JavaScript Object Accessors

Back to JS page

Table of Contents

- JavaScript Object Accessors
  - JavaScript Getter (The get Keyword)
    - Example 1
  - JavaScript Setter (The set Keyword)
    - Example 2
  - JavaScript Function or Getter?
    - Example 3
  - Data Quality
    - Example 4
  - Object.defineProperty()
    - Example 5
  - Document
  - Reference

JavaScript Getter (The get Keyword)

JavaScript accessors (getters and setters) allow you to define object properties as functions.

The `get` keyword binds an object property to a function that will be called when that property is looked up:

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

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

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

<script>
const person = {
  firstName: "John",
  lastName: "Doe",
  get fullName() {
    return this.firstName + " " + this.lastName;
  }
};

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

</body>
</html>
```

Example 1

Result View Example

JavaScript Setter (The set Keyword)

The `set` keyword binds an object property to a function that will be called when an attempt is made to set that property:

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

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

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

<script>
const person = {
  firstName: "John",
  lastName: "Doe",
  language: "",
  set lang(lang) {
    this.language = lang;
  }
};

person.lang = "English";
document.getElementById("demo").innerHTML = person.language;
</script>

</body>
</html>
```

Example 2

Result View Example

JavaScript Function or Getter?

Getters give you a simpler syntax. Compare a getter vs a regular function:

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

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

<h4>Function vs Getter</h4>
<p id="demo1"></p>
<p id="demo2"></p>

<script>
const person = {
  firstName: "John",
  lastName: "Doe",
  fullName: function() {
    return this.firstName + " " + this.lastName;
  },
  get name() {
    return this.firstName + " " + this.lastName;
  }
};

// Using function - needs parentheses
document.getElementById("demo1").innerHTML = "Function: " + person.fullName();

// Using getter - no parentheses needed
document.getElementById("demo2").innerHTML = "Getter: " + person.name;
</script>

</body>
</html>
```

Example 3

Result View Example

Data Quality

Setters can be used to execute validation before setting a value:

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

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

<h4>Data Quality</h4>
<p id="demo"></p>

<script>
const person = {
  firstName: "John",
  lastName: "Doe",
  age: 0,
  set setAge(a) {
    if (a > 0 && a < 120) {
      this.age = a;
    } else {
      console.log("Invalid age");
    }
  }
};

person.setAge = 25;
document.getElementById("demo").innerHTML = "Age: " + person.age;
</script>

</body>
</html>
```

Example 4

Result View Example

Object.defineProperty()

The `Object.defineProperty()` method can also be used to define getters and setters:

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

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

<h4>Object.defineProperty()</h4>
<p id="demo"></p>

<script>
const person = {
  firstName: "John",
  lastName: "Doe"
};

Object.defineProperty(person, "fullName", {
  get: function() {
    return this.firstName + " " + this.lastName;
  },
  set: function(value) {
    const parts = value.split(" ");
    this.firstName = parts[0];
    this.lastName = parts[1];
  }
});

// Use the getter
document.getElementById("demo").innerHTML = "Before: " + person.fullName + "<br>";

// Use the setter
person.fullName = "Jane Smith";
document.getElementById("demo").innerHTML += "After: " + person.fullName;
</script>

</body>
</html>
```

Example 5

Result View Example

Document

Document in project

You can Download PDF file.

Reference

- W3Schools JavaScript Object Accessors