# JavaScript Events

 (https://github.com/learn-co-curriculum/phase-0-javascript-events)  (https://github.com/learn-co-curriculum/phase-0-javascript-events/issues/new)

## Learning Goals

- Define a JavaScript event
- Identify different types of user events

## Introduction

We've experimented with selecting and manipulating nodes in the DOM using JavaScript: deleting nodes, editing nodes, etc. But most web applications are *not* used by people opening up the console and editing the DOM using Chrome's DevTools. Instead, people *do something* and then *work happens*.

"Doing work" in response to "something happening" is known as *event handling*. *Events* are the "something the user does" and the "*callback function*" is the work that will happen in response to the event being triggered.

In this lesson we'll go over some of the most commonly-used JavaScript events. In the following lessons, we'll learn how to use *event listeners* to tell JavaScript which event or events we want it to listen for. We'll also learn how to implement *callback functions* to handle the *work happens* part of event handling.

## Define a JavaScript Event

JavaScript has the ability to "listen" for things that happen inside the browser. It can listen for events like whether the browser is resized, or whether someone clicked on a specific image on the screen. The event you're probably most familiar with is "click."

We'll go over a few of the more common types of events in this lesson.

## Identify Different Types of User Events

# Mouse Click

Mouse or trackpad events are some of the most common ones you'll be handling using JavaScript eventing. For example, JavaScript can recognize a single click on an element in the page and change the styling of the element to highlight it. Or it can recognize a double-click on an element and open a zoomed-in view of that element.

There are many other mouse events you can use; take a look at the list of JavaScript's **mouse events here (https://developer.mozilla.org/en-US/docs/Web/API/Element#mouse_events)** .

# Key Press

While click events will likely make up the majority of events you'll use, the keyboard is another important source of events. JavaScript currently includes two **keyboard events (https://developer.mozilla.org/en-US/docs/Web/API/Element#keyboard_events)** : `keydown` and `keyup` . (A third, `keypress` , has been deprecated.) When a key is pressed, these events provide a code to indicate which key it was. For example, a game program might listen for `keydown` events and, if the space bar was pressed, make the character jump over the hole.

# Form Submission

HTML pages often use a submit button to submit a form to a server. When a user submits a form, the `submit` event is fired. An event handler here might pop up a thank you overlay or log in the user and take them to their home page.

# Other Events

As you seek to build more complicated applications, you'll need to handle and trigger work on many more events than the few we've discussed in this lesson. Some other common events you are likely to encounter are `scroll` , `mouseenter` and `mouseleave` , `focus` , `blur` , and `onchange` .

One important thing to keep in mind is that not all JavaScript events are supported by all browsers. This **list of browser events (https://developer.mozilla.org/en-US/docs/Web/Events#standard_events)** includes the ones that can be used in most browsers.

# Conclusion

JavaScript allows us to trigger work when it detects events. You set up an event handler and, when JavaScript recognizes that event, it will execute the event handler's work, which is stored in a *callback function*.

Take a few minutes to look through the **list of common events (https://developer.mozilla.org/en-US/docs/Web/Events#standard_events)** to familiarize yourself with the many many ways you can use event handling to enhance your users' experience.

# Resources

- **MDN - Web Events (https://developer.mozilla.org/en-US/docs/Web/Events)**
- **MDN - Mouse Events (https://developer.mozilla.org/en-US/docs/Web/API/Element#mouse_events)**
- **MDN - Keyboard Events (https://developer.mozilla.org/en-US/docs/Web/API/Element#keyboard_events)**