

JavaScript

Las buenas partes

Recomendación para leer luego.

JavaScript: The Good Parts
(Douglas Crockford)

<http://shop.oreilly.com/product/9780596517748.do>



¿Qué es programar?

¿Qué es programar?

"Programar"

No es únicamente saber cómo es la estructura de un lenguaje de programación.

No es conocer la sintaxis.

No es saber dónde poner unas comillas o un punto y coma.

¿Qué es programar?

Es tener un modelo analítico de pensamiento diferente.

Es no trabajar con "Prueba y error".

Sólo necesitamos conocer:

- VARIABLES
- CONDICIONES
- CICLOS
- FUNCIONES
- OBJETOS
- EVENTOS

Programar

¿Qué necesitamos para programar?

Programar

Lenguaje de programación:

Lenguaje > Compilador > Ejecución

Programar

Lenguaje de programación:

Lenguaje > Compilador > Ejecución

El lenguaje es en dónde programamos.

Programar

Lenguaje de programación:

Lenguaje > Compilador > Ejecución

El lenguaje es en dónde programamos.

Lenguaje > Compilador > Ejecución

El compilador toma el lenguaje y lo convierte a código entendible por procesadores.

Programar

Lenguaje de programación:

Lenguaje > Compilador > Ejecución

El lenguaje es en dónde programamos.

Lenguaje > Compilador > Ejecución

El compilador toma el lenguaje y lo convierte a código entendible por procesadores.

Lenguaje > Compilador > Ejecución

La ejecución se da dentro de un SO.

En Windows sería un .exe

(En Linux y Mac son sólo ejecutables sin ninguna extensión)

Programar

Lenguaje de programación:

Lenguaje > Compilador > Ejecución

El lenguaje es en dónde programamos.

Lenguaje > Compilador > Ejecución

El compilador toma el lenguaje y lo convierte a código entendible por procesadores.

Lenguaje > Compilador > Ejecución

La ejecución se da dentro de un SO.

En Windows sería un .exe

(En Linux y Mac son sólo ejecutables sin ninguna extensión)

Por ejemplo:

C++	.net (Windows)	.exe
	GCC (Linux)	

HTML No es un lenguaje.

Es un contenedor estructural en donde va el lenguaje de programación.

Dentro del HTML vamos a incluir el lenguaje de programación que vamos a comenzar a conocer:

HTML



JS



JavaScript

JavaScript No es Java.

JavaScript es el estandar para el desarrollo de aplicaciones web.

JavaScript

JavaScript No tiene un compilador.

JavaScript utiliza una máquina virtual.

Una máquina virtual es una porción de código que convierte un lenguaje y lo ejecuta sin que se transforme en un archivo binario.

JavaScript

Cada navegador tiene una máquina virtual diferente. Es su motor de renderizado que transforma el código HTML y JS a elementos visuales.

Safari > webkit
Chrome > Blink
IE > Trident

JavaScript

La ejecución ocurre en un navegador:

JavaScript

¿Cómo comenzamos?

```
js.html
<!DOCTYPE html>
<html lang="es-ar">
<head>
  <meta charset="UTF-8">
  <title>JS</title>
  <script type="text/javascript"></script>
</head>
<body>

</body>
</html>
```

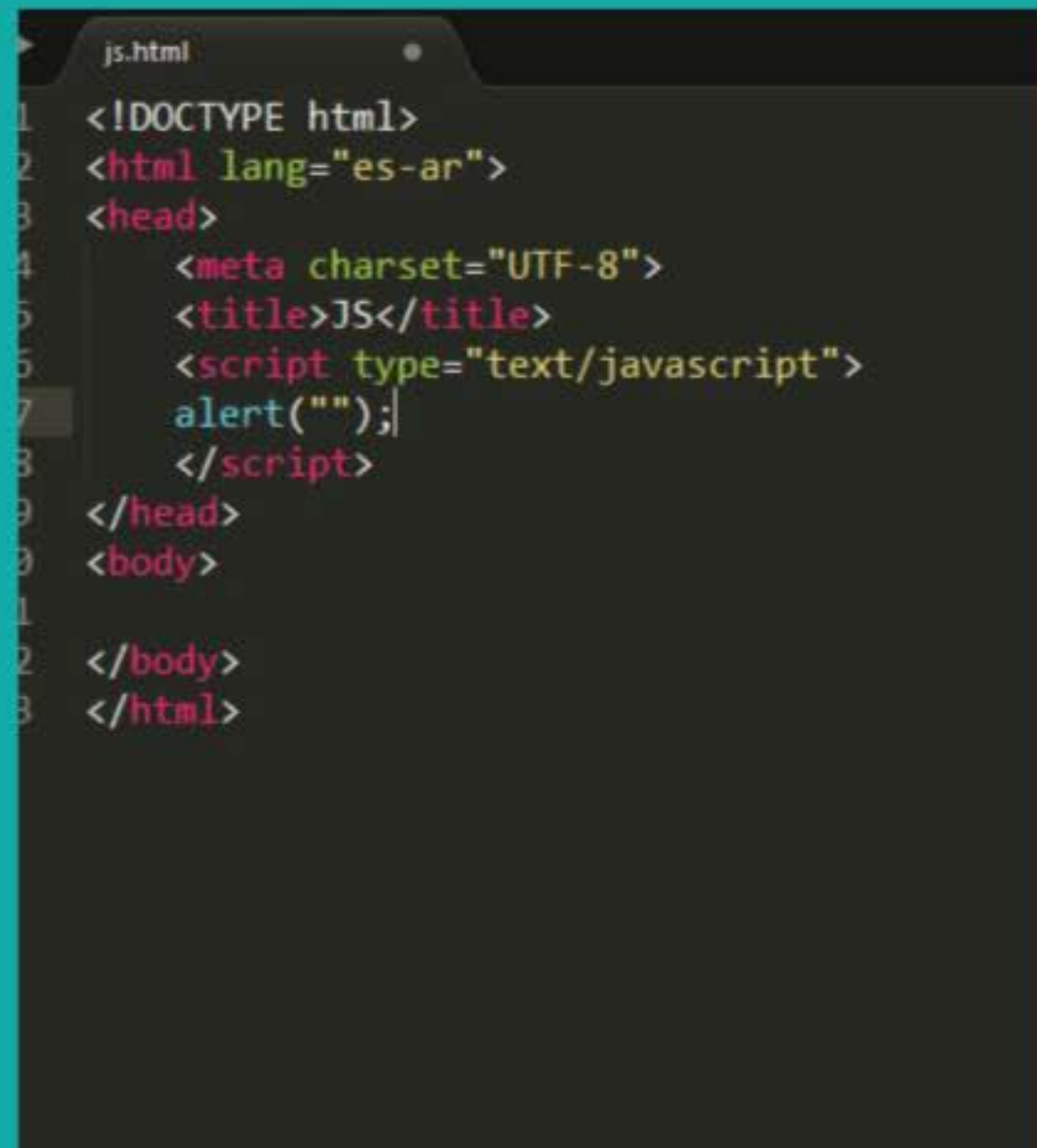
script es la etiqueta

type es el atributo

"text/javascript" es tipo de código

JavaScript

¿Cómo comenzamos?



A screenshot of a code editor window titled 'js.html'. The editor shows a basic HTML document structure with a JavaScript alert function. The code is as follows:

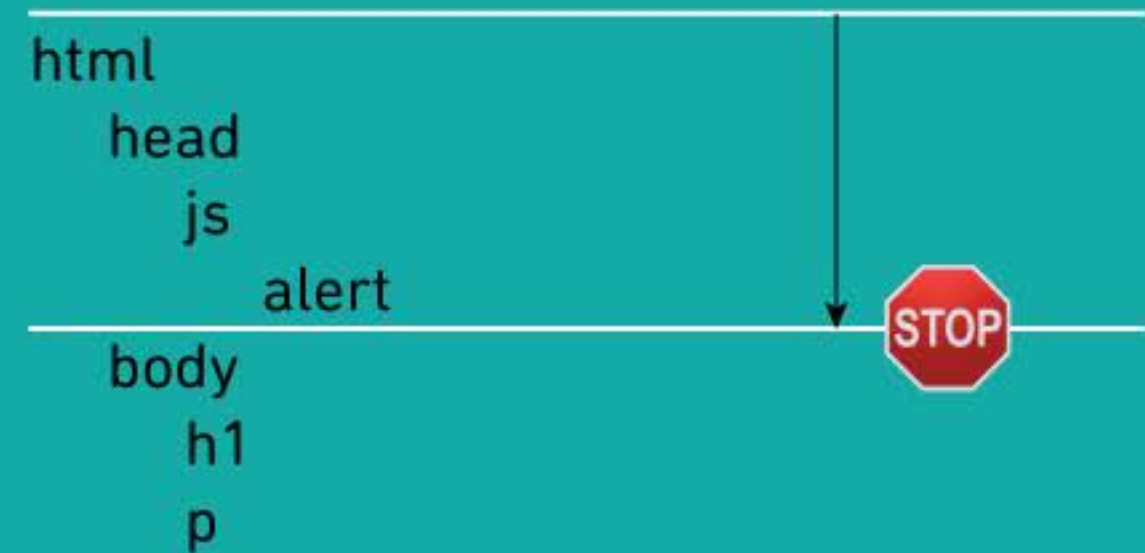
```
1 <!DOCTYPE html>
2 <html lang="es-ar">
3 <head>
4   <meta charset="UTF-8">
5   <title>JS</title>
6   <script type="text/javascript">
7     alert("");
8   </script>
9 </head>
10 <body>
11
12 </body>
13 </html>
```


JavaScript

¿Cómo comenzamos?

```
<!DOCTYPE html>
<html lang="es-ar">
<head>
  <meta charset="UTF-8">
  <title>JS</title>
  <script type="text/javascript">
    alert("Ya sé programar!");
  </script>
</head>
<body>
  <h1>JavaScript</body>
  <p>JavaScript dentro de HTML</p>
</html>
```

El código JS está ubicado de esta forma:

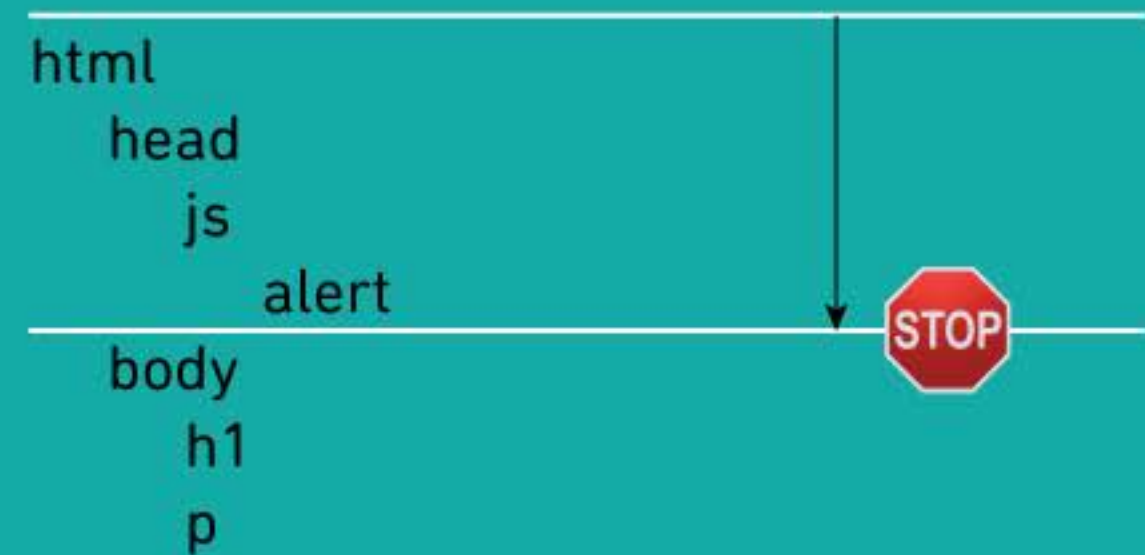


JavaScript

¿Cómo comenzamos?

```
<!DOCTYPE html>
<html lang="es-ar">
<head>
  <meta charset="UTF-8">
  <title>JS</title>
  <script type="text/javascript">
    alert("Ya sé programar!");
  </script>
</head>
<body>
  <h1>JavaScript</body>
  <p>JavaScript dentro de HTML</p>
</html>
```

El código JS está ubicado de esta forma:



Como ven, podemos frenar la ejecución de una web en el navegador usando programación.

JavaScript

¿Cómo comenzamos?

```
1 <!DOCTYPE html>
2 <html lang="es-ar">
3 <head>
4   <meta charset="UTF-8">
5   <title>JS</title>
6   <script type="text/javascript">
7     //alert: Función
8     // (): Parámetros de la función
9     // "": Cadena de texto
10    alert("Ya sé programar!");
11  </script>
12 </head>
13 <body>
14   <h1>JavaScript</body>
15   <p>JavaScript dentro de HTML</p>
16 </html>
```

alert es una función. Código empaquetado al cual podemos llamar en cualquier momento.

() Dentro de los paréntesis van los parámetros de la función.

"" Cadena de texto. La memoria RAM funciona en bloques y en cada bloque sólo puede entrar una letra (carácter). Esos bloques se denominan bytes (con un rango de número que va del 0 al 255). En el pasado, se creaba el texto encadenando bytes

JavaScript

¿Cómo comenzamos?

```
1 <!DOCTYPE html>
2 <html lang="es-ar">
3 <head>
4   <meta charset="UTF-8">
5   <title>JS</title>
6   <script type="text/javascript">
7     //alert: Función
8     // (): Parámetros de la función
9     // "": Cadena de texto
10    alert(0123456789);
11  </script>
12 </head>
13 <body>
14   <h1>JavaScript</body>
15   <p>JavaScript dentro de HTML</p>
16 </html>
```

¿Qué pasa si ponemos números dentro de los parámetros?

JavaScript

¿Cómo comenzamos?

```
1 <!DOCTYPE html>
2 <html lang="es-ar">
3 <head>
4   <meta charset="UTF-8">
5   <title>JS</title>
6   <script type="text/javascript">
7     //alert: Función
8     // (): Parámetros de la función
9     // "": Cadena de texto
10    alert(6*6);
11  </script>
12 </head>
13 <body>
14   <h1>JavaScript</body>
15   <p>JavaScript dentro de HTML</p>
16 </html>
```

¿Qué pasa si ponemos una fórmula matemática?

JavaScript

¿Cómo comenzamos?

```
1 <!DOCTYPE html>
2 <html lang="es-ar">
3 <head>
4   <meta charset="UTF-8">
5   <title>JS</title>
6   <script type="text/javascript">
7     //alert: Función
8     // (): Parámetros de la función
9     // "": Cadena de texto
10    alert("6*6");
11  </script>
12 </head>
13 <body>
14   <h1>JavaScript</body>
15   <p>JavaScript dentro de HTML</p>
16 </html>
```

¿Qué pasa si ponemos una fórmula matemática entre comillas?

JavaScript

¿Cómo comenzamos?

```
1 <!DOCTYPE html>
2 <html lang="es-ar">
3 <head>
4   <meta charset="UTF-8">
5   <title>JS</title>
6   <script type="text/javascript">
7     //alert: Función
8     // (): Parámetros de la función
9     // "": Cadena de texto
10    alert("666"+1);
11  </script>
12 </head>
13 <body>
14   <h1>JavaScript</body>
15   <p>JavaScript dentro de HTML</p>
16 </html>
```

¿Qué pasa si ponemos un número entre comillas y le sumamos otro?

JavaScript

¿Cómo comenzamos?

```
1 <!DOCTYPE html>
2 <html lang="es-ar">
3 <head>
4   <meta charset="UTF-8">
5   <title>JS</title>
6   <script type="text/javascript">
7     //alert: Función
8     // (): Parámetros de la función
9     // "": Cadena de texto
10    alert("666"+1);
11  </script>
12 </head>
13 <body>
14   <h1>JavaScript</body>
15   <p>JavaScript dentro de HTML</p>
16 </html>
```

¿Qué pasa si ponemos un número entre comillas y le sumamos otro?

Lo que pasó acá es que se **concatenaron** tipos de datos. Se "pegan".

Por un lado tomó al "666" como caracteres y le pegó el 1 al final.

Si ordenamos los parámetros al revés, el resultado será el mismo.

```
alert(1+"666");
```

Resultado = 1666

JavaScript

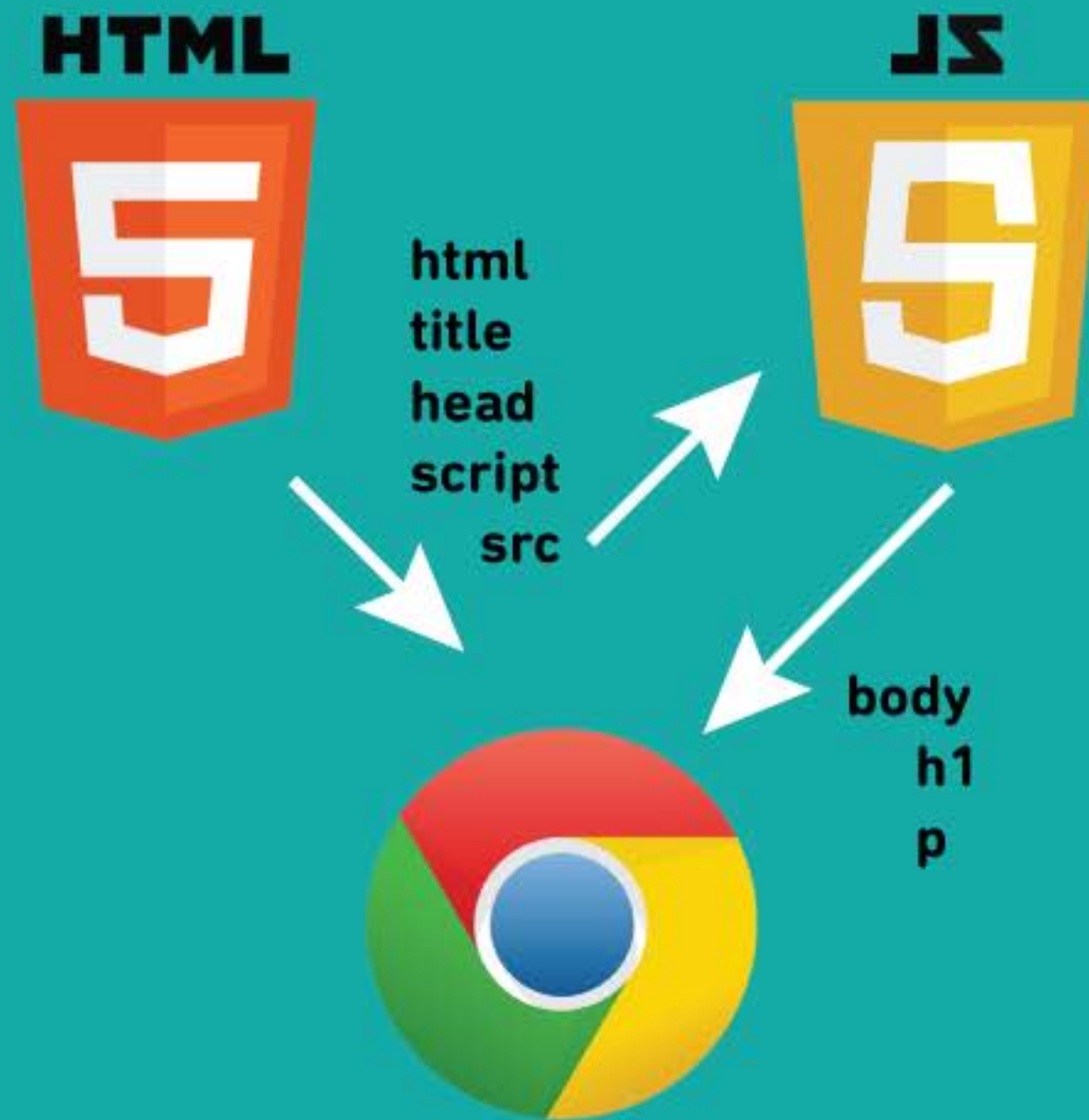
Buena práctica

```
<!DOCTYPE html>
<html lang="es-ar">
<head>
  <meta charset="UTF-8">
  <title>JS</title>
  <script type="text/javascript" src="
    codigo.js">


  </script>
</head>
<body>
  <h1>JavaScript</body>
  <p>JavaScript dentro de HTML</p>
</html>
```

Tener separado el HTML del JavaScript es una buena práctica. Para ello tenemos que referenciar un documento .js en nuestro .html

JavaScript



JavaScript



```
js.html x codigo.js
1 //alert: Función
2 // (): Parámetros de la función
3 // ": Cadena de texto
4 var nombre = "Marcelo";
5 var apellido = "Tinelli";
6 var edad = 45;
```

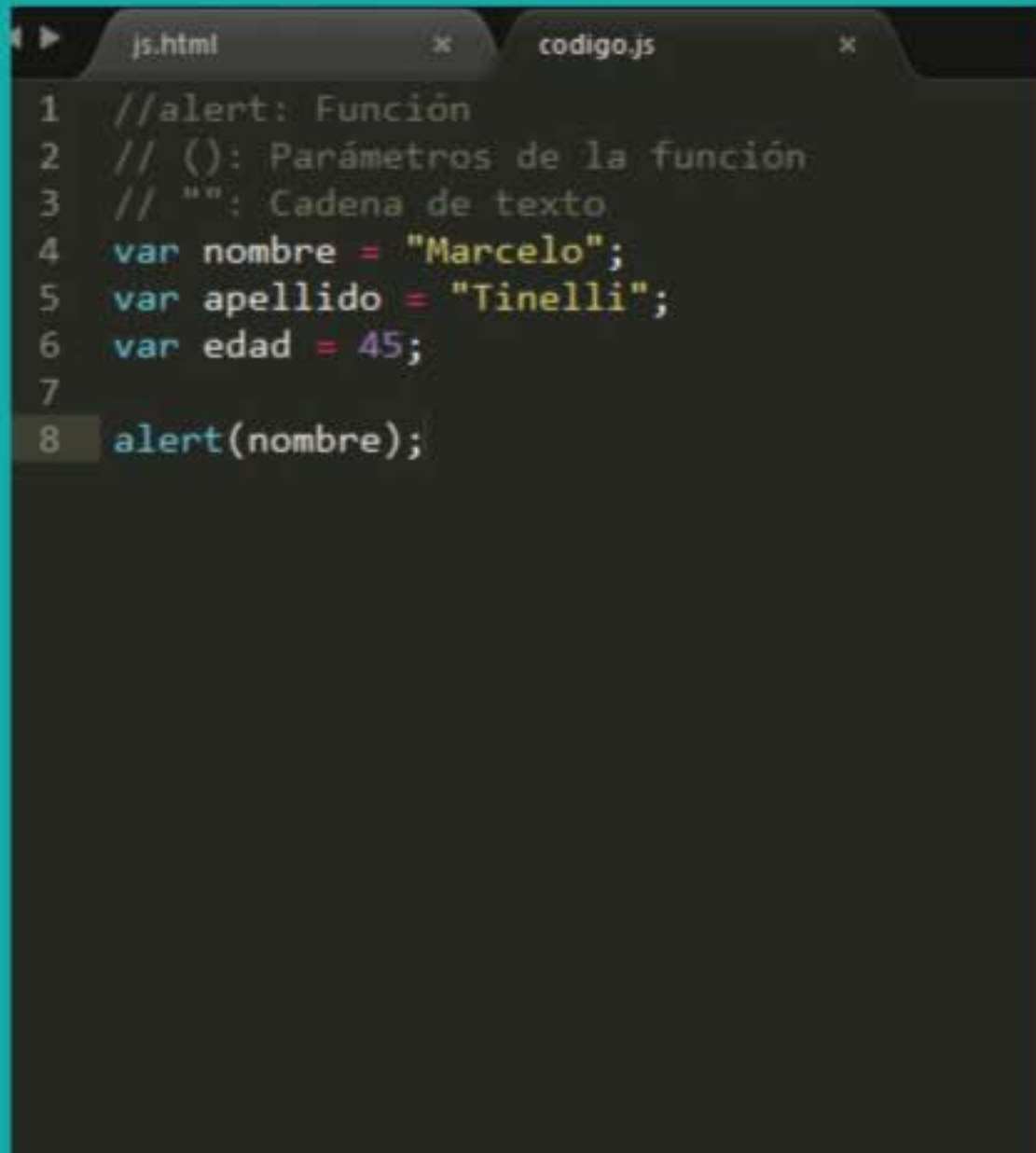
var: es una instrucción nativa de JavaScript y nos permite declarar variables.

nombre: cómo le nombramos a nuestra variable.

"": El valor de nuestra variable.

Las variables son "cajas" en las cuales podemos guardar cosas.

JavaScript

A screenshot of a code editor with two tabs: 'js.html' and 'codigo.js'. The 'codigo.js' tab is active, showing a JavaScript script. The code includes comments in Spanish explaining the 'alert' function and its parameters, followed by variable declarations for 'nombre', 'apellido', and 'edad', and a final 'alert' call using the 'nombre' variable.

```
1 //alert: Función
2 // (): Parámetros de la función
3 // " ": Cadena de texto
4 var nombre = "Marcelo";
5 var apellido = "Tinelli";
6 var edad = 45;
7
8 alert(nombre);
```

JavaScript

A screenshot of a code editor with two tabs: 'js.html' and 'codigo.js'. The 'codigo.js' tab is active and shows the following JavaScript code:

```
1 //alert: Función
2 // (): Parámetros de la función
3 // "": Cadena de texto
4 var nombre = "Marcelo";
5 var apellido = "Tinelli";
6 var edad = 45;
7
8 alert(nombre + apellido);
```

JavaScript



```
1 //alert: Función
2 // (): Parámetros de la función
3 // "": Cadena de texto
4 var nombre = "Marcelo";
5 var apellido = "Tinelli";
6 var edad = 45;
7
8 alert(nombre + " " + apellido + " " + edad + " años.");
```

JavaScript



The image shows a code editor with two tabs: 'js.html' and 'codigo.js'. The 'codigo.js' tab is active, displaying the following JavaScript code:

```
1 //alert: Función
2 // (): Parámetros de la función
3 // "": Cadena de texto
4 var nombre = "Marcelo";
5 var apellido = "Tinelli";
6 var edad = 45;
7
8 alert(nombre + " " + apellido + " " + edad + " años.");
```

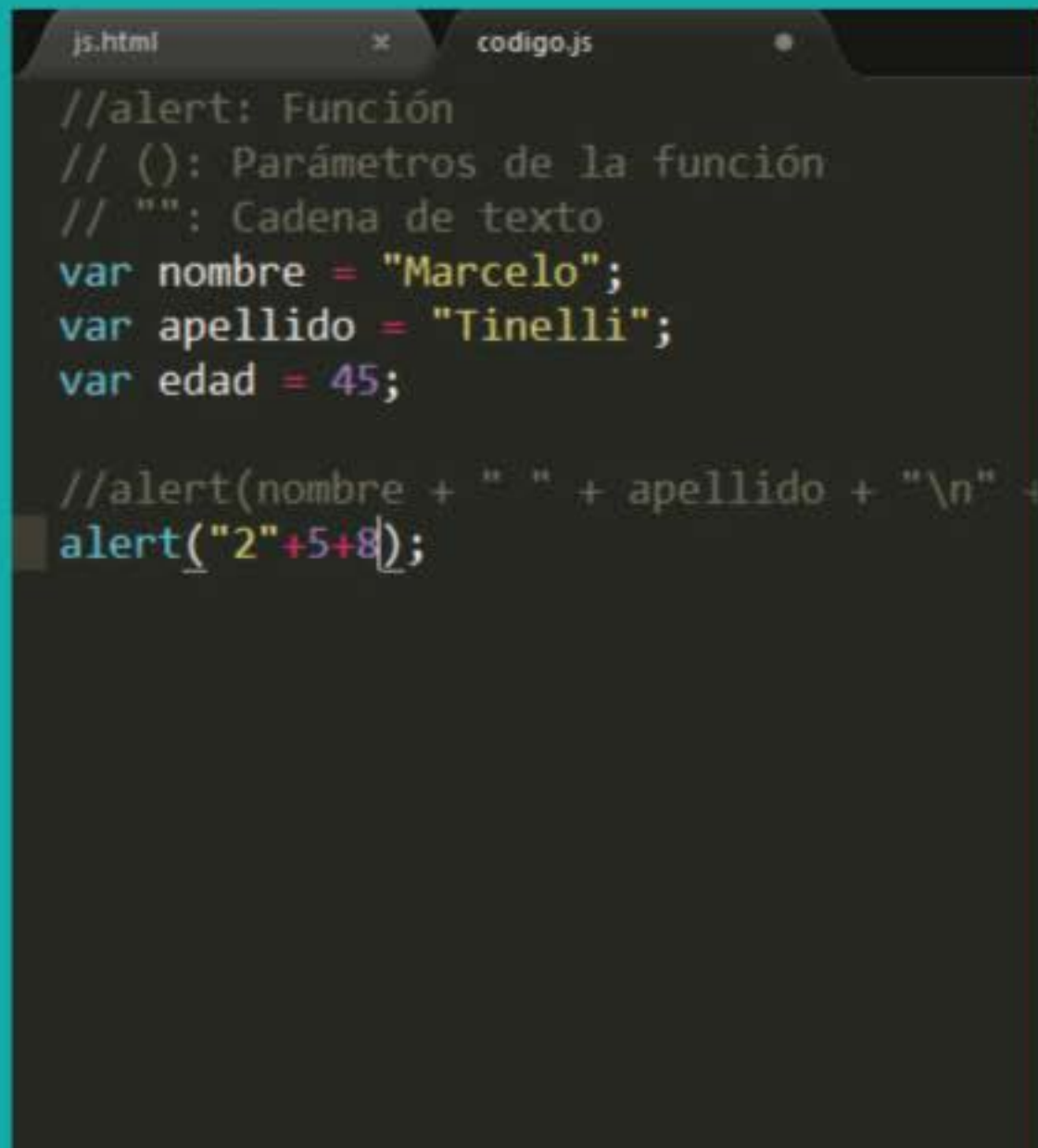

JavaScript



```
js.html x codigo.js
1 //alert: Función
2 // (): Parámetros de la función
3 // "": Cadena de texto
4 var nombre = "Marcelo";
5 var apellido = "Tinelli";
6 var edad = 45;
7
8 //alert(nombre + " " + apellido + "\n");
9 alert(5+8+"2");
```

¿Qué nos va a mostrar el alert?

JavaScript

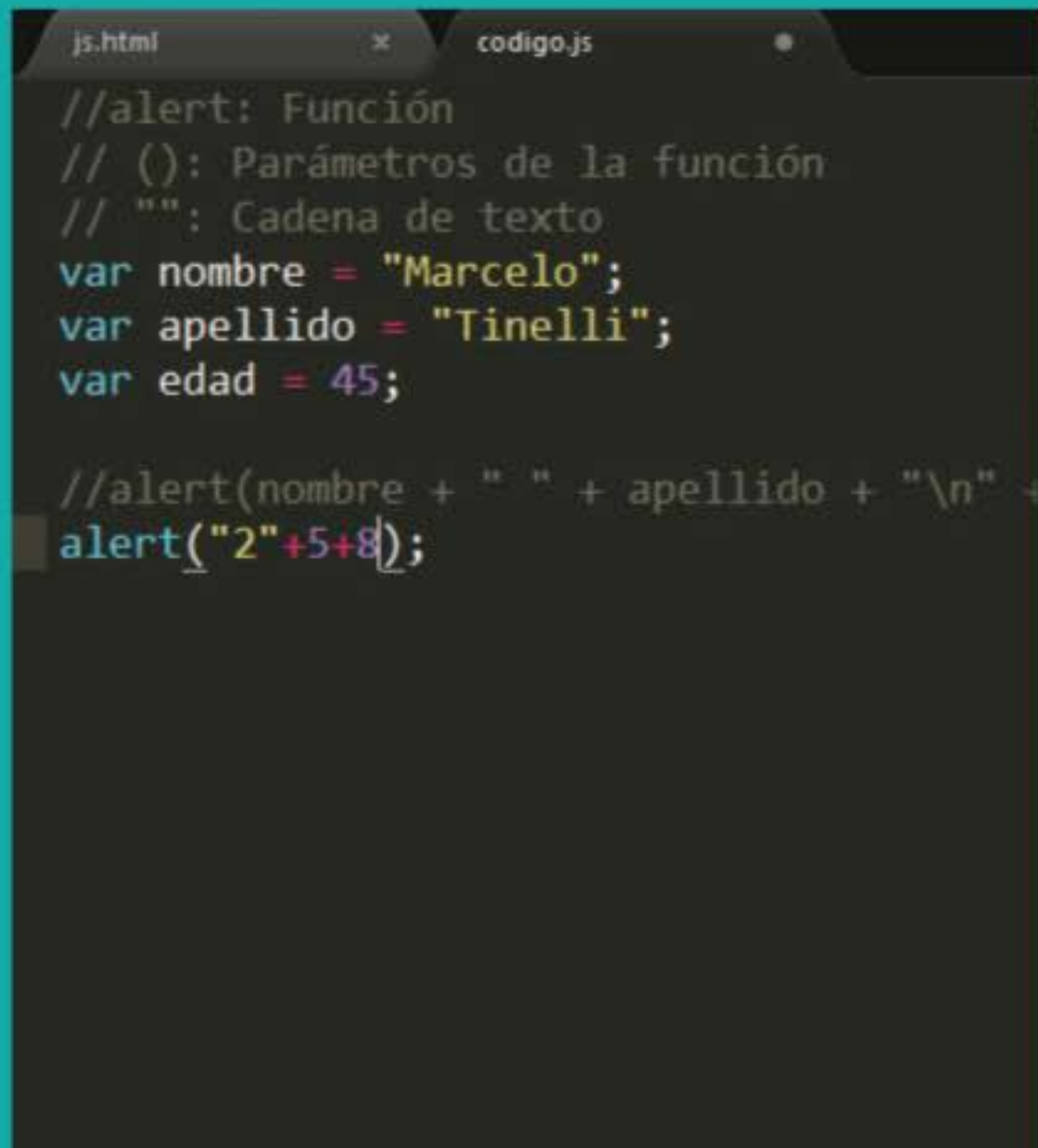


```
js.html x código.js
//alert: Función
// (): Parámetros de la función
// "": Cadena de texto
var nombre = "Marcelo";
var apellido = "Tinelli";
var edad = 45;

//alert(nombre + " " + apellido + "\n" +
alert("2"+5+8);
```

¿En este caso qué nos va a mostrar el alert?

JavaScript



```
js.html x código.js
//alert: Función
// (): Parámetros de la función
// "": Cadena de texto
var nombre = "Marcelo";
var apellido = "Tinelli";
var edad = 45;

//alert(nombre + " " + apellido + "\n" +
alert("2"+5+8);
```

¿En este caso qué nos va a mostrar el alert?

Siempre cuando primero declaramos un parámetro con cadena de texto, va a concatenar todo, tomando lo que sigue como otras cadenas de texto.

JavaScript

```
js.html x codigo.js
1 //alert: Función
2 // (): Parámetros de la función
3 // "": Cadena de texto
4 var nombre = "Marcelo";
5 var apellido = "Tinelli";
6 var edad = 45;
7
8 //alert(nombre + " " + apellido + "\n");
9 alert("2"+5*8);
```

¿Qué pasa si multiplicamos?

JavaScript

```
js.html x código.js
1 //alert: Función
2 // (): Parámetros de la función
3 // "": Cadena de texto
4 var nombre = "Marcelo";
5 var apellido = "Tinelli";
6 var edad = 45;
7
8 //alert(nombre + " " + apellido + "\n");
9 alert("2"+5*8);
```

¿Qué pasa si multiplicamos?

La multiplicación y la división tiene más peso que la suma y la resta.

Demasiado



Paremos hoy acá...