Reacting to Page Events

This lesson is about the type of page events that are occurred by interacting with a page.

WE'LL COVER THE FOLLOWING

- Page Loading
- Page Closing

Page Loading

Depending on how complex it is, a web page can take time to be entirely loaded by the browser. You can add an event listener on the load event produced by the window object (which represents the browser window) to know when this happens. This avoids messy situations where JavaScript interacts with pages that aren't fully loaded.

The following code displays a message in the console once the page is fully loaded.

JavaScript

```
// Web page loading event
window.addEventListener("load", e => {
  console.log("The page has been loaded!");
});
```

Console

Clear

```
The page has been loaded!
```

Page Closing

You sometimes want to react to page closing. Closing happens when the user closes the tab displaying the page or navigates to another page in this tab. A frequent use case consists of showing a confirmation dialog to the user. Handling page closing is done by adding a handler for the `beforeunload` event on the `window` object.

JavaScript

```
// Handle page closing
window.addEventListener("beforeunload", e => {
  const message = "Should you stay or should you go?";
  // Standard way of showing a confirmation dialog
  e.returnValue = message;
  // Browser-specific way of showing a confirmation dialog
  return message;
});
```

Console

Clear

> Setting the value of the `returnValue` property on the `Event` object is the standard way of triggering a confirmation dialog showing this value. However, some browsers use the return value of the event listener instead. The previous code associate the two techniques to be universal.