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>

<!DOCTYPE html>

<html>

<head>

    <title>JavaScript Function on Click</title>

</head>

<body>

    <h2>What Can JavaScript Do?</h2>

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

    <button onclick="change()">change</button>

    <script>

        function change(){

           var demo = document.getElementById("demo")

           demo.textContent="content changed by js"

        }

    </script>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

    <title>Click Count Example</title>

    <script>

        // Initialize a counter variable

        let clickCount = 0;

        // Define the JavaScript function

        function handleClick() {

            // Increment the click count

            clickCount++;

            // Update the display

            document.getElementById('clickCountDisplay').innerText = 'Clicked: ' + clickCount;

        }

    </script>

</head>

<body>

    <!-- Example button that increments the click count -->

    <button onclick="handleClick()">Click me</button>

    <!-- Display the click count -->

    <div id="clickCountDisplay">Click count: 0</div>

</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>

<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>

<!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="clickon()">Turn on the light</button>

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

<button onclick="clickoff()">Turn off the light</button>

    <script>

        function clickon(){

            var myImage = document.getElementById('myImage')

            myImage.src='pic\_bulbon.gif'

        }

        function clickoff(){

            var myImage = document.getElementById('myImage')

            myImage.src='pic\_bulboff.gif'

        }

    </script>

</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>

<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>

<!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:

<!DOCTYPE html>

<html>

<body>

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

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

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

</body>

</html>

<!DOCTYPE html>

<html>

<body>

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

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

<button onclick="clicked()">Click Me!</button>

<script>

    function clicked(){

        var demo=document.getElementById('demo')

         demo.style.display='none'

    }

</script>

</body>

</html>

## JavaScript Can Show HTML Elements

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

--------------------------------------------------------------------------------------------------------------------------------

<!DOCTYPE html>

<html>

<body>

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

<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>

<!DOCTYPE html>

<html>

<body>

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

<p id="demo" style="display: none;">JavaScript can hide HTML elements.</p>

<button onclick="clicked()">Click Me!</button>

<script>

    function clicked(){

        var demo=document.getElementById('demo')

         demo.style.display='block'

    }

</script>

</body>

</html>

<!DOCTYPE html>

<html>

<body>

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

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

<button onclick="clicked()">Click Me!</button>

<script>

    function clicked(){

        var demo=document.getElementById('demo')

         demo.style.backgroundColor='red'

    }

</script>

</body>

</html>

-------------------------------------------------------------------------------------------------------------------------------

# JavaScript Where To

## The <script> Tag

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript in Body</h2>

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

<script>

document.getElementById("demo").innerHTML = "My First JavaScript";

</script>

</body>

</html>

--------------------------------------------------------------------------------------------------------------------------------------

## 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>

<h2>Demo JavaScript in Head</h2>  
  
<p id="demo">A Paragraph</p>  
<button type="button" onclick="myFunction()">Try it</button>

</body>  
</html>

## 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>  
  
<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.";  
}

<!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>

# JavaScript Output

JavaScript Display Possibilities

JavaScript can "display" data in different ways:

* Writing into an HTML element, using innerHTML.
* Writing into the HTML output using document.write().
* Writing into an alert box, using window.alert().
* Writing into the browser console, using console.log().

<!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>

## 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>

## 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>

## Using console.log()

For debugging purposes, you can call the console.log() method in the browser to display data.

You will learn more about debugging in a later chapter.

### Example

<!DOCTYPE html>  
<html>  
<body>  
  
<script>  
console.log(5 + 6);  
</script>  
  
</body>  
</html>

## JavaScript Print

JavaScript does not have any print object or print methods.

You cannot access output devices from JavaScript.

The only exception is that you can call the window.print() method in the browser to print the content of the current window.

### Example

<!DOCTYPE html>  
<html>  
<body>  
  
<button onclick="window.print()">Print this page</button>  
  
</body>  
</html>