



Ministerio de
**TRABAJO, EMPLEO
Y SEGURIDAD SOCIAL**



*Paraguay
de la gente*

Introducción a Javascript

- SINTAXIS

Definición

JavaScript (JS) es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo (just-in-time) con funciones de primera clase. Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, y es usado en muchos entornos fuera del navegador, tal como Node.js, Apache CouchDB and Adobe Acrobat. JavaScript es un lenguaje de programación basado en prototipos, multi-paradigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa (por ejemplo programación funcional). Lea más en [acerca de JavaScript](#).

Características

Conozcamos ahora las características de JavaScript que haces de este lenguaje, uno de los más populares en la actualidad.

1. Es Liviano.
2. Multiplataforma, ya que se puede utilizar en Windows, Linux o Mac o en el navegador de tu preferencia.
3. Es Imperativo y estructurado, mediante un conjunto de instrucciones indica al computador qué tarea debe realizar.
4. Prototipado, debido a que usa prototipos en vez de clases para el uso de herencia.
5. Orientado a objetos y eventos.
6. Es Interpretado, no se compila para poder ejecutarse.

Sintaxis

La sintaxis de JavaScript es muy similar a la de otros lenguajes de programación como Java y C. Las normas básicas que definen la sintaxis de JavaScript son las siguientes:

- **No se tienen en cuenta los espacios en blanco y las nuevas líneas**
- **Se distinguen las mayúsculas y minúsculas**
- **No se define el tipo de las variables**
- **No es necesario terminar cada sentencia con el carácter de punto y coma (;)**
- **Se pueden incluir comentarios**

Comentarios

JavaScript define dos tipos de comentarios: los de una sola línea y los que ocupan varias líneas.

❖ Ejemplo de comentario de una sola línea:

```
// a continuación se muestra un mensaje  
alert("mensaje de prueba");
```

Los comentarios de una sola línea se definen añadiendo dos barras oblicuas (//) al principio de la línea.

❖ Ejemplo de comentario de varias líneas:

```
/* Los comentarios de varias líneas son muy útiles  
cuando se necesita incluir bastante información  
en los comentarios */  
alert("mensaje de prueba");
```

Tipo dinámico

JavaScript es un lenguaje de *tipado débil* o *dinámico*. Esto significa que no es necesario declarar el tipo de variable antes de usarla. El tipo será determinado automáticamente cuando el programa comience a ser procesado. Esto también significa que puedes tener la misma variable con diferentes tipos:

```
var foo = 42; // foo es ahora un Number
```

```
var foo = "bar"; // foo es ahora un String
```

```
var foo = true; // foo es ahora un Boolean
```

Tipos de Datos

La última definición del estándar ECMAScript define siete tipos de datos:

Seis tipos de datos que son primitivos

1. Undefined: En JavaScript, una variable sin valor tiene el valor `undefined`. El tipo también está `undefined`
2. Boolean
3. Null
4. Number
5. String
6. Symbol (nuevo en ECMAScript 6)
7. y Object

1. Tipo de dato Number(numérico)

Los números se escriben con o sin decimales

Ejemplo:

Archivo codigo.js	Archivo index.html	Vista Navegador
<pre>var x; // Ahora x es undefined x = 10.5; // Ahora x es numerico document.getElementById("de mo").innerHTML = x;</pre>	<pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="wi dth=device-width, initial-scale=1.0"> <title>Document</title> </head> <body> <h2>JavaScript Numeros</h2> <p>Los números se escriben con o sin decimales.</p> <p id="demo"></p> <script src="codigo.js"></script> </body> </html></pre>	<div>JavaScript Numeros</div> <div>Los números se escriben con o sin decimales.</div> <div>10.5</div>

2. Strings(Cadenas)

Las cadenas son textos que pueden ser escrito entre comillas dobles o simples

Archivo codigo.js	Archivo index.html	Vista Navegador
<pre>var x; // Ahora x es undefined x = 'Juan Perez';// Ahora x es String document.getElementById("demo").innerHTML = x;</pre>	<pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Document</title> </head> <body> <h2>JavaScript Strings</h2> <p>Las cadenas son textos que pueden ser escrito entre comillas dobles o simples </p> <p id="demo"></p> <script src="codigo.js"></script> </body> </html></pre>	<div>JavaScript Strings</div> <div>Las cadenas son textos que pueden ser escrito entre comillas dobles o simples</div> <div>Juan Perez</div>

3. Booleans

Booleans pueden ser: true o false

Archivo codigo.js	Archivo index.html	Vista Navegador
<pre>var x = 5; var y = 5; var z = 6; document.getElementById("demo").innerHTML = (x == y) + "
" + (x == z);</pre>	<pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Document</title> </head> <body> <h2>JavaScript Booleans</h2> <p>Booleans pueden ser: true o false:</p> <p id="demo"></p> <script src="codigo.js"></script> </body> </html></pre>	<div><h3>JavaScript Booleans</h3><p>Booleans pueden ser: true o false:</p><p>true false</p></div>

4. Objects

Los objetos de JavaScript se escriben con llaves {}.

Las propiedades de los objetos se escriben como **nombre: valores**, separados por comas.

Archivo codigo.js	Archivo index.html	Vista Navegador
<pre>var persona = { nombre : "Juan", apellido : "Perez", edad : 50, colordeojos : "marron" }; document.getElementById("demo").innerHTML = persona.nombre + " tiene " + persona.edad + " años.";</pre>	<pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Document</title> </head> <body> <h2>JavaScript Objects</h2> <p id="demo"></p> <script src="codigo.js"></script> </body> </html></pre>	<div>JavaScript Objects</div> <div>Juan tiene 50 años.</div>

5.Null

En JavaScript null es "nada". Se supone que es algo que no existe. Desafortunadamente, en JavaScript, el tipo de datos nulo es un objeto.

Puede considerar un error en JavaScript que typeof null es un objeto. Debería ser nulo.

Puede vaciar un objeto configurándolo como null.

Archivo codigo.js	Archivo index.html	Vista Navegador
<pre>var persona = {nombre:"Juan", apellido:"Perez", edad:50, colordeojos:"marron"}; persona = null; document.getElementById("de mo").innerHTML = typeof persona;</pre>	<pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="wi dth=device-width, initial-scale=1.0"> <title>Document</title> </head> <body> <h2>JavaScript</h2> <p>Los objetos se pueden vaciar estableciendo el valor en null.</p> <p id="demo"></p> <script src="codigo.js"></script> </body> </html></pre>	<div><h3>JavaScript</h3><p>Los objetos se pueden vaciar estableciendo el valor en null.</p><p>object</p></div>

6. Arrays

Los arrays de JavaScript se escriben entre corchetes.

Los elementos del array están separados por comas.

Archivo codigo.js	Archivo index.html	Vista Navegador
<pre>var automoviles = ["Saab","Volvo","BMW"]; document.getElementById("de mo").innerHTML = automoviles[0];</pre>	<pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="widt h=device-width, initial-scale=1.0"> <title>Document</title> </head> <body> <h2>JavaScript Arrays</h2> <p>Los índices del array están basados en zero-based, lo que significa que el primer elemento es [0].</p> <p id="demo"></p> <script src="codigo.js"></script> </body> </html></pre>	

JavaScript Arrays

Los índices del array están basados en zero-based, lo que significa que el primer elemento es [0].

Saab

Algunas funciones estándar en javascript

1. Función alert

Esta función es un método del objeto Window y es una de las más utilizadas al momento de iniciarnos en JavaScript, pues es la encargada de mostrar una pequeña ventana de aviso en la pantalla, así, si se requiere que aparezca un mensaje cuando ocurra determinada acción en el programa, podemos hacer uso de esta función. La función alert recibe como parámetro el mensaje que se debe mostrar en la ventana.

Sintaxis:

`alert(mensaje a mostrar)`

A continuación se muestra un ejemplo donde aparece en la pantalla una pequeña ventana con un mensaje de saludo:

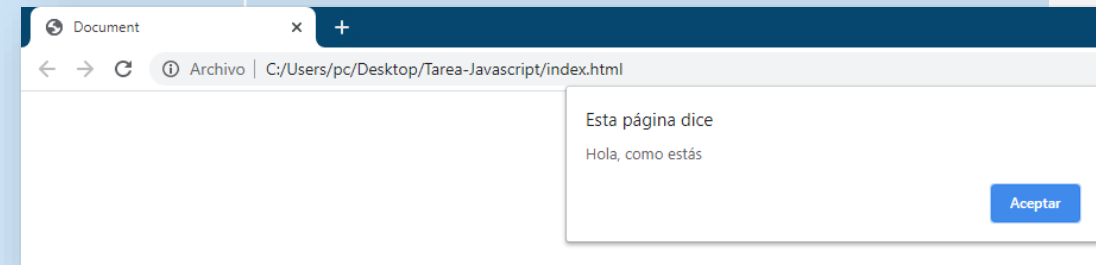
Archivo codigo.js

```
alert ( "Hola, como estás" )
```

Archivo index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script src="codigo.js"></script>
</body>
</html>
```

Vista Navegador

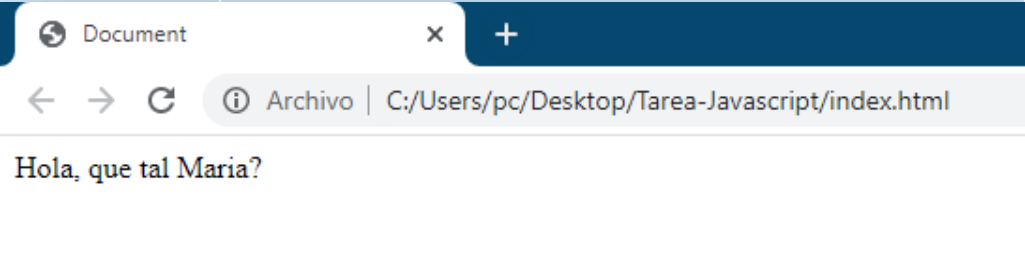


Función write

Esta función es un método del objeto **document** y lo que hace es escribir en la página el texto que se ingresa como parámetro.

Sintaxis:

`document.write(mensaje)`

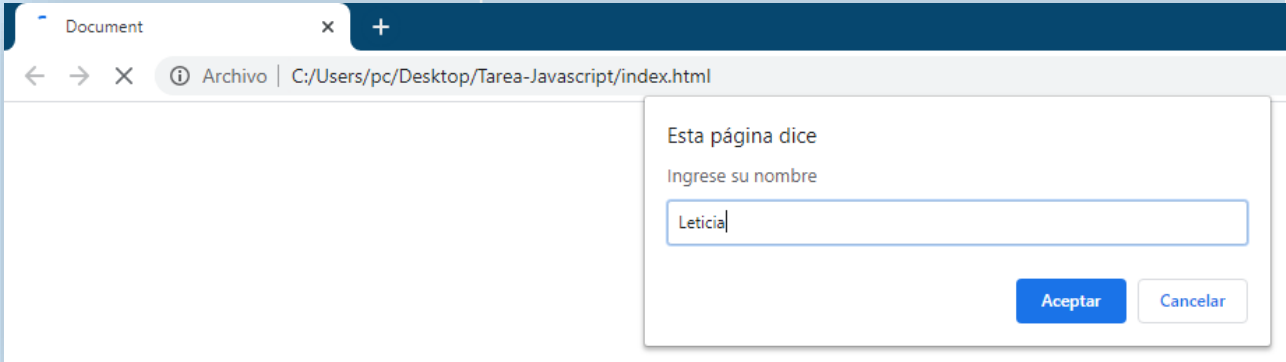
Archivo codigo.js	Archivo index.html	Vista Navegador
<pre>document.write("Hola, que tal Maria? ")</pre>	<pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content= "width=device-width, initial- scale=1.0"> <title>Document</title> </head> <body> <script src="codigo.js"></script> </body> </html></pre>	 A screenshot of a web browser window. The title bar shows a single tab labeled 'Document'. The address bar contains the text 'Archivo C:/Users/pc/Desktop/Tarea-Javascript/index.html'. The main content area of the browser displays the text 'Hola, que tal Maria?' in a simple black font on a white background.

Función prompt

Al igual que la función **alert**, la función **prompt** es también un método del objeto Window. Esta función se utiliza cuando el usuario ingresa datos por medio del teclado. Con esta función aparece una ventana en la pantalla, con un espacio para el valor que se debe ingresar y un botón **aceptar** para que la información sea guardada. Esta función recibe dos parámetros, el primero es el mensaje que se muestra en la ventana y el segundo es el valor inicial del área de texto.

Sintaxis:

`variable = prompt(mensaje, valor inicial)`

Archivo codigo.js	Archivo index.html	Vista Navegador
<pre>var nombre = prompt ("Ingrese su nombre", "") document.write("Hola "+ nombre)</pre>	<pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Document</title> </head> <body> <script src="codigo.js"></script> </body> </html></pre>	 A screenshot of a web browser window. The address bar shows the file path 'C:/Users/pc/Desktop/Tarea-Javascript/index.html'. A modal dialog box is displayed in the foreground with the title 'Esta página dice' and the text 'Ingrese su nombre'. Below the text is a text input field containing the name 'Leticia'. At the bottom right of the dialog are two buttons: 'Aceptar' (Accept) and 'Cancelar' (Cancel).

ACCEDER A ELEMENTOS JAVASCRIPT

El método `getElementById` permite, como su nombre indica, seleccionar un elemento del documento por medio del valor del atributo `id` que se le haya asignado. Su sintaxis es la siguiente:

```
document.getElementById('id_del_elemento');
```

Por medio de esta línea podríamos elegir cualquier elemento cuyo identificador conozcamos para manipularlo posteriormente. Por ejemplo, por medio de `document.getElementById('contenido')` seleccionaríamos el elemento con `id="contenido"`.

Fuentes Consultadas

<https://devcode.la/blog/que-es-javascript/>

<https://developer.mozilla.org/es/docs/Web/JavaScript>

<https://uniwebsidad.com/libros/javascript>

<https://aprendeenlinea.udea.edu.co/>