

Sintaxis y características

Ubicación del código

El código JavaScript se ubica en 2 lugares fundamentalmente:

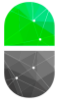
1. En un documento web integrándose con el código HTML.
2. En un archivo independiente con extensión .js



- **Integrado con HTML:** Tanto si la ubicación del código es en <head> o <body>, el código deberá ir entre las etiquetas `<script type="text/javascript">` y `</script>` como se muestra en las siguientes imágenes:

```
1 <html>
2   <head>
3     <title>Bienvenidos a mi web</title>
4     <script type="text/javascript">
5       //Código JavaScript
6     </script>
7   </head>
8   <body>
9
10
11 </body>
12 </html>
```

Código JavaScript
en el <head> de la
página web

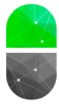


```
1 <html>
2   <head>
3     <title>Bienvenidos a mi web</title>
4   </head>
5
6   <body>
7     <script type="text/javascript">
8       //Código JavaScript
9     </script>
10
11   </body>
12 </html>
```

Código JavaScript
en el body de la
página web

- **En el <head>:** si ubicamos el código en la zona del <head> del documento web, conseguiremos que el código JavaScript se ejecute antes de cargar cualquier elemento de la página.
- **En el <body>:** al ubicarlo en la zona del <body> se cargará junto con el resto de elementos de la página web. El código JavaScript en el <body> se podrá repetir tantas veces como lo necesitemos.
- **En archivo independiente:** El código JavaScript se ubica en un archivo con extensión .js. Después este archivo se podrá linkar (enlazar) desde una página web. La ventaja de esta operativa es que si tenemos que incluir el mismo código JavaScript en varias páginas de un mismo sitio web, no es necesario repetir el código. Simplemente se va enlazando en la zona del <head> de las diferentes páginas tal y como se muestra en la siguiente imagen:

```
1 <html>
2   <head>
3     <title>Bienvenidos a mi web</title>
4     <script type="text/javascript" src="codigo_javascript.js"></script>
5   </head>
6
7   <body>
8
9   </body>
10 </html>
```

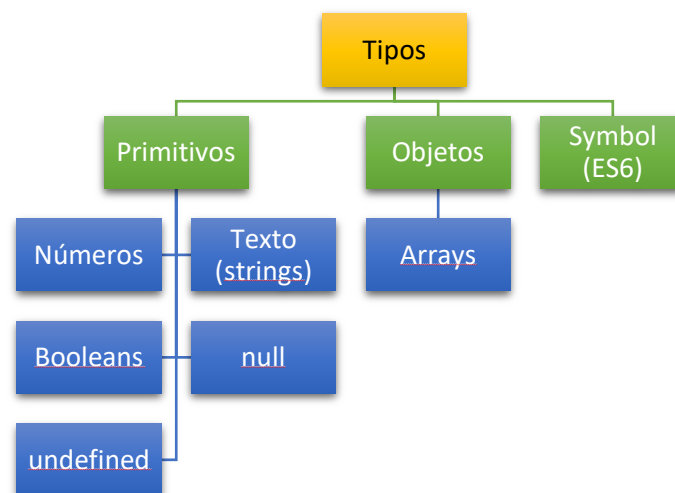


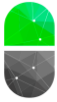
En este caso se agrega el atributo **“src”** en la etiqueta de apertura `<script>` en el cual deberemos indicar la ruta al archivo externo JavaScript con extensión `.js`. Lo más habitual es que este archivo se encuentre en el mismo directorio que la propia página web (como en el ejemplo de la imagen) o en una carpeta que por convención se la suele llamar **“js”** (aunque se le puede dar cualquier nombre).

Características y Sintaxis

- Características del lenguaje JavaScript:
 - Dinámico.
 - Alto nivel.
 - Interpretado.
 - Adaptado a POO.
 - Tipado débil.
 - Sintaxis similar a Java.
 - Se ejecuta tanto en FrontEnd como en BackEnd.
- Sintaxis:
 - Case-Sensitive
 - Ignora espacio vertical
 - Uso de comentarios
 - `//` Una línea
 - `/*.....*/` varias líneas
 - Posee literales
 - Posee identificadores y palabras reservadas
 - Admite caracteres UNICODE
 - Las sentencias deben terminar en punto y coma

Tipos de valores, variables y constantes





- Variables

- ¿Qué son?
 - Espacio en la memoria del ordenador donde se almacenará un valor que podrá cambiar (variar) durante la ejecución de un programa.
- ¿Qué es declarar una variable?
 - Especificar su nombre y ocasionalmente el valor que almacenaremos en su interior.
 - 2 tipos de declaración
 - Implícita: sin utilizar “var”, “let” o “const” en la declaración (ver imagen)

```
nombre="Juan";  
apellido="Díaz";  
document.write("El nombre del usuario es: " + nombre + " " + apellido);
```

declaración implícita de variables
"nombre" y "apellido"

- Explícita: utilizando “var”, “let” o “const” en la declaración (ver imagen)

```
let nombre="Juan";  
let apellido="Díaz";  
document.write("El nombre del usuario es: " + nombre + " " + apellido);
```

Declaración explícita de variables
"nombre" y "apellido"

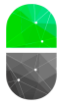
Se recomienda la declaración explícita de variables. La diferencia entre utilizar “var”, “let” y “const” en lo referente a alcances de variables se explicará más adelante en el curso.

- Constantes

- ¿Qué son?
 - Espacio en la memoria del ordenador donde se almacenará un valor que **NO podrá cambiar** (variar) durante la ejecución de un programa.
- ¿Qué es declarar una constante?
 - Especificar su nombre y valor que almacenaremos en su interior. Al contrario de lo que ocurre con las variables, en el caso de las constantes es obligatorio especificar en el momento de la declaración el valor almacenado en su interior
 - Ejemplo de declaración de constante:

```
const iva=0.21;  
let importe_producto=150+(150*iva);  
document.write("El importe a pagar es: " + importe_producto);
```

declaración de constante



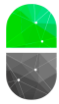
Operadores en JavaScript

- Asignación -> **=**
- Aritméticos
 - Suma -> **+**
 - Resta -> **-**
 - Multiplicación -> *****
 - División -> **/**
- Incremento y decremento
 - Incremento -> **++**
 - Decremento -> **--**
- Resto o módulo -> **%**
- Aritméticos abreviados
 - Suma -> **+=**
 - Resta -> **-=**
 - Multiplicación -> ***=**
 - División -> **/=**
- Concatenación de cadenas -> **+**
- Operadores lógicos (detallados a continuación)

- Operadores lógicos

Estos operadores nos permitirán construir las pruebas lógicas (condiciones) en condicionales y bucles, además de ser utilizados en numerosos escenarios a la hora de programar. En JavaScript disponemos de los siguientes:

- Y Lógico -> **&&**
- O Lógico -> **||**
- Operador NO/Negación -> **!**
- Operadores de comparación
 - Igualdad -> **==**
 - Igualdad estricta -> **===**
 - Desigualdad estricta -> **!==**
 - Mayor que -> **>**
 - Menor que -> **<**
 - Mayor o igual que -> **>=**
 - Menor o igual que -> **<=**



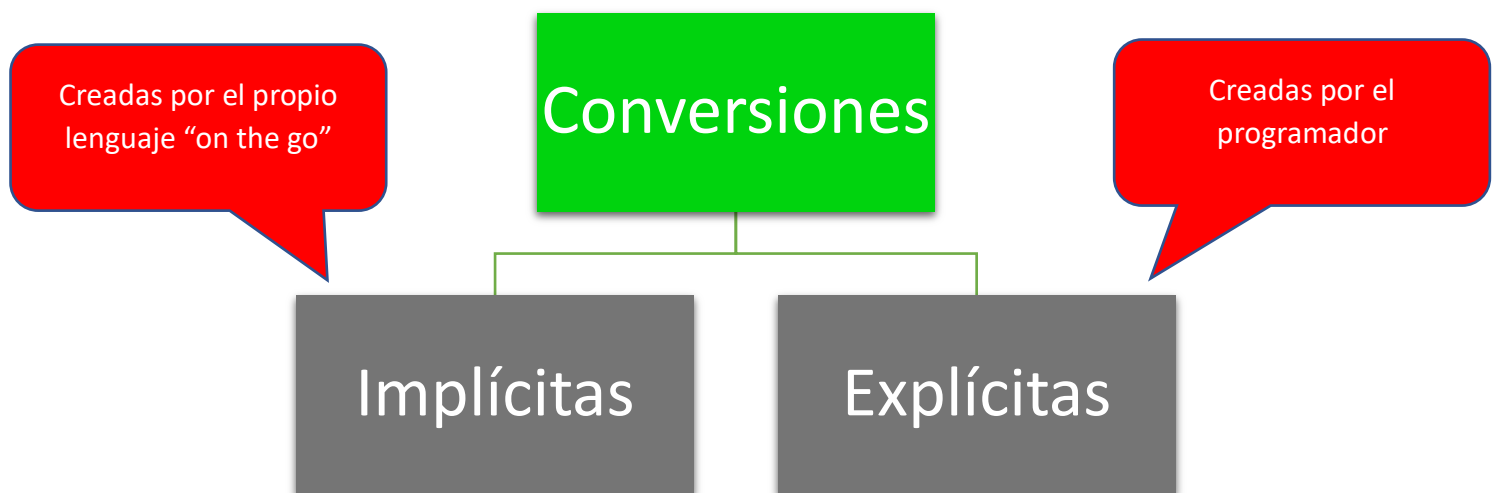
- Otros Operadores

Operadores para realizar otro tipo de operaciones en JavaScript y que se verán y detallarán más adelante.

- Operador agrupador
- Operador void
- Operador typeof
- Operador ternario

Conversiones de tipos

Las conversiones surgen de la necesidad de convertir un tipo de dato en otro a la hora de programar.



- Ejemplo de conversión implícita:

```
let a=7;

let b="5";

document.write(a*b);
```

En este ejemplo la variable "a" es de tipo **numérico** y la variable "b" es de tipo **texto**. Sin embargo, el programa realiza la operación (multiplicación) correctamente ya que de forma "**implícita**" convierte la variable "b" de tipo texto a tipo numérico.



- Ejemplo de conversión explícita:

```
let a=7;  
  
let b=parseInt("5");  
  
document.write(a+b);
```

En este ejemplo se realiza una conversión explícita de la variable "b" utilizando para ello la función `parseInt()`. Si no se realiza esta conversión, Python concatenará el valor de las variables en vez de sumar, es decir, sin conversión explícita el resultado sería "75" y tras la conversión el resultado es "12".