Chapter1

JavaScript is the programming language of HTML and the Web.

Programming makes computers do what you want them to do.

Why Study JavaScript?

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

   1. **HTML** to define the content of web pages

   2. **CSS** to specify the layout of web pages

   3. **JavaScript** to program the behavior of web pages

<!DOCTYPE html>

<html>

<body>

<h1>My First JavaScript</h1>

<button type="button"

onclick="document.getElementById('demo').innerHTML = Date()">

Click me to display Date and Time.</button>

<p id="demo"></p>

</body>

</html>

## JavaScript Can Change HTML Content

One of many JavaScript HTML methods is **getElementById()**.

This example uses the method to "find" an HTML element (with id="demo") and changes the element content (**innerHTML**) to "Hello JavaScript":

### Example

### <!DOCTYPE html>

### <html>

### <body>

### <h1>What Can JavaScript Do?</h1>

### <p id="demo">JavaScript can change HTML content.</p>

### <button type="button" onclick="document.getElementById('demo').innerHTML = 'Hello JavaScript!'">Click Me!</button>

### </body>

### </html>

[Try it Yourself »](http://www.w3schools.com/js/tryit.asp?filename=tryjs_intro_inner_html)

## JavaScript Can Change HTML Attributes

This example changes an HTML image by changing the src (source) attribute of an <img> tag:

<!DOCTYPE html>

<html>

<body>

<h1>What Can JavaScript Do?</h1>

<p>JavaScript can change HTML attributes.</p>

<p>In this case JavaScript changes the src (source) attribute of an image.</p>

<button onclick="document.getElementById('myImage').src='pic\_bulbon.gif'">Turn on the light</button>

<img id="myImage" src="pic\_bulboff.gif" style="width:100px">

<button onclick="document.getElementById('myImage').src='pic\_bulboff.gif'">Turn off the light</button>

</body>

</html>

## JavaScript Can Change HTML Styles (CSS)

Changing the style of an HTML element, is a variant of changing an HTML attribute:

<!DOCTYPE html>

<html>

<body>

<h1>What Can JavaScript Do?</h1>

<p id="demo">JavaScript can change the style of an HTML element.</p>

<button type="button" onclick="document.getElementById('demo').style.fontSize='35px'">Click Me!</button>

</body>

</html>

## JavaScript Can Hide HTML Elements

Hiding HTML elements can be done by changing the display style:

<!DOCTYPE html>

<html>

<body>

<h1>What Can JavaScript Do?</h1>

<p id="demo">JavaScript can hide HTML elements.</p>

<button type="button" onclick="document.getElementById('demo').style.display='none'">Click Me!</button>

</body>

</html>

## JavaScript Can Show HTML Elements

Showing hidden HTML elements can also be done by changing the display style:

<!DOCTYPE html>

<html>

<body>

<h1>What Can JavaScript Do?</h1>

<p>JavaScript can show hidden HTML elements.</p>

<p id="demo" style="display:none">Hello JavaScript!</p>

<button type="button" onclick="document.getElementById('demo').style.display='block'">Click Me!</button>

</body>

</html>

# JavaScript Where To

[« Previous](http://www.w3schools.com/js/js_intro.asp)

[Next Chapter »](http://www.w3schools.com/js/js_output.asp)

JavaScript can be placed in the <body> and the <head> sections of an HTML page.

## The <script> Tag

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

<script>  
document.getElementById("demo").innerHTML = "My First JavaScript";  
</script>

Older examples may use a type attribute: <script type="text/javascript">.  
This type attribute is not required. JavaScript is the default scripting language in HTML.

## 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.

You will learn much more about functions and events in later chapters.

## JavaScript in <head> or <body>

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

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

Keeping all code in one place, is always a good habit.

## JavaScript in <head>

In this example, a JavaScript function is placed in the <head> section of an HTML page.

The function is invoked (called) when a button is clicked:

### Example

<!DOCTYPE html>  
<html>

<head>  
<script>  
function myFunction() {  
    document.getElementById("demo").innerHTML = "Paragraph changed.";  
}  
</script>  
</head>

<body>

<h1>My Web Page</h1>

<p id="demo">A Paragraph</p>

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

</body>  
</html>

[Try it Yourself »](http://www.w3schools.com/js/tryit.asp?filename=tryjs_whereto_head)

## JavaScript in <body>

In this example, a JavaScript function is placed in the <body> section of an HTML page.

The function is invoked (called) when a button is clicked:

### Example

<!DOCTYPE html>  
<html>  
<body>   
  
<h1>My Web Page</h1>  
  
<p id="demo">A Paragraph</p>  
  
<button type="button" onclick="myFunction()">Try it</button>  
  
<script>  
function myFunction() {  
   document.getElementById("demo").innerHTML = "Paragraph changed.";  
}  
</script>  
  
</body>  
</html>

[Try it Yourself »](http://www.w3schools.com/js/tryit.asp?filename=tryjs_whereto_body)

It is a good idea to place scripts at the bottom of the <body> element.  
This can improve page load, because script compilation can slow down the display.

## External JavaScript

Scripts can also be placed in external files:

### myScript.js

function myFunction() {  
   document.getElementById("demo").innerHTML = "Paragraph changed.";  
}

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 src (source) attribute of a <script> tag:

### Example

<!DOCTYPE html>  
<html>  
<body>  
<script src="myScript.js"></script>  
</body>  
</html>

[Try it Yourself »](http://www.w3schools.com/js/tryit.asp?filename=tryjs_whereto_external)

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 the <script> tag is located.

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

[« Previous](http://www.w3schools.com/js/js_intro.asp)

[Next Chapter »](http://www.w3schools.com/js/js_output.asp)

# JavaScript Output

[« Previous](http://www.w3schools.com/js/js_whereto.asp)

[Next Chapter »](http://www.w3schools.com/js/js_syntax.asp)

JavaScript does NOT have any built-in print or display functions.

## 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 window.alert()

You can use an alert box to display data:

### Example

<!DOCTYPE html>  
<html>  
<body>  
  
<h1>My First Web Page</h1>  
<p>My first paragraph.</p>  
  
<script>  
window.alert(5 + 6);  
</script>  
  
</body>  
</html>

[Try it Yourself »](http://www.w3schools.com/js/tryit.asp?filename=tryjs_output_alert)

## Using document.write()

For testing purposes, it is convenient to use **document.write()**:

### Example

<!DOCTYPE html>  
<html>  
<body>  
  
<h1>My First Web Page</h1>  
<p>My first paragraph.</p>  
  
<script>  
document.write(5 + 6);  
</script>  
  
</body>  
</html>

[Try it Yourself »](http://www.w3schools.com/js/tryit.asp?filename=tryjs_output_write)

Using document.write() after an HTML document is fully loaded, will **delete all existing HTML**:

### Example

<!DOCTYPE html>  
<html>  
<body>  
  
<h1>My First Web Page</h1>  
<p>My first paragraph.</p>  
  
<button onclick="document.write(5 + 6)">Try it</button>  
  
</body>  
</html>

[Try it Yourself »](http://www.w3schools.com/js/tryit.asp?filename=tryjs_output_write_over)

The document.write() method should only be used for testing.

## 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:

### Example

<!DOCTYPE html>  
<html>  
<body>  
  
<h1>My First Web Page</h1>  
<p>My First Paragraph</p>  
  
<p id="demo"></p>  
  
<script>  
document.getElementById("demo").innerHTML = 5 + 6;  
</script>  
  
</body>  
</html>

[Try it Yourself »](http://www.w3schools.com/js/tryit.asp?filename=tryjs_output_dom)

To "display data" in HTML, (in most cases) you will set the value of an innerHTML property.