JavaScript Symbol Datatype

Back to JS page

Table of Contents

- JavaScript Symbol Datatype
  - Creating Symbols
    - Example 1
  - Using Symbols as Object Keys
    - Example 2
  - Symbols are Not Included in for...in
    - Example 3
  - Global Symbols
    - Example 4
  - Document
  - Reference

Creating Symbols

A Symbol is a unique and immutable primitive value. Two symbols with the same description are still different:

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

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

<h4>Creating Symbols</h4>
<p id="demo"></p>

<script>
// Create symbols
const sym1 = Symbol();
const sym2 = Symbol();
const sym3 = Symbol("id");

// Symbols with same description are still different
const sym4 = Symbol("id");

let text = "sym1: " + sym1.toString() + "<br>";
text += "sym2: " + sym2.toString() + "<br>";
text += "sym3: " + sym3.toString() + "<br><br>";

text += "sym1 === sym2: " + (sym1 === sym2) + "<br>";
text += "sym3 === sym4: " + (sym3 === sym4) + " (same description, different symbols)<br><br>";
text += "typeof Symbol: " + typeof sym1;

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

</body>
</html>
```

Example 1

Result View Example

Using Symbols as Object Keys

Symbols can be used as unique object keys - they are hidden identifiers:

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

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

<h4>Using Symbols as Object Keys</h4>
<p id="demo"></p>

<script>
// Create symbol for secret key
const secretKey = Symbol("secret");

// Use symbol as object key
const person = {
  name: "John",
  age: 30,
  [secretKey]: "Hidden value"
};

let text = "person.name: " + person.name + "<br>";
text += "person.age: " + person.age + "<br><br>";

// Access using the symbol
text += "person[secretKey]: " + person[secretKey] + "<br><br>";

// Object.keys() does NOT show symbols
text += "Object.keys(person): " + Object.keys(person) + "<br>";
text += "Symbol is hidden from normal enumeration";

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

</body>
</html>
```

Example 2

Result View Example

Symbols are Not Included in for...in

Symbols are not included in `for...in` loops or `JSON.stringify()`:

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

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

<h4>Symbols are Not Included in for...in</h4>
<p id="demo"></p>

<script>
const secretKey = Symbol("secret");

const person = {
  name: "John",
  age: 30,
  [secretKey]: "Hidden"
};

// for...in does NOT show symbol keys
let text = "for...in loop:<br>";
for (let key in person) {
  text += key + ": " + person[key] + "<br>";
}

text += "<br>";

// JSON.stringify does NOT show symbols
text += "JSON.stringify(person):<br>";
text += JSON.stringify(person);

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

</body>
</html>
```

Example 3

Result View Example

Global Symbols

Global symbols are shared across the entire program using `Symbol.for()`:

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

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

<h4>Global Symbols</h4>
<p id="demo"></p>

<script>
// Regular symbols - always different
const s1 = Symbol("id");
const s2 = Symbol("id");

// Global symbols - same key returns same symbol
const g1 = Symbol.for("id");
const g2 = Symbol.for("id");

let text = "Regular symbols:<br>";
text += "s1 === s2: " + (s1 === s2) + "<br><br>";

text += "Global symbols (Symbol.for):<br>";
text += "g1 === g2: " + (g1 === g2) + "<br><br>";

text += "Symbol.keyFor(g1): " + Symbol.keyFor(g1) + "<br>";

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

</body>
</html>
```

Example 4

Result View Example

Document

Document in project

You can Download PDF file.

Reference

- W3Schools JavaScript Symbol Datatype