



What is JavaScript?

- JavaScript is the programming language of the web.
- It can update and change both HTML and CSS.
- It can calculate, manipulate and validate data.



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

JavaScript Can Change HTML Content

One of many JavaScript HTML methods is `getElementById()`. The example below "finds" an HTML element (with `id="demo"`), and changes the element content (`innerHTML`) to "Hello JavaScript":

```
<!DOCTYPE html>
<html>
<body>

<h2>What Can JavaScript Do?</h2>

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

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

</body>
</html>
```

JavaScript Can Change HTML Attribute Values

In this example JavaScript changes the value of the `src` (source) attribute of an `<img>` tag:



```
<!DOCTYPE html>
<html>
<body>

<h2>What Can JavaScript Do?</h2>

<p>JavaScript can change HTML attribute values.</p>

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

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



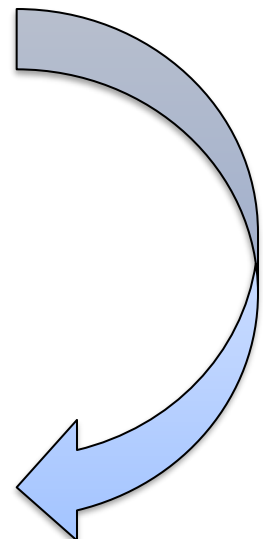
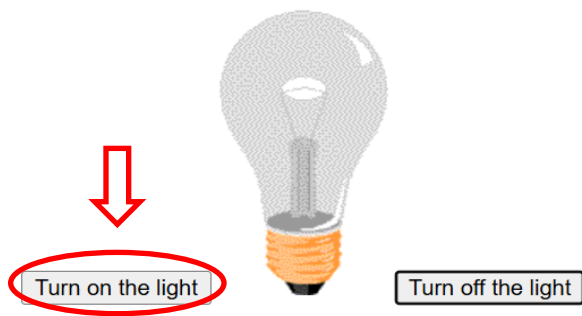
<button onclick="document.getElementById('myImage').src='pic_bulboff.gif'">Turn off the
light</button>

</body>
</html>
```

What Can JavaScript Do?

JavaScript can change HTML attribute values.

In this case JavaScript changes the value of the src (source) attribute of an image.





What Can JavaScript Do?

JavaScript can change HTML attribute values.

In this case JavaScript changes the value of the src (source) attribute of an image.



Turn on the light

Turn off the light

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>

<h2>What Can JavaScript Do?</h2>

<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:

```
document.getElementById("demo").style.display = "none";
```



JavaScript Can Show HTML Elements

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

```
document.getElementById("demo").style.display = "block";
```

The <script> Tag

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

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript in Body</h2>

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

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

</body>
</html>
```

Functions and Events

JavaScript function is a block of JavaScript code, that can be executed when "called" for. For example, a function can be called when an **event** occurs, like when the user clicks a button.

<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. 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:

```
<!DOCTYPE html>
<html>
<head>
```



```
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
</head>
<body>
<h2>Demo JavaScript in Head</h2>

<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>
</body>
</html>
```

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:

```
<!DOCTYPE html>
<html>
<body>

<h2>Demo JavaScript in Body</h2>

<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>
```

External JavaScript

Scripts can also be placed in external files:



External file: **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:

```
<!DOCTYPE html>  
<html>  
<body>  
  
<h2>Demo External JavaScript</h2>  
  
<p id="demo">A Paragraph.</p>  
  
<button type="button" onclick="myFunction()">Try it</button>  
  
<p>This example links to "myScript.js".</p>  
<p>(myFunction is stored in "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 the `<script>` tag is located.

External JavaScript Advantages

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



To add several script files to one page - use several script tags:

```
<script src="myScript1.js"></script>  
<script src="myScript2.js"></script>
```

External References

An external script can be referenced in 3 different ways:

- With a full URL (a full web address)
- With a file path (like /js/)
- Without any path

This example uses a **full URL** to link to myScript.js:

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

This example uses a **file path** to link to myScript.js:

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

This example uses no path to link to myScript.js:

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