JavaScript Tutorial

JavaScript is the Web programming language.

All modern HTML pages are using JavaScript.

JavaScript is one of **3** languages all web developers **MUST** learn:

* **HTML** to define the content of web pages
* **CSS** to specify the layout of web pages
* **JavaScript** to [program](http://www.w3schools.com/js/default.asp) the behavior of web pages

JavaScript Introduction

JavaScript is the most popular programming language in the world for HTML, for the Web, for computers, servers, [laptops](http://www.w3schools.com/js/js_intro.asp), tablets, smart [phones](http://www.w3schools.com/js/js_intro.asp), and more.

JavaScript Can Change HTML Elements

The HTML DOM (the **D**ocument **O**bject **M**odel) is the official W3C standard for accessing HTML elements.

JavaScript can manipulate the DOM (change HTML contents).

One of many HTML methods is document.getElementById().

The method **document.getElementById()** is one of many methods in the HTML DOM.

You can use JavaScript to:

* Change HTML elements
* Delete HTML elements
* Create new HTML elements
* Copy and clone HTML elements
* And much more ...

JavaScript Can Change HTML Attributes

This example changes the value of the source attribute (src) of an HTML <image> element:

With JavaScript, you can change almost any HTML attribute.

JavaScript Can Change HTML Styles (CSS)

With JavaScript, you can change almost any CSS values.

JavaScript Can Validate Data

JavaScript is often used to validate input:

JavaScript Where To

JavaScripts can be put in the <body> and in the <head> section of an HTML page.

In HTML, JavaScripts must be inserted between <script> and </script> tags.

The <script> Tag

To insert a **JavaScript** into an HTML page, use the <script> tag.

The <script> and </script> tells where the JavaScript starts and ends.

The lines between <script> and </script> contain the JavaScript code.

Older examples may use a type attribute: <script type="text/javascript">.

The type attribute is not required.

JavaScript is the default scripting language in all modern browsers and in HTML5.

JavaScript Functions and Events

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

For example, a function can be executed when an event occurs, like when the user clicks a button.

JavaScript code inside a **function**, can be invoked later, when an event occurs.

**Invoke a function = Call upon a function** (ask for the code in the function to be executed).

JavaScript in <head> and <body>

You can place any number of scripts in an HTML document.

It is a good idea to place scripts at the bottom of the <body> element.

This improves page load, because HTML display is not blocked by scripts loading.

External JavaScripts

Scripts can also be placed in external files.

External scripts are practical when the same code is used in many different web pages.

JavaScript files have the **file extension .js**.

To use an external script, put the name of the script file in the source (src) attribute of the <script> tag :

<!DOCTYPE html>

<html>

<body>

<h1>My Web Page</h1>

<p id="demo">A Paragraph.</p>

<button type="button" onclick="myFunction()">Try it</button>

<p><strong>Note:</strong> myFunction is stored in an external file called "myScript.js".</p>

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

</body>

</html>

You can place an external script reference in <head> or <body> as you like.

The script will behave as if it was located exactly where you put the reference in the HTML document.

External scripts cannot contain <script> tags.

External JavaScript Advantages

Placing JavaScripts 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.

JavaScript Output

**JavaScript** does not have any print or output functions.

In HTML, JavaScript can only be used to manipulate HTML elements.

JavaScript Display Possibilities

JavaScript can "display" data in different ways:

Writing into an alert box, using window.alert().

Writing into the HTML output using document.write().

Writing into an HTML element, using innerHTML.

Writing into the browser console, using console.log().

Using innerHTML

To access an HTML element, JavaScript can use the document.getElementById(id) method.

The id attribute defines the HTML element. The innerHTML property defines the HTML content:

Manipulating HTML Elements

To access an HTML element from JavaScript, you can use the document.getElementById(*id*) method.

Use the "id" attribute to identify the HTML element, and innerHTML to refer to the element content:

<!DOCTYPE html>

<html>

<body>

<h1>My First Web Page</h1>

<p id="demo">My First Paragraph.</p>

<script>

document.getElementById("demo").innerHTML = "Paragraph changed.";

</script>

</body>

</html>

The JavaScript statement above (inside the <script> tag) is [executed](http://www.w3schools.com/js/js_output.asp) by the web browser:

**document.getElementById("demo")** is JavaScript code for finding an HTML element using the id attribute.

**innerHTML = "Paragraph changed."** is JavaScript code for changing an element's HTML content (innerHTML).

Writing **output** into a <p> element with id="demo".

JavaScript Syntax

JavaScript syntax is the rules, how JavaScript programs are constructed.

The principles by which sentences are constructed in a language.

JavaScript Programs

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

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

JavaScript is a programming language.

JavaScript statements are separated by semicolon.

In HTML, JavaScript programs can be executed by the web browser.

JavaScript Statements

JavaScript statements are composed of:

Values, Operators, Expressions, Keywords, and Comments.

JavaScript Values

The JavaScript syntax defines two types of values: Fixed values and variable values.

Fixed values are called literals. Variable values are called variables.

JavaScript Variables

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

JavaScript uses the var keyword to define variables.

An equal sign is used to assign values to variables.

JavaScript Variables

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

JavaScript uses the var keyword to define variables.

An equal sign is used to assign values to variables.

**Array literals** defines an array:

[40, 100, 1, 5, 25, 10]

**Object literals** defines an object:

{firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"}

**Function literals** defines a function:

function myFunction(a, b) { return a \* b;}

JavaScript Character Set

JavaScript uses the Unicode character set.

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

JavaScript Statements

In HTML, JavaScript statements are "instructions" to be "executed" by the web browser.

JavaScript Programs

Most JavaScript programs contain many JavaScript statements.

The statements are executed, one by one, in the same order as they are written.

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

Semicolons separate JavaScript statements.

JavaScript White Space

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

JavaScript Line Length and Line Breaks

For best readability, programmers often like to avoid code lines longer than 80 characters.

If a JavaScript statement does not fit on one line, the best place to break it, is after an operator:

JavaScript Keywords

JavaScript statements often start with a keyword to identify the JavaScript action to be performed.

Keyword Description

break Terminates a switch or a loop

continue Jumps out of a loop and starts at the top

debugger Stops the execution of JavaScript, and calls (if available) the debugging function

do ... while Executes a block of statements, and repeats the block, while a condition is true

for Marks a block of statements to be executed, as long as a condition is true

function Declares a function

if ... else Marks a block of statements to be executed, depending on a condition

return Exits a function

switch Marks a block of statements to be executed, depending on different cases

try ... catch Implements error handling to a block of statements

var Declares a variable

<http://www.w3schools.com/js/js_statements.asp>