JavaScript Typed Array Reference

Back to JS page

Table of Contents

- JavaScript Typed Array Reference
  - Typed Array Types Overview
    - Example 1
  - Common Methods Reference
    - Example 2
  - Working with Buffers
    - Example 3
  - Document
  - Reference

Typed Array Types Overview

JavaScript provides the following typed array types:

Constructor	Type	Range	Size	Description
`Int8Array`	8-bit signed integer	-128 to 127	1 byte	
`Uint8Array`	8-bit unsigned integer	0 to 255	1 byte	
`Uint8ClampedArray`	8-bit unsigned integer (clamped)	0 to 255	1 byte	Canvas data
`Int16Array`	16-bit signed integer	-32768 to 32767	2 bytes	
`Uint16Array`	16-bit unsigned integer	0 to 65535	2 bytes	
`Int32Array`	32-bit signed integer	-2^31 to 2^31-1	4 bytes	
`Uint32Array`	32-bit unsigned integer	0 to 2^32-1	4 bytes	
`BigInt64Array`	64-bit signed integer	-(2^63) to 2^63-1	8 bytes	
`BigUint64Array`	64-bit unsigned integer	0 to 2^64-1	8 bytes	
`Float32Array`	32-bit floating point	~1.2E-38 to 3.4E38	4 bytes	
`Float64Array`	64-bit floating point	~5.0E-324 to 1.8E308	8 bytes	

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

<h2>JavaScript Typed Array Reference</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Typed Array Types Overview</h4>
<p id="demo"></p>

<script>
// Demonstrate different typed array types
const int8 = new Int8Array([-10, 0, 50, 100]);
const uint8 = new Uint8Array([0, 100, 200, 255]);
const int16 = new Int16Array([-1000, 0, 1000, 30000]);
const int32 = new Int32Array([-100000, 0, 100000]);
const float32 = new Float32Array([1.5, 3.14, 9.99]);
const float64 = new Float64Array([0.123456789, 123456.789]);

let text = "";

text += "Int8Array: " + Array.from(int8).join(", ") + "<br>";
text += "Uint8Array: " + Array.from(uint8).join(", ") + "<br>";
text += "Int16Array: " + Array.from(int16).join(", ") + "<br>";
text += "Int32Array: " + Array.from(int32).join(", ") + "<br>";
text += "Float32Array: " + Array.from(float32).join(", ") + "<br>";
text += "Float64Array: " + Array.from(float64).join(", ");

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

</body>
</html>
```

Example 1

Result View Example

Common Methods Reference

Typed arrays support many standard array methods. Here's a quick reference of commonly used methods:

Method	Description
`from()`	Creates a typed array from an array-like object
`of()`	Creates a typed array from variable arguments
`fill()`	Fills elements with a static value
`find()`	Returns first element that passes a test
`some()`	Checks if any element passes a test
`every()`	Checks if all elements pass a test
`filter()`	Creates a new array with elements that pass a test
`map()`	Creates a new array by applying a function
`reduce()`	Reduces the array to a single value
`sort()`	Sorts the elements
`slice()`	Extracts a section of the array
`includes()`	Checks if an element is in the array
`indexOf()`	Returns the index of a specified element
`join()`	Joins all elements into a string
`reverse()`	Reverses the order of elements
`forEach()`	Executes a function for each element

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

<h2>JavaScript Typed Array Reference</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Common Methods Reference</h4>
<p id="demo"></p>

<script>
const numbers = new Int8Array([10, 5, 20, 15, 25, 30]);

// Using various array methods on typed arrays
let text = "Original: " + Array.from(numbers).join(", ") + "<br><br>";

// filter
const filtered = numbers.filter(function(v) { return v > 15; });
text += "filter(v > 15): " + Array.from(filtered).join(", ") + "<br>";

// map
const doubled = numbers.map(function(v) { return v * 2; });
text += "map(v * 2): " + Array.from(doubled).join(", ") + "<br>";

// includes
text += "includes(20): " + numbers.includes(20) + "<br>";

// indexOf
text += "indexOf(15): " + numbers.indexOf(15) + "<br>";

// every
const allPositive = numbers.every(function(v) { return v > 0; });
text += "every(v > 0): " + allPositive + "<br>";

// reduce
const sum = numbers.reduce(function(acc, v) { return acc + v; }, 0);
text += "reduce(sum): " + sum;

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

</body>
</html>
```

Example 2

Result View Example

Working with Buffers

Typed arrays are built on top of `ArrayBuffer`. Multiple typed arrays can share the same buffer:

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

<h2>JavaScript Typed Array Reference</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Working with Buffers</h4>
<p id="demo"></p>

<script>
// Create a buffer of 16 bytes
const buffer = new ArrayBuffer(16);

// Create different views of the same buffer
const int8View = new Int8Array(buffer);
const uint16View = new Uint16Array(buffer);
const int32View = new Int32Array(buffer);

// Write data using one view
int8View[0] = 10;
int8View[1] = 20;
int32View[2] = 100000;

let text = "Same buffer, different views:<br><br>";
text += "Int8Array view: ";
for (let i = 0; i < int8View.length; i++) text += int8View[i] + " ";
text += "<br><br>";

text += "Buffer byteLength: " + buffer.byteLength + " bytes<br><br>";

text += "Int8Array length: " + int8View.length + " (1 byte each)<br>";
text += "Uint16Array length: " + uint16View.length + " (2 bytes each)<br>";
text += "Int32Array length: " + int32View.length + " (4 bytes each)";

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

</body>
</html>
```

Example 3

Result View Example

Document

Document in project

You can Download PDF file.

Reference

- W3Schools JavaScript Typed Array Reference