JavaScript Can Change HTML Content

Searching for (and fixing) errors in programming code is called

code debugging.

A JavaScript object is a collection of **named values**

Object values are written as **name : value** pairs (name and value separated by a colon).

JavaScript objects are containers for named values, called properties and methods.

Methods are functions stored as object properties.

Any JavaScript object can be converted to an array using Object.values():

In HTML, JavaScript code is inserted between <script> and </script> tags.

A JavaScript function is a block of JavaScript code, that can be executed when "called" for.

Scripts can be placed in the <body>, or in the <head> section of an HTML page, or in both.

Scripts can also be placed in external files by means of:

<script src="myScript.js"></script>

<script src="/js/myScript.js"></script>

<script src="https://www.w3schools.com/js/myScript.js"></script>

Placing scripts in external files has some advantages:

* It separates HTML and code
* It makes HTML and JavaScript easier to read and maintain
* Cached JavaScript files can speed up page loads

El metodo join(), toma los elementos de un array y los junta devolviendo una cadena. Donde su parametro es el separador dentro de la cadena.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

El metodo split(), toma una cadena y devuelve un array. Donde sus parametros, (separador,limite); Separador, indica el carácter a utilizar como referencia para dividir la cadena y limite es el entero que indica el numero de cadenas que se obtendran de la cadena original, o el numero de elementos en el array resultante.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

## JavaScript Programs

A **computer program** is a list of "instructions" to be "executed" by a computer.

In a programming language, these programming instructions are called **statements**.

A **JavaScript program** is a list of programming **statements**.

In HTML, JavaScript programs are executed by the web browser.

JavaScript programs (and JavaScript statements) are often called JavaScript code.

## Semicolons ;

Semicolons separate JavaScript statements.

## JavaScript White Space

JavaScript ignores multiple spaces. You can add white space to your script to make it more readable.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

## JavaScript Variables

In a programming language, **variables** are used to **store** data values.

JavaScript uses the keywords var, let and const to **declare** variables.

## Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto Descripción generada automáticamente Texto Descripción generada automáticamente JavaScript is Case Sensitive

All JavaScript identifiers are **case sensitive**.

## JavaScript and Camel Case

Historically, programmers have used different ways of joining multiple words into one variable name:

**Hyphens:**

first-name, last-name, master-card, inter-city.

Hyphens are not allowed in JavaScript. They are reserved for subtractions.

JavaScript programmers tend to use camel case that starts with a lowercase letter:

firstName, lastName, masterCard, interCity.

## JavaScript Character Set

JavaScript uses the **Unicode** character set.

Unicode covers (almost) all the characters, punctuations, and symbols in the world.

For a closer look, please study our [Complete Unicode Reference](https://www.w3schools.com/charsets/ref_html_utf8.asp).

**UTF-8** is a [variable-width](https://en.wikipedia.org/wiki/Variable-width_encoding) [character encoding](https://en.wikipedia.org/wiki/Character_encoding) used for electronic communication. Defined by the [Unicode Standard](https://en.wikipedia.org/wiki/Unicode_Standard), the name is derived from *Unicode* (or *Universal Coded Character Set*) *Transformation Format – 8-bit*.[[1]](https://en.wikipedia.org/wiki/UTF-8#cite_note-1)

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

## What are Variables?

Variables are containers for storing data (storing data values).

In this example, x, y, and z, are variables, declared with the var keyword:

JavaScript const variables must be assigned a value when they are declared:

Variables defined with const cannot be Redeclared.

Variables defined with const cannot be Reassigned.

Variables defined with const have Block Scope.

When to use JavaScript const?

As a general rule, always declare a variable with const unless you know that the value will change.

Use const when you declare:

* A new Array
* A new Object
* A new Function
* A new RegExp

When adding a number and a string, JavaScript will treat the number as a string.

JavaScript evaluates expressions from left to right. Different sequences can produce different results:

## avaScript Types are Dynamic

JavaScript has dynamic types. This means that the same variable can be used to hold different data types:

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

You can use quotes inside a string, as long as they don't match the quotes surrounding the string:}

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

JavaScript has only one type of numbers.

Numbers can be written with, or without decimals:

## JavaScript Objects

JavaScript objects are written with curly braces {}.

Object properties are written as name:value pairs, separated by commas.

## Undefined

In JavaScript, a variable without a value, has the value undefined. The type is also undefined.

## JavaScript Function Syntax

A JavaScript function is defined with the function keyword, followed by a **name**, followed by parentheses **()**.

Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).

The parentheses may include parameter names separated by commas:  
**(parameter1, parameter2, ...)**

The code to be executed, by the function, is placed inside curly brackets: **{}**

function name(parameter1, parameter2, parameter3) {  
  // code to be executed  
}

Function **parameters** are listed inside the parentheses () in the function definition.

Function **arguments** are the **values** received by the function when it is invoked.

Inside the function, the arguments (the parameters) behave as local variables.

A Function is much the same as a Procedure or a Subroutine, in other programming languages.

## Function Return

When JavaScript reaches a return statement, the function will stop executing.

If the function was invoked from a statement, JavaScript will "return" to execute the code after the invoking statement.

Functions often compute a **return value**. The return value is "returned" back to the "caller"

## Object Methods

Objects can also have **methods**.

Methods are **actions** that can be performed on objects.

Methods are stored in properties as **function definitions**.

|  |  |
| --- | --- |
| **Property** | **Property Value** |
|  |  |
|  |  |
| firstName | John |
| lastName | Doe |
| age | 50 |
| eyeColor | blue |
| fullName | function() {return this.firstName + " " + this.lastName;} |

# **JavaScript Even**

HTML events are **"things"** that happen to HTML elements.

When JavaScript is used in HTML pages, JavaScript can **"react"** on these events.

## JavaScript strings are for storing and manipulating text.

## Escape Character

Because strings must be written within quotes, JavaScript will misunderstand this string:

let text = "We are the so-called "Vikings" from the north.";

The string will be chopped to "We are the so-called ".

The solution to avoid this problem, is to use the **backslash escape character**.

The backslash (\) escape character turns special characters into string characters:

|  |  |  |
| --- | --- | --- |
| **Code** | **Result** | **Description** |
| \' | ' | Single quote |
| \" | " | Double quote |
| \\ | \ | Backslash |

The sequence \"  inserts a double quote in a string:

## String Length

To find the length of a string, use the built-in length property:

let length = text.length;

|  |  |
| --- | --- |
| \n | New Line |

## Note

All string methods return a new string. They don't modify the original string.

Formally said:

Strings are immutable: Strings cannot be changed, only replaced.

## Quotes Inside Strings

With **template literals**, you can use both single and double quotes inside a string:

### **Example** ` alt+96 = `

let text = `He's often called "Johnny"`;

## JavaScript Numbers are Always 64-bit Floating Point

Unlike many other programming languages, JavaScript does not define different types of numbers, like integers, short, long, floating-point etc.

JavaScript numbers are always stored as double precision floating point numbers, following the international IEEE 754 standard.  
  
This format stores numbers in 64 bits, where the number (the fraction) is stored in bits 0 to 51, the exponent in bits 52 to 62, and the sign in bit 63:

|  |  |  |
| --- | --- | --- |
| **Value (aka Fraction/Mantissa)** | **Exponent** | **Sign** |
| 52 bits (0 - 51) | 11 bits (52 - 62) | 1 bit (63) |

\*\*SPA Sigle Page Application.

You can use the global JavaScript function isNaN() to find out if a value is a not a number:

### **Example**

let x = 100 / "Apple";  
isNaN(x);

This returns true

Division by 0 (zero) also generates Infinity:

## Hexadecimal

JavaScript interprets numeric constants as hexadecimal if they are preceded by 0x.

### **Example**

let x = 0xFF;

An array is a special variable, which can hold more than one value:

## The Difference Between Arrays and Objects

In JavaScript, **arrays** use **numbered indexes**.

In JavaScript, **objects** use **named indexes**.

Arrays are a special kind of objects, with numbered indexes.

When to Use Arrays. When to use Objects.

* JavaScript does not support associative arrays.
* You should use **objects** when you want the element names to be **strings (text)**.
* You should use **arrays** when you want the element names to be **numbers**.