: A una variable a la cual se le asigna un valor al declarar se le dice variable inicializada

```
<script>
  //Declaración de variable ES5.
  var nombreVariable1;

  //Declaración de variable ES6.
  let  nombreVariable2;
  const LENGUAJE = "JAVASCRIPT";
</script>
```

TIPOS DE VALORES

- En una variable podemos asignar distintos tipos de valores, ya sea un número, un texto, o resultados de operaciones entre ambos.

```
<script>  
    let variableNumerica;  
    var variableTexto;  
  
    variableNumerica = 5;  
    variableTexto = "Mi texto";  
    variableTexto = 'Mi texto';  
</script>
```

¿Y CÓMO SABEMOS QUE VALORES TIENEN NUESTRAS VARIABLES?

- **console.log()**
- Para imprimir las variables y conocer su valor utilizamos una función ya incluida en JS llamada `console.log`;

```
let nombre = "Leia";
```

```
console.log(nombre);
```

Nótese que esta es una herramienta para que **desarrolladores/as** entiendan mejor el código. Jamás usaremos esto para mostrarle algo al usuario final.

¡A PRACTICAR!

- >> **Consigna:** Definí en un archivo JS variables con:

1. Tu nombre

2. Tu apellido

3. Tu edad

4. Si tienes mascota

Luego imprimí las variables en la consola.

OPERACIONES BÁSICAS

- Con variables de valores numéricos puedes realizar operaciones matemáticas: sumas, restas, multiplicaciones, etc.

```
<script>
    var    numeroA = 1;
    let    numeroB = 2;
    const  NUMEROC = 3;
    //Suma de dos números (1 + 2 = 3)
    let resultadoSuma = numeroA + numeroB;
    //Resta de dos números (2 - 1 = 1)
    let resultadoResta = numeroB - numeroA;
    //Producto de dos números (2 * 3 = 6)
    let resultadoProducto = numeroB * NUMEROC;
</script>
```

OPERACIONES BÁSICAS

```
<script>
  var  textoA = "FULL";
  let  textoB = "STACK";
  const BLANCO = " ";
  //Concatenar textoA y textoB ("FULL" + "STACK" = "FULL STACK")
  let resultadoA = textoA + textoB;
  //Concatenar textoB y 1 ("STACK" + 1 = "STACK1")
  let resultadoB = textoB + 1;
  //Concatenar textoA, BLANCO y textoB ("FULL" + " " + "STACK" = "FULL STACK")
  let resultadoC = textoA + BLANCO + textoB;
</script>
```

PROMPT, CONSOLA Y ALERT

127.0.0.1:5500 dice

Ingresa una letra

Aceptar Cancelar

Red {} Editor de estilo 🔊 Rendimiento 🗑 Memoria 📄 Almacenamiento

🗑 Salida del filtro

Mensaje de prueba

🌐 127.0.0.1:5500

¡Hola Mundo!

Aceptar

PROMPT

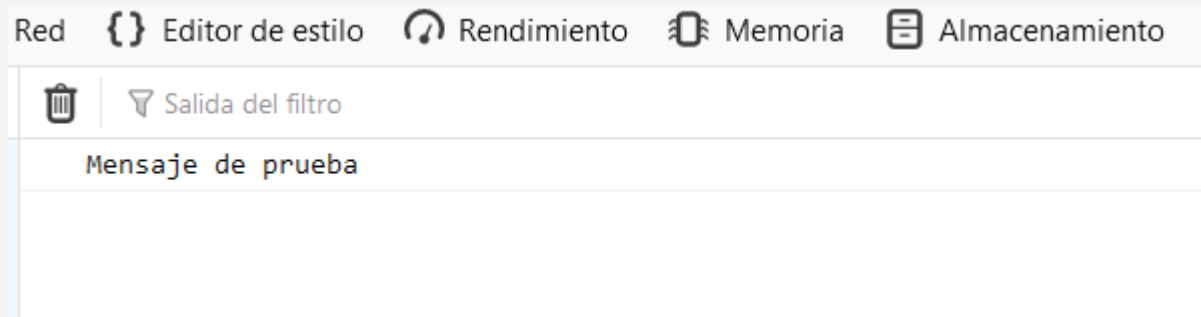
- La sentencia `prompt()` **mostrará un cuadro de diálogo para que el usuario ingrese un dato**. Se puede proporcionar un mensaje que se colocará sobre el campo de texto. El valor que devuelve es una cadena que representa lo que el usuario ingresó.

```
<script>  
  let nombreIngresado = prompt("Ingrese su nombre");  
</script>
```

CONSOLA

- La sentencia `console.log()` muestra el mensaje que pasemos como parámetro a la llamada en la consola JavaScript del Navegador web

```
<script>  
    console.log("Mensaje de prueba");  
</script>
```



ALERT

- La sentencia `alert()` **mostrará una ventana sobre la página** web que estemos accediendo mostrando el mensaje que se pase como parámetro a la llamada.

En la pantalla del navegador, el usuario verá una ventana sobre la web que muestra un mensaje. Al valor que mostramos al usuario como un resultado se lo conoce por el término de **salida**.

```
<script>  
    alert("¡Hola Mundo!");  
</script>
```

🌐 127.0.0.1:5500

¡Hola Mundo!

Aceptar

ALGORITMOS

- En programación, un algoritmo es un **conjunto de procedimientos o funciones ordenados** que se necesitan para realizar cierta operación o acción. Por ejemplo, en una suma el algoritmo implica tomar un dato, sumarlo a otro y obtener un resultado.

EJEMPLOS DE SCRIPT COMPLETO

- Este es un ejemplo de un Script JS corriendo en un archivo HTML.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mi primer App - FullStack</title>
    <script>
      let entrada = prompt("Ingresar una letra");
      let salida = entrada + " " + "ingresada";
      alert(salida);
    </script>
  </head>
  <body>
    <h2>Esta página contiene una app</h2>
  </body>
</html>
```

SI INGRESO “A”...

127.0.0.1:5500 dice

Ingresar una letra

OBTENGO...

127.0.0.1:5500 dice

A ingresada

EJEMPLOS DE PROMPT

- En la pantalla del navegador, el usuario verá una ventana sobre la web que le solicitará un dato. Al valor que el usuario ingresa se lo conoce por el término de **entrada**.



A screenshot of a web browser dialog box. The text inside the dialog reads: "127.0.0.1:5500 dice" followed by "Ingrese su nombre". Below this text is a text input field containing the text "ANA". At the bottom right of the dialog are two buttons: "Aceptar" (Accept) and "Cancelar" (Cancel).

EJERCICIO 1: CREAR UN ALGORITMO JS SIMPLE

- Crea un script en JS que le solicite al usuario ingresar los datos nombre, apellido y dirección y luego, mediante JavaScript, los muestre por consola.

EJERCICIO 2:

- Crea un script en JS que le solicite al usuario ingresar uno o más datos. Luego, con JavaScript, realiza operaciones matemáticas o de concatenación sobre las entradas teniendo en cuenta el tipo de dato. Al finalizar mostrar el resultados con `alert()` y `console.log()`

EJERCICIO 3:

- Pedir nombre mediante prompt y mostrarlo en consola junto con algún texto de saludo.
Ejemplo: ¡Hola, Juan! –
- Pedir un número mediante prompt, parsearlo, sumarlo a otro que se encuentre almacenado en una variable y luego mostrar el resultado en consola.
- Pedir un texto mediante prompt, luego otro, concatenarlos y mostrarlo en un alerta.