JavaScript Validation API

Back to JS page

Table of Contents

- JavaScript Validation API
  - Constraint Validation DOM Methods
    - Example 1
  - Constraint Validation DOM Properties
    - Example 2
  - Validity Properties
    - Example 3
  - The rangeOverflow Property
    - Example 4
  - The rangeUnderflow Property
    - Example 5
  - Document
  - Reference

Constraint Validation DOM Methods

The Constraint Validation API provides methods for validating form elements:

Method	Description
`checkValidity()`	Returns true if an element contains valid data
`setCustomValidity()`	Sets a custom validation message

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

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

<h4>Constraint Validation DOM Methods</h4>

<input type="text" id="id1" required minlength="3" placeholder="Enter at least 3 chars">
<button onclick="validate()">Check Validity</button>
<p id="demo"></p>

<script>
function validate() {
  const input = document.getElementById("id1");

  if (input.checkValidity()) {
    document.getElementById("demo").innerHTML = "Input is valid!";
  } else {
    document.getElementById("demo").innerHTML = "Input is invalid. " + input.validationMessage;
  }
}
</script>

</body>
</html>
```

Example 1

Result View Example

Constraint Validation DOM Properties

Form elements have these validation properties:

Property	Description
`validity`	Returns a ValidityState object
`validationMessage`	Returns the validation message
`willValidate`	Returns true if element will be validated

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

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

<h4>Constraint Validation DOM Properties</h4>

<input type="email" id="email" placeholder="Enter email address" required>
<button onclick="checkEmail()">Check Email</button>
<p id="demo2"></p>

<script>
function checkEmail() {
  const input = document.getElementById("email");
  let text = "Validation Properties:<br><br>";

  text += "willValidate: " + input.willValidate + "<br>";
  text += "validationMessage: " + input.validationMessage + "<br>";

  if (input.validity) {
    text += "validity.valid: " + input.validity.valid + "<br>";
  }

  if (input.value === "") {
    text += "<br>Please enter an email address.";
  }

  document.getElementById("demo2").innerHTML = text;
}
</script>

</body>
</html>
```

Example 2

Result View Example

Validity Properties

The `validity` object contains several boolean properties:

Property	Description
`valueMissing`	Element has no value (when required)
`typeMismatch`	Value doesn't match type (e.g., email)
`patternMismatch`	Value doesn't match pattern
`tooLong`	Value is too long
`tooShort`	Value is too short
`rangeUnderflow`	Value is below minimum
`rangeOverflow`	Value is above maximum
`stepMismatch`	Value doesn't match step
`badInput`	Value is invalid
`customError`	Custom validation error
`valid`	Element's value is valid

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

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

<h4>Validity Properties</h4>

<form id="myForm">
  <input type="text" id="username" required minlength="3" maxlength="10" placeholder="Username (
  <br><br>
  <input type="number" id="age" min="18" max="120" placeholder="Age (18–120)">
  <br><br>
  <input type="email" id="email2" placeholder="Email address">
  <br><br>
  <button type="button" onclick="showValidity()">Check All Fields</button>
</form>

<p id="demo3"></p>

<script>
function showValidity() {
  const fields = ["username", "age", "email2"];
  let text = "";

  for (let id of fields) {
    const input = document.getElementById(id);
    const v = input.validity;

    text += "<b>" + id + "</b>: ";
    if (v.valid) {
      text += "✓ Valid";
    } else {
      text += "✗ Invalid: ";
      if (v.valueMissing) text += "Required, ";
      if (v.typeMismatch) text += "Wrong type, ";
      if (v.patternMismatch) text += "No pattern match, ";
      if (v.tooShort) text += "Too short, ";
      if (v.tooLong) text += "Too long, ";
      if (v.rangeUnderflow) text += "Below minimum, ";
      if (v.rangeOverflow) text += "Above maximum, ";
      if (v.stepMismatch) text += "Wrong step, ";
      if (v.customError) text += "Custom error, ";
    }
    text += "<br>";
  }

  document.getElementById("demo3").innerHTML = text;
}
```

```
</script>

</body>
</html>
```

Example 3

Result View Example

The rangeOverflow Property

The `rangeOverflow` property is true when a value exceeds the `max` attribute:

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

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

<h4>The rangeOverflow Property</h4>

<input type="number" id="quantity" max="100" placeholder="Max: 100">
<button onclick="checkRangeOverflow()">Check</button>
<p id="demo4"></p>

<script>
function checkRangeOverflow() {
  const input = document.getElementById("quantity");
  const value = input.value;

  if (value === "") {
    document.getElementById("demo4").innerHTML = "Please enter a value.";
    return;
  }

  if (input.validity.rangeOverflow) {
    document.getElementById("demo4").innerHTML =
      "Value " + value + " is too high! Max is " + input.max;
  } else {
    document.getElementById("demo4").innerHTML =
      "Value " + value + " is within range. Valid!";
  }
}
</script>

</body>
</html>
```

Example 4

Result View Example

The rangeUnderflow Property

The `rangeUnderflow` property is true when a value is less than the `min` attribute:

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

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

<h4>The rangeUnderflow Property</h4>

<input type="number" id="score" min="50" placeholder="Min: 50">
<button onclick="checkRangeUnderflow()">Check</button>
<p id="demo5"></p>

<script>
function checkRangeUnderflow() {
  const input = document.getElementById("score");
  const value = input.value;

  if (value === "") {
    document.getElementById("demo5").innerHTML = "Please enter a value.";
    return;
  }

  if (input.validity.rangeUnderflow) {
    document.getElementById("demo5").innerHTML =
      "Value " + value + " is too low! Min is " + input.min;
  } else {
    document.getElementById("demo5").innerHTML =
      "Value " + value + " is within range. Valid!";
  }
}
</script>

</body>
</html>
```

Example 5

Result View Example

Document

Document in project

You can Download PDF file.

Reference

- W3Schools JavaScript Validation API