JS Event Handling

What is an event?

- A JavaScript event is a callback that gets fired when something happens related to the Document Object Model on your website
- For instance, when you scroll, an element is clicked or hovered over, etc.
- We can intercept these events by attaching a "callback" function to them, known as "event handlers"
- Event handlers are given an "event" object argument.

Listening for events - native JS

```
// Define our button log function.
function buttonLog(event) {
  console.log("You clicked the button!");
  console.log(event);
}

// Grab the element for use in JS. Refer to the
// "Dom Manipulation" slides for more info.
var button = document.getElementById("my-button");

// attach an event handler - a function to
// execute - when the button is clicked.
button.addEventListener("click", buttonLog);
```

Anonymous functions

- Not all functions need to be defined and named before they are referenced
- Functions that are not are called "anonymous functions"

```
var button = document.getElementById("my-button");

button.addEventListener("click", function(event) {
  console.log("You clicked the button!");
  console.log(event);
});
```

What's in the `event` object?

- Handler functions are passed an object with data and controls for the event.
- All events share some properties
- Some events have special properties
- Events also come with functions that can alter the event

The `event` object - Shared props

- `target` - What the user interacted with
- `currentTarget` - What element the event was bound to, usually the same as `target`
- `type` - What type of event it was, same as `addEventListener` argument
- `timeStamp` - When the event happened
- And many more

The `event` object - Unique props

- Mouse events like `click` and `mousemove` provide the mouse's X and Y coordinates
- Keyboard events like `keydown` and `keyup` provide which key was pressed
- The `scroll` event provides how much scrolling was done, and in what direction

The `event` object - Functions

- Not only can we get information about events, we can alter them
- `preventDefault()` will stop the event from doing its default browser behavior
- `stopPropogation()` will prevent parent elements from triggering the event

Special events

- Events can happen on things other than just DOM elements, or on special elements
- Images and other assets have loading-related events like `onload` and `onerror`
- `window` can listen for events related to the window like `resize`, `scroll`, `onpopstate` (back button) and more
- `document` can listen for events related to the document like `DOMContentLoaded` (when the page is ready to show)

Exercise

Create an event that listens for the click on an <h1> element

Have it alert whatever text is inside the the <h1>

Resources

Codecademy

Make an Interactive Website - Events

TeamTreeHouse

Interactive Web Pages with JavaScript - Selecting Elements and Adding Events with JavaScript