JavaScript Strings

Back to JS page

Table of Contents

- JavaScript Strings
  - Using Quotes
    - Example 1
  - Quotes Inside Quotes
    - Example 2
  - String Length
    - Example 3
  - Escape Characters
    - Example 4
  - Breaking Long Lines
    - Example 5
  - Template Strings
    - Example 6
  - JavaScript Strings as Objects
    - Example 7
  - Document
  - Reference

Using Quotes

A JavaScript string is zero or more characters written inside quotes.

You can use single or double quotes:

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

<h2>JavaScript Strings</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Using Quotes</h4>
<p id="demo"></p>

<script>
let carName1 = "Volvo XC60";
let carName2 = 'Volvo XC60';
document.getElementById("demo").innerHTML =
  carName1 + "<br>" + carName2;
</script>

</body>
</html>
```

Example 1

Result View Example

Quotes Inside Quotes

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

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

<h2>JavaScript Strings</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Quotes Inside Quotes</h4>
<p id="demo"></p>

<script>
let answer1 = "It's alright";
let answer2 = "He is called 'Johnny'";
let answer3 = 'He is called "Johnny"';
document.getElementById("demo").innerHTML =
  answer1 + "<br>" + answer2 + "<br>" + answer3;
</script>

</body>
</html>
```

Example 2

Result View Example

String Length

The `length` property returns the length of a string:

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

<h2>JavaScript Strings</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>String Length</h4>
<p id="demo"></p>

<script>
let text = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
document.getElementById("demo").innerHTML = "Length: " + text.length;
</script>

</body>
</html>
```

Example 3

Result View Example

Escape Characters

The backslash escape character `\` turns special characters into string characters:

Code	Result
`\'`	'
`\"`	"
`\\`	\
`\n`	new line

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

<h2>JavaScript Strings</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Escape Characters</h4>
<p id="demo"></p>

<script>
let text = "We are the so-called \"Vikings\" from the north.";
document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

Example 4

Result View Example

Breaking Long Lines

For best readability, programmers often like to avoid code lines longer than 80 characters. You can break up a code line within a text string with a single backslash:

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

<h2>JavaScript Strings</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Breaking Long Lines</h4>
<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello \
Dolly!";
</script>

</body>
</html>
```

Example 5

Result View Example

Template Strings

Template strings use backticks (`) instead of quotes to define a string.

With template strings you can use both single and double quotes inside a string:

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

<h2>JavaScript Strings</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Template Strings</h4>
<p id="demo"></p>

<script>
let text = `He's often called "Johnny"`;
document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

Example 6

Result View Example

JavaScript Strings as Objects

Normally, JavaScript strings are primitive values, created from literals.

But strings can also be defined as objects with the keyword `new`:

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

<h2>JavaScript Strings</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Strings as Objects</h4>
<p id="demo1"></p>
<p id="demo2"></p>

<script>
let x = "John";
let y = new String("John");
document.getElementById("demo1").innerHTML = typeof x + "<br>" + typeof y;
document.getElementById("demo2").innerHTML = "x === y is " + (x === y);
</script>

</body>
</html>
```

Example 7

Result View Example

Document

Document in project

You can Download PDF file.

Reference

- W3Schools JavaScript Strings