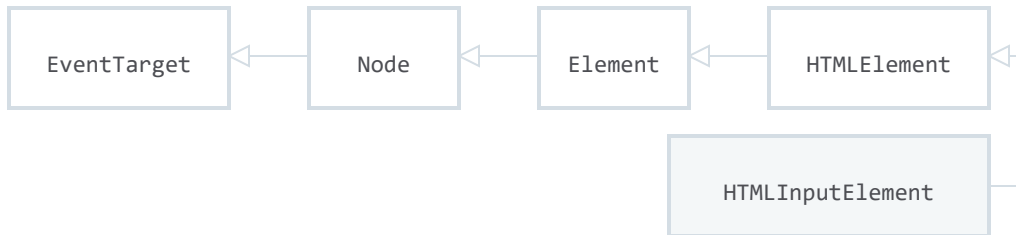


English ▼

# HTMLInputElement

The `HTMLInputElement` interface provides special properties and methods for manipulating the options, layout, and presentation of `<input>` elements.




## Properties

### Properties related to the parent form

<code>form</code> Read only	<code>HTMLFormElement</code> object: <b>Returns</b> a reference to the parent <code>&lt;form&gt;</code> element.
<code>formAction</code>	<code>string</code> : <b>Returns / Sets</b> the element's <code>formaction</code> attribute, containing the URI of a program that processes information submitted by the element. This overrides the <code>action</code> attribute of the parent form.
<code>formEnctype</code>	<code>string</code> : <b>Returns / Sets</b> the element's <code>formenctype</code> attribute, containing the type of content that is used to submit the form to the server. This overrides the <code>enctype</code> attribute of the parent form.
<code>formMethod</code>	<code>string</code> : <b>Returns / Sets</b> the element's <code>formmethod</code> attribute, containing the HTTP method that the browser uses to submit the form. This overrides the <code>method</code> attribute of the parent form.
<code>formNoValidate</code>	<code>Boolean</code> : <b>Returns / Sets</b> the element's <code>formnovalidate</code> attribute, indicating that the form is not to be validated when it is submitted. This overrides the <code>novalidate</code> attribute of the parent form.

formTarget	<i>string</i> : <b>Returns / Sets</b> the element's <code>formtarget</code> attribute, containing a name or keyword indicating where to display the response that is received after submitting the form. This overrides the <code>target</code> attribute of the parent form.
------------	---

Properties that apply to any type of input element that is not hidden

name	<i>string</i> : <b>Returns / Sets</b> the element's <code>name</code> attribute, containing a name that identifies the element when submitting the form.
type	<i>string</i> : <b>Returns / Sets</b> the element's <code>type</code> attribute, indicating the type of control to display. See <code>type</code> attribute of <code>&lt;input&gt;</code> for possible values.
disabled	<i>Boolean</i> : <b>Returns / Sets</b> the element's <code>disabled</code> attribute, indicating that the control is not available for interaction. The input values will not be submitted with the form. See also <code>readonly</code>
autofocus	<i>Boolean</i> : <b>Returns / Sets</b> the element's <code>autofocus</code> attribute, which specifies that a form control should have input focus when the page loads, unless the user overrides it, for example by typing in a different control. Only one form element in a document can have the <code>autofocus</code> attribute. It cannot be applied if the <code>type</code> attribute is set to <code>hidden</code> (that is, you cannot automatically set focus to a hidden control).
required	<i>Boolean</i> : <b>Returns / Sets</b> the element's <code>required</code> attribute, indicating that the user must fill in a value before submitting a form.
value	<i>string</i> : <b>Returns / Sets</b> the current value of the control. <div>  <b>Note:</b> If the user enters a value different from the value expected, this may return an empty string. </div>
validity <div>Read only</div>	<i>ValidityState object</i> : <b>Returns</b> the element's current validity state.
validationMessage <div>Read only</div>	<i>string</i> : <b>Returns</b> a localized message that describes the validation constraints that the control does not satisfy (if any). This is the empty string if the control is not a candidate for constraint validation ( <code>willvalidate</code> is <code>false</code> ), or it satisfies its constraints. This value can be set by the <code>setCustomValidity</code> method.
willValidate <div>Read only</div>	<i>Boolean</i> : <b>Returns</b> whether the element is a candidate for constraint validation. It is <code>false</code> if any conditions bar it from constraint validation, including: its <code>type</code> is <code>hidden</code> , <code>reset</code> , or <code>button</code> ; it has a <code>&lt;datalist&gt;</code> ancestor; its <code>disabled</code> property is <code>true</code> .

Properties that apply only to elements of type `checkbox` or `radio`

<code>checked</code>	<i>Boolean</i> : <b>Returns / Sets</b> the current state of the element when <code>type</code> is <code>checkbox</code> or <code>radio</code> .
<code>defaultChecked</code>	<i>Boolean</i> : <b>Returns / Sets</b> the default state of a radio button or checkbox as originally specified in HTML that created this object.
<code>indeterminate</code>	<i>Boolean</i> : <b>Returns</b> whether the checkbox or radio button is in indeterminate state. For checkboxes, the effect is that the appearance of the checkbox is obscured/greyed in some way as to indicate its state is indeterminate (not checked but not unchecked). Does not affect the value of the <code>checked</code> attribute, and clicking the checkbox will set the value to false.

Properties that apply only to elements of type `image`

<code>alt</code>	<i>string</i> : <b>Returns / Sets</b> the element's <code>alt</code> attribute, containing alternative text to use when <code>type</code> is <code>image</code> .
<code>height</code>	<i>string</i> : <b>Returns / Sets</b> the element's <code>height</code> attribute, which defines the height of the image displayed for the button, if the value of <code>type</code> is <code>image</code> .
<code>src</code>	<i>string</i> : <b>Returns / Sets</b> the element's <code>src</code> attribute, which specifies a URI for the location of an image to display on the graphical submit button, if the value of <code>type</code> is <code>image</code> ; otherwise it is ignored.
<code>width</code>	<i>string</i> : <b>Returns / Sets</b> the document's <code>width</code> attribute, which defines the width of the image displayed for the button, if the value of <code>type</code> is <code>image</code> .

Properties that apply only to elements of type `file`

<code>accept</code>	<i>string</i> : <b>Returns / Sets</b> the element's <code>accept</code> attribute, containing comma-separated list of file types accepted by the server when <code>type</code> is <code>file</code> .
<code>allowdirs</code> ⚠	<i>Boolean</i> : Part of the non-standard Directory Upload API; <b>indicates</b> whether or not to allow directories and files both to be selected in the file list. Implemented only in Firefox and is hidden behind a preference.
<code>files</code>	<b>Returns/accepts</b> a <code>FileList</code> object, which contains a list of <code>File</code> objects representing the files selected for upload.
<code>webkitdirectory</code> ⚠	<i>Boolean</i> : <b>Returns</b> the <code>webkitdirectory</code> attribute; if true, the file system picker interface only accepts directories instead of files.
<code>webkitEntries</code> ⚠	Array of <code>FileSystemEntry</code> objects: <b>Describes</b> the currently selected files or directories.


Properties that apply only to text/number-containing elements

autocomplete	<i>string</i> : <b>Returns / Sets</b> the element's <code>autocomplete</code> attribute, indicating whether the value of the control can be automatically completed by the browser. Ignored if the value of the <code>type</code> attribute is <code>hidden</code> , <code>checkbox</code> , <code>radio</code> , <code>file</code> , or a button type ( <code>button</code> , <code>submit</code> , <code>reset</code> , <code>image</code> ). Possible values are: <code>on</code> : the browser can autocomplete the value using previously stored value <code>off</code> : the user must explicitly enter a value
max	<i>string</i> : <b>Returns / Sets</b> the element's <code>max</code> attribute, containing the maximum (numeric or date-time) value for this item, which must not be less than its minimum ( <code>min</code> attribute) value.
maxLength	<i>Long</i> : <b>Returns / Sets</b> the element's <code>maxlength</code> attribute, containing the <b>maximum number of characters</b> (in Unicode code points) that the value can have. (If you set this to a negative number, an exception will be thrown.)
min	<i>string</i> : <b>Returns / Sets</b> the element's <code>min</code> attribute, containing the minimum (numeric or date-time) value for this item, which must not be greater than its maximum ( <code>max</code> attribute) value.
minLength	<i>Long</i> : <b>Returns / Sets</b> the element's <code>minlength</code> attribute, containing the <b>minimum number of characters</b> (in Unicode code points) that the value can have. (If you set this to a negative number, an exception will be thrown.)
pattern	<i>string</i> : <b>Returns / Sets</b> the element's <code>pattern</code> attribute, containing a <b>regular expression</b> that the control's value is checked against. Use the <code>title</code> attribute to describe the pattern to help the user. This attribute applies when the value of the <code>type</code> attribute is <code>text</code> , <code>search</code> , <code>tel</code> , <code>url</code> or <code>email</code> ; otherwise it is ignored.
placeholder	<i>string</i> : <b>Returns / Sets</b> the element's <code>placeholder</code> attribute, containing a hint to the user of what can be entered in the control. The placeholder text must not contain carriage returns or line-feeds. This attribute applies when the value of the <code>type</code> attribute is <code>text</code> , <code>search</code> , <code>tel</code> , <code>url</code> or <code>email</code> ; otherwise it is ignored.
readOnly	<i>boolean</i> : <b>Returns / Sets</b> the element's <code>readonly</code> attribute, indicating that the user cannot modify the value of the control. <code>HTML5</code> This is ignored if the value of the <code>type</code> attribute is <code>hidden</code> , <code>range</code> , <code>color</code> , <code>checkbox</code> , <code>radio</code> , <code>file</code> , or a button type.
selectionStart	<i>unsigned Long</i> : <b>Returns / Sets</b> the beginning index of the selected text. When nothing is selected, this returns the position of the text input cursor (caret) inside of the <code>&lt;input&gt;</code> element.
selectionEnd	<i>unsigned Long</i> : <b>Returns / Sets</b> the end index of the selected text. When there's no selection, this returns the offset of the character immediately following the

	current text input cursor position.
selectionDirection	<p><i>string</i>: <b>Returns / Sets</b> the direction in which selection occurred. Possible values are:</p> <p>forward if selection was performed in the start-to-end direction of the current locale</p> <p>backward for the opposite direction</p> <p>none if the direction is unknown</p>
size	<p><i>unsigned Long</i>: <b>Returns / Sets</b> the element's <i>size</i> attribute, containing <b>visual size of the control</b>. This value is in pixels unless the value of <i>type</i> is <i>text</i> or <i>password</i>, in which case, it is an integer number of characters. Applies only when <i>type</i> is set to <i>text</i>, <i>search</i>, <i>tel</i>, <i>url</i>, <i>email</i>, or <i>password</i>; otherwise it is ignored.</p>

### Properties not yet categorized

defaultValue	<i>string</i> : <b>Returns / Sets</b> the default value as originally specified in the HTML that created this object.
dirName	<i>string</i> : <b>Returns / Sets</b> the directionality of the element.
accessKey	<i>string</i> : <b>Returns</b> a string containing a single character that switches input focus to the control when pressed.
list <div>Read only</div>	<i>HTMLElement object</i> : <b>Returns</b> the element pointed by the <i>list</i> attribute. The property may be <i>null</i> if no HTML element found in the same tree.
multiple	<i>Boolean</i> : <b>Returns / Sets</b> the element's <i>multiple</i> attribute, indicating whether more than one value is possible (e.g., multiple files).
files	<i>FileList array</i> : <b>Returns</b> the list of selected files.
HTMLInputElement.labels <div>Read only</div>	<i>NodeList array</i> : <b>Returns</b> a list of <i>&lt;label&gt;</i> elements that are labels for this element.
step	<p><i>string</i>: <b>Returns / Sets</b> the element's <i>step</i> attribute, which works with <i>min</i> and <i>max</i> to limit the increments at which a numeric or date-time value can be set. It can be the string <i>any</i> or a positive floating point number. If this is not set to <i>any</i>, the control accepts only values at multiples of the step value greater than the minimum.</p>
valueAsDate	<i>Date object</i> : <b>Returns / Sets</b> the value of the element, interpreted as a date, or <i>null</i> if conversion is not possible.
valueAsNumber	<i>double</i> : <b>Returns</b> the value of the element, interpreted as one of the

	following, in order: <ul style="list-style-type: none"> <li>• A time value</li> <li>• A number</li> <li>• <code>NaN</code> if conversion is impossible</li> </ul>
<code>autocapitalize</code> 	<i>string</i> : <b>Defines</b> the capitalization behavior for user input. Valid values are <code>none</code> , <code>off</code> , <code>characters</code> , <code>words</code> , or <code>sentences</code> .
<code>inputmode</code>	Provides a hint to browsers as to the type of virtual keyboard configuration to use when editing this element or its contents.

### HTMLInputElement.align

*string*: **Represents** the alignment of the element. *Use CSS instead.*

### HTMLInputElement.useMap

*string*: **Represents** a client-side image map.

## Methods

<code>blur()</code>	Removes focus from the input element; keystrokes will subsequently go nowhere.
<code>click()</code>	Simulates a click on the input element.
<code>focus()</code>	Focuses on the input element; keystrokes will subsequently go to this element.
<code>select()</code>	Selects all the text in the input element, and focuses it so the user can subsequently replace all of its content.
<code>setSelectionRange()</code>	Selects a range of text in the input element (but does not focus it).
<code>setRangeText()</code>	Replaces a range of text in the input element with new text.
<code>setCustomValidity()</code>	Sets a custom validity message for the element. If this message is not the empty string, then the element is suffering from a custom validity error, and does not validate.
<code>checkValidity()</code>	Returns a <code>Boolean</code> that is <code>false</code> if the element is a candidate for constraint validation, and it does not satisfy its constraints. In this case, it also fires an <code>invalid</code> event at the element. It returns <code>true</code> if the element is not a candidate for constraint validation, or if it satisfies its constraints.

`reportValidity()`

Runs the `checkValidity()` method, and if it returns false (for an invalid input or no pattern attribute provided), then it reports to the user that the input is invalid in the same manner as if you submitted a form.

### `HTMLInputElement.stepDown()`

Decrements the `value` by (`step` \* n), where n defaults to 1 if not specified. Throws an `INVALID_STATE_ERR` exception:

- if the method is not applicable to for the current `type` value,
- if the element has no `step` value,
- if the `value` cannot be converted to a number,
- if the resulting value is above the `max` or below the `min`.

### `HTMLInputElement.stepUp()`

Increments the `value` by (`step` \* n), where n defaults to 1 if not specified. Throws an `INVALID_STATE_ERR` exception:

- if the method is not applicable to for the current `type` value.,
- if the element has no `step` value,
- if the `value` cannot be converted to a number,
- if the resulting value is above the `max` or below the `min`.

### `HTMLInputElement.mozSetFileArray()` ⚠

Sets the files selected on the input to the given array of `File` objects. This is an alternative to `mozSetFileNameArray()` which can be used in frame scripts: a chrome script can [open files as File objects](#) and send them via [message manager](#).

### `HTMLInputElement.mozGetFileNameArray()` ⚠

Returns an array of all the file names from the input.

### `HTMLInputElement.mozSetFileNameArray()` ⚠

Sets the filenames for the files selected on the input. Not for use in [frame scripts](#), because it accesses the file system.

---

## Events

Listen to these events using `addEventListener()` or by assigning an event listener to the `oneventname` property of this interface:

input

Fires when the value of an <input>, <select>, or <textarea> element has been changed. Note that this is actually fired on the HTMLElement interface and also applies to contenteditable elements, but we've listed it here because it is most commonly used with form input elements.

Also available via the oninput event handler property.

invalid

Fired when an element does not satisfy its constraints during constraint validation.

Also available via the oninvalid event handler property.

search

Fired when a search is initiated on an <input> of type="search".

Also available via the onsearch event handler property.



Specifications

Specification	Status	Comment
HTML Living Standard The definition of 'HTMLInputElement' in that specification.	<div><div>LS</div>Living Standard</div>	



Specification	Status	Comment
<a href="#">HTML5</a> The definition of 'HTMLInputElement' in that specification.	<div>REC</div> Recommendation	<p>Technically, the <code>tabindex</code> and <code>accesskey</code> properties, as well as the <code>blur()</code>, <code>click()</code>, and <code>focus()</code> methods, are now defined on <a href="#">HTMLElement</a>.</p> <p>The following properties are now obsolete: <code>align</code> and <code>useMap</code>.</p> <p>The following properties have been added: <code>autocomplete</code>, <code>autofocus</code>, <code>dirName</code>, <code>files</code>, <code>formAction</code>, <code>formEncctype</code>, <code>formMethod</code>, <code>formNoValidate</code>, <code>formTarget</code>, <code>height</code>, <code>indeterminate</code>, <code>labels</code>, <code>list</code>, <code>max</code>, <code>min</code>, <code>multiple</code>, <code>pattern</code>, <code>placeholder</code>, <code>required</code>, <code>selectionDirection</code>, <code>selectionEnd</code>, <code>selectionStart</code>, <code>step</code>, <code>validationMessage</code>, <code>validity</code>, <code>valueAsDate</code>, <code>valueAsNumber</code>, <code>width</code>, and <code>willValidate</code>.</p> <p>The following methods have been added: <code>checkValidity()</code>, <code>setCustomValidity()</code>, <code>setSelectionRange()</code>, <code>stepUp()</code>, and <code>stepDown()</code>.</p>
<a href="#">Document Object Model (DOM) Level 2 HTML Specification</a> The definition of 'HTMLInputElement' in that specification.	Obsolete	<p>The <code>size</code> property is now an <code>unsigned long</code>. The <code>type</code> property must be entirely given in lowercase characters.</p>
<a href="#">Document Object Model (DOM) Level 1 Specification</a> The definition of 'HTMLInputElement' in that specification.	Obsolete	Initial definition.

## Browser compatibility

[Update compatibility data on GitHub](#)

Chrome	1
Edge	12
Firefox	1
IE	5.5
Opera	8
Safari	3
WebView Android	1
Chrome Android	18
Firefox Android	4
Opera Android	10.1
Safari iOS	1
Samsung Internet Android	1.0
autocomplete	
Chrome	14
Edge	12
Firefox	4
IE	10
Opera	≤12.1
Safari	6
WebView Android	≤37
Chrome Android	18
Firefox Android	4
Opera Android	≤12.1
Safari iOS	6
Samsung Internet Android	1.0
autofocus	
Chrome	Yes
Edge	12

Firefox	Yes
IE	Yes
Opera	Yes
Safari	Yes
WebView Android	Yes
Chrome Android	Yes
Firefox Android	Yes
Opera Android	Yes
Safari iOS	Yes
Samsung Internet Android	Yes
checkValidity	
Chrome	4
Edge	12
Firefox	4
IE	10
Opera	≤12.1
Safari	5
WebView Android	≤37
Chrome Android	18
Firefox Android	Yes
Opera Android	≤12.1
Safari iOS	5
Samsung Internet Android	1.0
files	
Chrome	1
Edge	12
Firefox	3
IE	10★

Opera	≤12.1
Safari	4
WebView Android	1
Chrome Android	18
Firefox Android	Yes
Opera Android	≤12.1
Safari iOS	3.2
Samsung Internet Android	1.0

formAction	
Chrome	9
Edge	12
Firefox	4
IE	10
Opera	≤12.1
Safari	5.1
WebView Android	≤37
Chrome Android	18
Firefox Android	Yes
Opera Android	≤12.1
Safari iOS	5.1
Samsung Internet Android	1.0

formEncType	
Chrome	Yes
Edge	12
Firefox	4
IE	Yes
Opera	Yes
Safari	Yes

WebView Android	Yes
Chrome Android	Yes
Firefox Android	?
Opera Android	?
Safari iOS	?
Samsung Internet Android	Yes
formMethod	
Chrome	9
Edge	12
Firefox	4
IE	10
Opera	≤12.1
Safari	5.1
WebView Android	≤37
Chrome Android	18
Firefox Android	Yes
Opera Android	≤12.1
Safari iOS	5.1
Samsung Internet Android	1.0
formNoValidate	
Chrome	4
Edge	12
Firefox	4
IE	10
Opera	≤12.1
Safari	5.1
WebView Android	≤37
Chrome Android	18

Firefox Android	Yes
Opera Android	≤12.1
Safari iOS	5.1
Samsung Internet Android	1.0

formTarget	
Chrome	9
Edge	12
Firefox	4
IE	10
Opera	≤12.1
Safari	5.1

WebView Android	≤37
Chrome Android	18
Firefox Android	Yes
Opera Android	≤12.1
Safari iOS	5.1
Samsung Internet Android	1.0

height	
Chrome	21
Edge	12
Firefox	16
IE	5.5
Opera	≤12.1
Safari	6.1

WebView Android	≤37
Chrome Android	25
Firefox Android	Yes
Opera Android	≤12.1

Safari iOS	6.1
Samsung Internet Android	1.5
indeterminate	
Chrome	1
Edge	12
Firefox	3
IE	5.5
Opera	≤12.1
Safari	3
WebView Android	1
Chrome Android	18
Firefox Android	4
Opera Android	≤12.1
Safari iOS	1
Samsung Internet Android	1.0
invalid event	
Chrome	10
Edge	12
Firefox	4
IE	10
Opera	10
Safari	5
WebView Android	4
Chrome Android	18
Firefox Android	64
Opera Android	12
Safari iOS	5
Samsung Internet Android	4.0

labels	
Chrome	6
Edge	18
Firefox	56
IE	No
Opera	≤12.1
Safari	5.1
WebView Android	≤37
Chrome Android	18
Firefox Android	Yes
Opera Android	≤12.1
Safari iOS	5.1
Samsung Internet Android	1.0
list	
Chrome	20
Edge	12
Firefox	4
IE	10
Opera	≤12.1
Safari	12.1
WebView Android	≤37
Chrome Android	25
Firefox Android	4
Opera Android	≤12.1
Safari iOS	12.2
Samsung Internet Android	1.5
multiple	
Chrome	2
Edge	12



Firefox	3.6
IE	10
Opera	≤12.1
Safari	4
WebView Android	≤37
Chrome Android	18
Firefox Android	Yes
Opera Android	≤12.1
Safari iOS	3.2
Samsung Internet Android	1.0

onsearch

⚠

Chrome	Yes
Edge	79
Firefox	No
IE	No
Opera	Yes
Safari	Yes
WebView Android	Yes
Chrome Android	Yes
Firefox Android	No
Opera Android	?
Safari iOS	?
Samsung Internet Android	Yes

pattern

Chrome	4
Edge	12
Firefox	4
IE	10

Opera	≤12.1
Safari	5
WebView Android	≤37
Chrome Android	18
Firefox Android	Yes
Opera Android	≤12.1
Safari iOS	5
Samsung Internet Android	1.0

placeholder	
Chrome	3
Edge	12
Firefox	4
IE	10
Opera	≤12.1
Safari	4
WebView Android	≤37
Chrome Android	18
Firefox Android	Yes
Opera Android	≤12.1
Safari iOS	3.2
Samsung Internet Android	1.0

reportValidity	
Chrome	40
Edge	17
Firefox	49
IE	No
Opera	27
Safari	10.1

WebView Android	40
Chrome Android	40
Firefox Android	64
Opera Android	27
Safari iOS	10.3
Samsung Internet Android	4.0

required

Chrome	4
Edge	12
Firefox	4
IE	10
Opera	≤12.1
Safari	5

WebView Android	≤37
Chrome Android	18
Firefox Android	Yes
Opera Android	≤12.1
Safari iOS	5
Samsung Internet Android	1.0

search event

⚠

Chrome	Yes
Edge	79
Firefox	No
IE	No
Opera	Yes
Safari	Yes
WebView Android	Yes
Chrome Android	Yes


Firefox Android	No
Opera Android	?
Safari iOS	?
Samsung Internet Android	Yes

selectionDirection	
Chrome	15
Edge	13
Firefox	8
IE	No
Opera	≤12.1
Safari	6
WebView Android	Yes
Chrome Android	Yes
Firefox Android	8
Opera Android	≤12.1
Safari iOS	6
Samsung Internet Android	Yes


setCustomValidity	
Chrome	4
Edge	12
Firefox	4
IE	10
Opera	≤12.1
Safari	5
WebView Android	≤37
Chrome Android	18
Firefox Android	Yes
Opera Android	≤12.1

Safari iOS	5
Samsung Internet Android	1.0
setRangeText	
Chrome	24
Edge	79
Firefox	27
IE	No
Opera	15
Safari	6.1
WebView Android	Yes
Chrome Android	25
Firefox Android	Yes
Opera Android	14
Safari iOS	7
Samsung Internet Android	1.5
setSelectionRange	
Chrome	1
Edge	12
Firefox	1
IE	9
Opera	8
Safari	3
WebView Android	1
Chrome Android	18
Firefox Android	4
Opera Android	10.1
Safari iOS	1
Samsung Internet Android	1.0

stepDown

Chrome	5
Edge	12
Firefox	16★ 
IE	10
Opera	≤12.1
Safari	5
WebView Android	≤37
Chrome Android	18
Firefox Android	Yes
Opera Android	≤12.1
Safari iOS	5
Samsung Internet Android	1.0

stepUp

Chrome	5
Edge	12
Firefox	16★ 
IE	10
Opera	≤12.1
Safari	5
WebView Android	≤37
Chrome Android	18
Firefox Android	Yes
Opera Android	≤12.1
Safari iOS	5
Samsung Internet Android	1.0

validationMessage

Chrome	5
--------	---

Edge	12
Firefox	4
IE	10
Opera	≤12.1
Safari	5
WebView Android	≤37
Chrome Android	18
Firefox Android	Yes
Opera Android	≤12.1
Safari iOS	5
Samsung Internet Android	1.0

validity

Chrome	3
Edge	12
Firefox	4
IE	10
Opera	≤12.1
Safari	5
WebView Android	≤37
Chrome Android	18
Firefox Android	Yes
Opera Android	≤12.1
Safari iOS	5
Samsung Internet Android	1.0

webkitdirectory

⚠

Chrome	6
Edge	13
Firefox	50

IE	No
Opera	15
Safari	11.1
WebView Android	≤37
Chrome Android	18
Firefox Android	50
Opera Android	14
Safari iOS	11.3
Samsung Internet Android	1.0

webkitEntries

⚠

Chrome	22
Edge	79
Firefox	50
IE	No
Opera	15
Safari	11.1
WebView Android	≤37
Chrome Android	25
Firefox Android	50
Opera Android	14
Safari iOS	11.3
Samsung Internet Android	1.5

weight

Chrome	Yes
Edge	≤18
Firefox	16
IE	?
Opera	?






Safari	?
WebView Android	Yes
Chrome Android	Yes
Firefox Android	?
Opera Android	?
Safari iOS	?
Samsung Internet Android	Yes
willValidate	
Chrome	1
Edge	12
Firefox	4
IE	10
Opera	≤12.1
Safari	4
WebView Android	1
Chrome Android	18
Firefox Android	Yes
Opera Android	≤12.1
Safari iOS	3.2
Samsung Internet Android	1.0

What are we missing?

- ..

Full support
- Partial support
- No support
- ..

Compatibility unknown

-  Non-standard. Expect poor cross-browser support.
  -  See implementation notes.
  -  User must explicitly enable this feature.
- 

## See also

- HTML element implementing this interface: `<input>`
- 

 **Last modified:** Nov 24, 2020, by [MDN contributors](#)

## Related Topics

### HTMLInputElement

#### ▼ Properties

`labels`

`multiple`

 `webkitdirectory`

 `webkitEntries`

#### ▼ Methods

 `mozGetFileNameArray()`

`select()`

`setRangeText()`

`setSelectionRange()`

`stepDown()`

`stepUp()`

#### ▼ Events

`invalid`

 `search`

▼ Inheritance:

HTMLElement

Element

Node

EventTarget

▼ Related pages for HTML DOM

BeforeUnloadEvent

DOMStringMap

ErrorEvent

GlobalEventHandlers

HTMLAnchorElement

HTMLAreaElement

HTMLAudioElement

HTMLBRElement

HTMLBaseElement

HTMLBaseFontElement

HTMLBodyElement

HTMLButtonElement

HTMLCanvasElement

HTMLContentElement

HTMLDListElement

HTMLDataElement

HTMLDataListElement

HTMLDialogElement

HTMLDivElement

HTMLDocument

HTMLElement

HTMLEmbedElement

HTMLFieldSetElement

HTMLFormControlsCollection

HTMLFormElement

HTMLFrameSetElement

HTMLHRElement  
HTMLHeadElement  
HTMLHeadingElement  
HTMLHtmlElement  
HTMLIFrameElement  
HTMLImageElement  
**HTMLIsIndexElement**  
HTMLKeygenElement  
HTMLLIElement  
HTMLLabelElement  
HTMLLegendElement  
HTMLLinkElement  
HTMLMapElement  
HTMLMediaElement  
HTMLMetaElement  
HTMLMeterElement  
HTMLModElement  
HTMLOListElement  
HTMLObjectElement  
HTMLOptGroupElement  
HTMLOptionElement  
HTMLOptionsCollection  
HTMLOutputElement  
HTMLParagraphElement  
HTMLParamElement  
HTMLPictureElement  
HTMLPreElement  
HTMLProgressElement  
HTMLQuoteElement  
HTMLScriptElement  
HTMLSelectElement  
HTMLShadowElement  
HTMLSourceElement  
HTMLSpanElement

HTMLStyleElement

HTMLTableCaptionElement

HTMLTableCellElement

HTMLTableColElement

HTMLTableDataCellElement

HTMLTableElement

HTMLTableHeaderCellElement

HTMLTableRowElement

HTMLTableSectionElement

HTMLTemplateElement

HTMLTextAreaElement

HTMLTimeElement

HTMLTitleElement

HTMLTrackElement

HTMLULListElement

HTMLUnknownElement

HTMLVideoElement

HashChangeEvent

History

ImageData

Location

MessageChannel

MessageEvent

MessagePort

Navigator

NavigatorGeolocation

NavigatorID

NavigatorLanguage

NavigatorOnLine

NavigatorPlugins

PageTransitionEvent

Plugin

PluginArray

PopStateEvent

PortCollection

PromiseRejectionEvent

RadioNodeList

Transferable

ValidityState

Window

WindowBase64

WindowEventHandlers

WindowTimers