

# JavaScript 验证 API

## 约束验证 DOM 方法

Property	Description
checkValidity()	如果 input 元素中的数据是合法的返回 true，否则返回 false。
setCustomValidity()	<p>设置 input 元素的 validationMessage 属性，用于自定义错误提示信息的方法。</p> <p>使用 setCustomValidity 设置了自定义提示后，validity.customError 就会变成true，则 checkValidity 总是会返回false。如果要重新判断需要取消自定义提示，方式如下：</p>

```
setCustomValidity('')
setCustomValidity(null)
setCustomValidity(undefined)
```

以下实例如果输入信息不合法，则返回错误信息：

### checkValidity() 方法

```
<input id="id1" type="number" min="100" max="300" required>
<button onclick="myFunction()">验证</button>
<p id="demo"></p>
<script>
function myFunction() {
var inpObj = document.getElementById("id1");
if (inpObj.checkValidity() == false) {
document.getElementById("demo").innerHTML = inpObj.validationMessage;
}
}
</script>
```

尝试一下 »

## 约束验证 DOM 属性

属性	描述
validity	布尔属性值，返回 input 输入值是否合法
validationMessage	浏览器错误提示信息
willValidate	指定 input 是否需要验证

# Validity 属性

input 元素的 **validity** 属性包含一系列关于 validity 数据属性:

属性	描述
customError	设置为 true, 如果设置了自定义的 validity 信息。
patternMismatch	设置为 true, 如果元素的值不匹配它的模式属性。
rangeOverflow	设置为 true, 如果元素的值大于设置的最大值。
rangeUnderflow	设置为 true, 如果元素的值小于它的最小值。
stepMismatch	设置为 true, 如果元素的值不是按照规定的 step 属性设置。
tooLong	设置为 true, 如果元素的值超过了 maxLength 属性设置的长度。
typeMismatch	设置为 true, 如果元素的值不是预期相匹配的类型。
valueMissing	设置为 true , 如果元素 (required 属性) 没有值。
valid	设置为 true , 如果元素的值是合法的。

## 实例

如果输入的值大于 100 , 显示一个信息 :

### rangeOverflow 属性

```
<input id="id1" type="number" max="100">
<button onclick="myFunction()">验证</button>
<p id="demo"></p>
<script>
function myFunction() {
var txt = "";
if (document.getElementById("id1").validity.rangeOverflow) {
txt = "输入的值太大了";
}
document.getElementById("demo").innerHTML = txt;
}
</script>
```

尝试一下 »

如果输入的值小于 100 , 显示一个信息 :

### rangeUnderflow 属性

```
<input id="id1" type="number" min="100" required>
<button onclick="myFunction()">OK</button>
```

```
<p id="demo"></p>
<script>
function myFunction() {
var txt = "";
var inpObj = document.getElementById("id1");
if(!isNumeric(inpObj.value)) {
txt = "你输入的不是数字";
} else if (inpObj.validity.rangeUnderflow) {
txt = "输入的值太小了";
} else {
txt = "输入正确";
}
document.getElementById("demo").innerHTML = txt;
}
// 判断输入是否为数字
function isNumeric(n) {
return !isNaN(parseFloat(n)) && isFinite(n);
}
</script>
```

[尝试一下 »](#)[← JavaScript 表单](#)[JavaScript HTML DOM 集合\(Collection\) →](#)

## 2 篇笔记

[写笔记](#)

setCustomValidity 的用法：

```
var inpObj = document.getElementById("id1");
inpObj.setCustomValidity(''); // 取消自定义提示的方式
if (inpObj.checkValidity() == false) {
    if(inpObj.value==""){
        inpObj.setCustomValidity("不能为空! ");
    }else if(inpObj.value<100 || inpObj.value>300){
        inpObj.setCustomValidity("请重新输入数值 (100~300之间)!");
    }
    document.getElementById("demo").innerHTML = inpObj.validationMessage;
} else {
    document.getElementById("demo").innerHTML = "输入正确";
}
```

[尝试一下 »](#)

不爱李磊的韩梅梅 2年前 (2017-05-24)



可能是我理解能力比较差，没看懂 checkValidity()。

这个函数具体的意思是什么，后来查阅到了一个比较靠谱的答案。留下证据，以备像我一样笨的后人浪费时间。

**HTMLSelectElement.checkValidity()** 会检查元素是否有任何输入约束条件，并且检查值是否符合约束条件。如果值是不符合约束条件的，浏览器就会在该元素上触发一个可以撤销的 `invalid` 事件。初步理解为。该函数里面有两个值，默认判断值为 `true`，可以修改为 `false`。如有不对，希望指正。以下是我测试的代码：

```
function myFunction() {  
  var x = document.getElementById("nu");  
  x.setCustomValidity(""); // 使用前先取消自定义，否则下次点击checkValidity总返false  
  if (x.checkValidity() == false) {  
    x.setCustomValidity("错误");  
    document.getElementById("demo").innerHTML = x.validationMessage;  
  }  
  else {  
    x.setCustomValidity("正确");  
    document.getElementById("demo").innerHTML = x.validationMessage;  
  }  
}
```

尝试一下 »

萌新 2个月前 (01-29)