

HTMLElement: change event



Baseline Widely available



The `change` event is fired for `<input>`, `<select>`, and `<textarea>` elements when the user modifies the element's value. Unlike the `input` event, the `change` event is not necessarily fired for each alteration to an element's `value`.

Depending on the kind of element being changed and the way the user interacts with the element, the `change` event fires at a different moment:

- When a `<input type="checkbox">` element is checked or unchecked (by clicking or using the keyboard);
- When a `<input type="radio">` element is checked (but not when unchecked);
- When the user commits the change explicitly (e.g., by selecting a value from a `<select>`'s dropdown with a mouse click, by selecting a date from a date picker for `<input type="date">`, by selecting a file in the file picker for `<input type="file">`, etc.);
- When the element loses focus after its value was changed: for elements where the user's interaction is typing rather than selection, such as a `<textarea>` or the `text`, `search`, `url`, `tel`, `email`, or `password` types of the `<input>` element.

The HTML specification lists the `<input>` types that should fire the `change` event [↗](#).

Syntax

Use the event name in methods like `addEventListener()`, or set an event handler property.

JS

```
addEventListener("change", (event) => { })
```

```
onchange = (event) => { }
```

Event type

A generic `Event`.

Examples

<select> element

HTML

HTML

```
<label>
  Choose an ice cream flavor:
  <select class="ice-cream" name="ice-cream">
    <option value="">Select One ...</option>
    <option value="chocolate">Chocolate</option>
    <option value="sardine">Sardine</option>
    <option value="vanilla">Vanilla</option>
  </select>
</label>

<div class="result"></div>
```

JavaScript

JS

```
const selectElement = document.querySelector(".ice-cream");
const result = document.querySelector(".result");

selectElement.addEventListener("change", (event) => {
  result.textContent = `You like ${event.target.value}`;
});
```

Result

Choose an ice cream flavor:

Text input element

For some elements, including `<input type="text">`, the `change` event doesn't fire until the control loses focus. Try entering something into the field below, and then click somewhere else to trigger the event.

HTML

```
<input placeholder="Enter some text" name="name" />
<p id="log"></p>
```

JavaScript

```
const input = document.querySelector("input");
const log = document.getElementById("log");

input.addEventListener("change", updateValue);

function updateValue(e) {
  log.textContent = e.target.value;
}
```

Result

Specifications

Specification

HTML

event-change [↗](#)

```
# handler-onchange ↗
```

Browser compatibility

[Report problems with this compatibility data](#) [↗](#) • [View data on GitHub](#) [↗](#)

Browser compatibility

[Report problems with this compatibility data](#)  • [View data on GitHub](#) 

[illegible]

	 Chrome	 Edge	 Firefox	 Opera	 Safari	 Chrome Android	 Firefox for Android	 Opera Android	 Safari on iOS	 Samsung Internet	 WebView Android	 WebView on iOS
change event	✓ 1	✓ 12	✓ 1	✓ 9	✓ 3	✓ 18	✓ 4	✓ 10.1	✓ 2	✓ 1	✓ 4.4	✓ 2

✓ Full support

Different browsers do not always agree whether a `change` event should be fired for certain types of interaction. For example, keyboard navigation in `<select>` elements used to never fire a `change` event in Gecko until the user hit Enter or switched the focus away from the `<select>` (see [Firefox bug 126379](#)). Since Firefox 63 (Quantum), this behavior is consistent between all major browsers, however.



Your blueprint for a better internet.