

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

<div>form</div> <div>Read only</div>	<i>HTMLFormElement</i> object: Returns a reference to the parent <code><form></code> element.
<div>formAction</div>	<i>string</i> : Returns / Sets 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.
<div>formEncType</div>	<i>string</i> : Returns / Sets 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.
<div>formMethod</div>	<i>string</i> : Returns / Sets 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.
<div>formNoValidate</div>	<i>Boolean</i> : Returns / Sets 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.
<div>formTarget</div>	<i>string</i> : Returns / Sets 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

<div>name</div>	<i>string</i> : Returns / Sets the element's <code>name</code> attribute, containing a name that identifies the element when submitting the form.
<div>type</div>	<i>string</i> : Returns / Sets the element's <code>type</code> attribute, indicating the type of control to display. See <code>type</code> attribute of <code><input></code> for possible values.
<div>disabled</div>	<i>Boolean</i> : Returns / Sets 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>
<div>autofocus</div>	<i>Boolean</i> : Returns / Sets 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).
<div>required</div>	<i>Boolean</i> : Returns / Sets the element's <code>required</code> attribute, indicating that the user must fill in a value before submitting a form.
<div>value</div>	<i>string</i> : Returns / Sets the current value of the control. <div>Note: If the user enters a value different from the value expected, this may return an empty string.</div>
<div>validity</div> <div>Read only</div>	<i>ValidityState</i> object: Returns the element's current validity state.
<div>validationMessage</div> <div>Read only</div>	<i>string</i> : Returns 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	Read only	<i>Boolean</i> : Returns 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><datalist></code> ancestor; its <code>disabled</code> property is <code>true</code> .
--------------	-----------	--

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

checked	<i>Boolean</i> : Returns / Sets the current state of the element when <code>type</code> is <code>checkbox</code> or <code>radio</code> .
defaultChecked	<i>Boolean</i> : Returns / Sets the default state of a radio button or checkbox as originally specified in HTML that created this object.
indeterminate	<i>Boolean</i> : Returns 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 <code>value</code> to <code>false</code> .

Properties that apply only to elements of type `image`

alt	<i>string</i> : Returns / Sets the element's <code>alt</code> attribute, containing alternative text to use when <code>type</code> is <code>image</code> .
height	<i>string</i> : Returns / Sets 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> .
src	<i>string</i> : Returns / Sets 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.
width	<i>string</i> : Returns / Sets 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`

accept	<i>string</i> : Returns / Sets 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> .
allowdirs	<i>Boolean</i> : Part of the non-standard Directory Upload API; indicates 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.
files	Returns/accepts a <code>FileList</code> object, which contains a list of <code>File</code> objects representing the files selected for upload.
webkitdirectory	<i>Boolean</i> : Returns the <code>webkitdirectory</code> attribute; if true, the file system picker interface only accepts directories instead of files.
webkitEntries	<i>Array of FileSystemEntry objects</i> : Describes the currently selected files or directories.

Properties that apply only to `text/number`-containing or elements

autocomplete	<i>string</i> : Returns / Sets 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: on: the browser can autocomplete the value using previously stored value off: the user must explicitly enter a value
max	<i>string</i> : Returns / Sets 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> : Returns / Sets the element's <code>maxlength</code> attribute, containing the maximum number of characters (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> : Returns / Sets 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> : Returns / Sets the element's <code>minlength</code> attribute, containing the minimum number of characters (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> : Returns / Sets the element's <code>pattern</code> attribute, containing a regular expression 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> : Returns / Sets 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> : Returns / Sets the element's <code>readonly</code> attribute, indicating that the user cannot modify the value of the control. <div>HTML5 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 <code>button</code> type.</div>
selectionStart	<i>unsigned Long</i> : Returns / Sets 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><input></code> element.
selectionEnd	<i>unsigned Long</i> : Returns / Sets 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	<i>string</i> : Returns / Sets the direction in which selection occurred. Possible values are: <code>forward</code> if selection was performed in the start-to-end direction of the current locale <code>backward</code> for the opposite direction <code>none</code> if the direction is unknown
size	<i>unsigned Long</i> : Returns / Sets the element's <code>size</code> attribute, containing visual size of the control . This value is in pixels unless the value of <code>type</code> is <code>text</code> or <code>password</code> , in which case, it is an integer number of characters. Applies only when <code>type</code> is set to <code>text</code> , <code>search</code> , <code>tel</code> , <code>url</code> , <code>email</code> , or <code>password</code> ; otherwise it is ignored.

Properties not yet categorized

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

valueAsNumber	<i>double</i> : Returns the value of the element, interpreted as one of the following, in order: <ul style="list-style-type: none"> • A time value • A number • NaN if conversion is impossible
autocapitalize	<i>string</i> : Defines 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> .
inputmode	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

blur()	Removes focus from the input element; keystrokes will subsequently go nowhere.
click()	Simulates a click on the input element.
focus()	Focuses on the input element; keystrokes will subsequently go to this element.
select()	Selects all the text in the input element, and focuses it so the user can subsequently replace all of its content.
setSelectionRange()	Selects a range of text in the input element (but does not focus it).
setRangeText()	Replaces a range of text in the input element with new text.
setCustomValidity()	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.
checkValidity()	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 <code>checkValidity()</code> method, and if it returns <code>false</code> (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 HTMLInputElement 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.	Living Standard	
HTML5 The definition of 'HTMLInputElement' in that specification.	Recommendation	Technically, the tabIndex and accessKey properties, as well as the blur(), click(), and focus() methods, are not part of the HTML5 specification. The following properties are now obsolete: align and useMap. The following properties have been added: autocomplete, autofocus, dirName, files, formAction, formEncType, formMethod, formNoValidate, and willValidate. The following methods have been added: checkValidity(), setCustomValidity(), setSelectionRange(), etc.
Document Object Model (DOM) Level 2 HTML Specification The definition of 'HTMLInputElement' in that specification.	Obsolete	The size property is now an unsigned long. The type property must be entirely given in lowercase characters.
Document Object Model (DOM) Level 1 Specification The definition of 'HTMLInputElement' in that specification.	Obsolete	Initial definition.

