JavaScript Primitive Types

Back to JS page

Table of Contents

- JavaScript Primitive Types
  - JavaScript Strings
    - Example 1
  - JavaScript Numbers
    - Example 2
  - JavaScript BigInt
    - Example 3
  - JavaScript Booleans
    - Example 4
  - Undefined, Empty Values, and null
    - Example 5
  - Document
  - Reference

JavaScript Strings

A string is a sequence of characters. Strings are written inside quotes. You can use single or double quotes:

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

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

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

<script>
// Strings can be written in single or double quotes
const name1 = "John Doe";
const name2 = 'John Doe';

// Numbers in quotes are still strings
const num1 = "25";
const num2 = "25.5";

let text = "name1: " + name1 + "<br>";
text += "name2: " + name2 + "<br>";
text += "typeof '25': " + typeof num1 + "<br>";
text += "typeof '25.5': " + typeof num2 + "<br><br>";

text += "Strings can be concatenated:<br>";
text += name1 + " is " + 25 + " years old";

document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

Example 1

Result View Example

JavaScript Numbers

Numbers can be written with or without decimals. They can also use exponential notation:

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

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

<h4>JavaScript Numbers</h4>
<p id="demo"></p>

<script>
// Numbers can be integers or floats
const x1 = 34.00;
const x2 = 34;
const x3 = 3.14;

// Exponential notation
const y1 = 123e5;      // 12300000
const y2 = 123e-5;     // 0.00123

let text = "x1 = 34.00: " + x1 + " (" + typeof x1 + ")<br>";
text += "x2 = 34: " + x2 + " (" + typeof x2 + ")<br>";
text += "x3 = 3.14: " + x3 + "<br><br>";
text += "123e5 = " + y1 + "<br>";
text += "123e-5 = " + y2 + "<br><br>";

text += "Integer (32-bit): " + (x2 | 0) + "<br>";
text += "Float (64-bit): " + x3;

document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

Example 2

Result View Example

JavaScript BigInt

`BigInt` is for numbers too large for normal Number. Created by appending `n` to the end:

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

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

<h4>JavaScript BigInt</h4>
<p id="demo"></p>

<script>
// BigInt is for very large integers
const big1 = 12345678901234567890123456789n;
const big2 = BigInt("12345678901234567890123456789");

// Normal number (limited precision)
const normal = 12345678901234567890123456789;

let text = "Normal number: " + normal + "<br>";
text += "(loses precision for large values)<br><br>";
text += "BigInt: " + big1 + "<br>";
text += "big2: " + big2 + "<br><br>";
text += "typeof BigInt: " + typeof big1 + "<br><br>";

text += "BigInt supports arithmetic:<br>";
text += "big1 + 1n: " + (big1 + 1n);

document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

Example 3

Result View Example

JavaScript Booleans

Booleans can only have two values: `true` or `false`:

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

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

<h4>JavaScript Booleans</h4>
<p id="demo"></p>

<script>
// Booleans are true or false
const isTrue = true;
const isFalse = false;

// Booleans from comparisons
const isGreater = 5 > 2;
const isEqual = 5 === 2;

let text = "true: " + isTrue + " (" + typeof isTrue + ")<br>";
text += "false: " + isFalse + " (" + typeof isFalse + ")<br><br>";
text += "5 > 2: " + isGreater + "<br>";
text += "5 === 2: " + isEqual;

document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

Example 4

Result View Example

Undefined, Empty Values, and null

`undefined` is a variable without a value. `null` is an empty value. `""` is an empty string:

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

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

<h4>Undefined, Empty Values, and null</h4>
<p id="demo"></p>

<script>
// Undefined - variable without a value
let car;   // undefined

// null - explicitly empty value
let person = null;

// Empty string
let text = "";

let output = "let car; → " + car + " (" + typeof car + ")<br>";
output += "let person = null; → " + person + " (" + typeof person + ")<br>";
output += "let text = ''; → '" + text + "' (" + typeof text + ")<br><br>";

// Setting to undefined
let x = "Hello";
x = undefined;
output += "x = 'Hello'; x = undefined → " + x + " (" + typeof x + ")";

document.getElementById("demo").innerHTML = output;
</script>

</body>
</html>
```

Example 5

Result View Example

Document

Document in project

You can Download PDF file.

Reference

- W3Schools JavaScript Primitive Types