mdn web docs_

Window: load event

Baseline Widely available

The `load` event is fired when the whole page has loaded, including all dependent resources such as stylesheets, scripts, iframes, and images, except those that are loaded lazily. This is in contrast to `DOMContentLoaded`, which is fired as soon as the page DOM has been loaded, without waiting for resources to finish loading.

This event is not cancelable and does not bubble.

> **Note:** *All events named `load` will not propagate to `Window`*, even with `bubbles` initialized to `true`. To catch `load` events on the `window`, that `load` event must be dispatched directly to the `window`.

> **Note:** The `load` event that is dispatched when the main document has loaded *is* dispatched on the `window`, but has two mutated properties: `target` is `document`, and `path` is `undefined`. These two properties are mutated due to legacy conformance.

Syntax

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

JS

```js
addEventListener("load", (event) => { })

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

Event type

A generic `Event`.

Examples

Log a message when the page is fully loaded:

JS

```js
window.addEventListener("load", (event) => {
  console.log("page is fully loaded");
});
```

The same, but using the `onload` event handler property:

JS

```js
window.onload = (event) => {
  console.log("page is fully loaded");
};
```

Live example

HTML

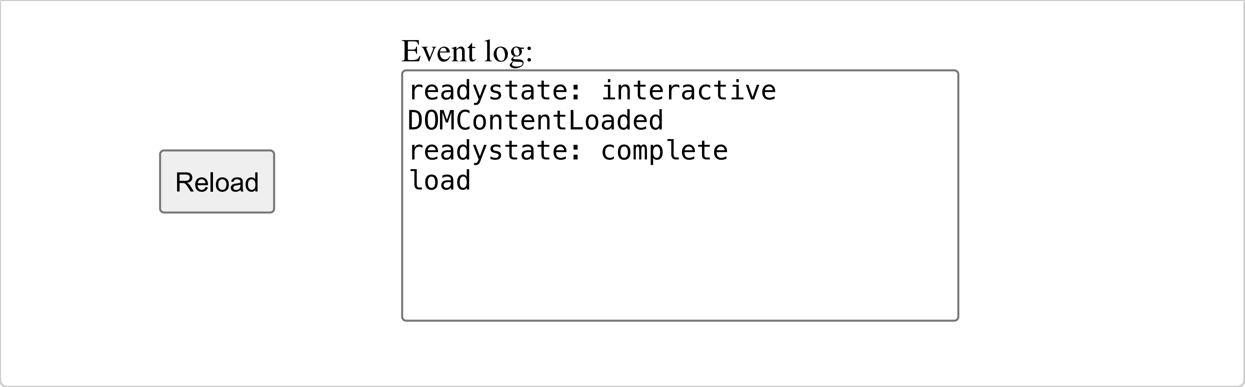
HTML Play

```html
<div class="controls">
  <button id="reload" type="button">Reload</button>
</div>

<div class="event-log">
  <label for="eventLog">Event log:</label>
  <textarea
    readonly
    class="event-log-contents"
    rows="8"
    cols="30"
    id="eventLog"></textarea>
</div>
```

JavaScript

```js
const log = document.querySelector(".event-log-contents");
const reload = document.querySelector("#reload");

reload.addEventListener("click", () => {
  log.textContent = "";
  setTimeout(() => {
    window.location.reload(true);
  }, 200);
});

window.addEventListener("load", (event) => {
  log.textContent += "load\n";
});

document.addEventListener("readystatechange", (event) => {
  log.textContent += `readystate: ${document.readyState}\n`;
});

document.addEventListener("DOMContentLoaded", (event) => {
  log.textContent += `DOMContentLoaded\n`;
});
```

Result



Event log:

```
readystate: interactive
DOMContentLoaded
readystate: complete
load
```

Specifications

Specification
UI Events # event-type-load