

Event-Driven Programming

▶	Introduction	3 min
▶	Buttons with Divs	6 min
☰	Try It! Buttons with Divs	30 min
▶	Changing Pages Interactively	7 min
☰	Try It! Change Pages Interactively	30 min
▶	Using HTML5 Canvas	10 min
☰	Try It! Canvas	30 min
▶	Inputs and Events	8 min
☰	Try It! Inputs and Events	30 min
★	Practice Quiz: Event-Driven Programming	3 questions

Green Screen Web Page

Look at the following HTML code that specifies a button.

```
1 <input type="button" value="click me" onclick="runcode()">
```

What will happen when the button is clicked?

- ☐ The function **dochange()** will be called and whatever code is in that function will be executed.
- ☐ An alert box will appear.
- ☐ The function **runcode()** will be called and whatever code is in that function will be executed.

Skip

Submit

5:52 / 6:37

Buttons with Divs

Have a question? Discuss this lecture in the week forums.



Interactive Transcript

Search Transcript

English ▼

0:03

We'll start learning how to use JavaScript to interact with the content on a webpage by looking at simple examples of HTML and CSS, and then adding JavaScript to these simple web pages. By starting with a simple web page, you'll be able to understand how adding interactivity works. We'll start by creating a very simple web page and then adding JavaScript to it. The first version of the web page will only use HTML and CSS with two div elements styled by CSS using class IDs. We'll introduce a new HTML element, a button and we'll connect this button to functionality that eventually will interact with elements of the web page. Clicking the button will cause different actions determined by you, the program. You'll write code using JavaScript that will be able to control and interact with the web page. In the code you'll see in this lesson, you'll use a JavaScript function that executes when a button is pressed. Recall that a function is a piece of code that can perform complex steps and it's not invoked on an object. We'll start with this simple example that simply alerts the user when a button is clicked, but your JavaScript code will get more complex as you learn more. And based on these simple first examples, we'll build towards an understanding of how to combine HTML, CSS and JavaScript as well as how to import JavaScript libraries, so you won't have to write all the code yourself. You'll learn how to access different elements in a web page programmatically using JavaScript to change text, to change color, to add new elements and more.

1:36

Hopefully, CodePen looks familiar. HTML in left window, CSS in the middle window and JavaScript that's currently blank on the right. There are two divs in the HTML window with width and font size specified in the CSS window. The div with ID d1 is specified by CSS to have a background color of light blue and the div with ID d2 is specified to have a background color of yellow. This is the simple web page we'll begin to make interactive. Let's look at how to create an HTML button and program it to have an effect when it's pressed. We'll use a new HTML tag, the input tag. It's similar to the image or img tag you have seen before that displays media as an image, because this tag doesn't have both a start and an end tag, but rather one tag with options with in the tag. The first tag element is the type of input. In our first examples, we'll only use button as the type of input. But later, we'll see that you can use color choosers as a type of input, so that users can choose a color as part of interacting with a web page. And there are other types of inputs we'll see as well, including sliders, file uploaders and more. The text displayed in the button is specified by the value element in the input tag. As you can see here, the word change will be displayed in the button. You'll also need the onclick event attribute to tell the button what to in the case of a certain event. In this case, the user clicking the mouse. Other examples of common HTML events are when a web page is loaded or user has moused over text or an image or an input field is changed. For example, when you type in a password. The keyword onclick is an HTML event attribute that indicates the JavaScript that follows should react in event of a click. This calls an event handler, which specifies what will happen when the button is pressed or clicked. In this case, the onclick event will cause an alert to be displayed. Let's see what this looks like in CodePen.

3:46

As you can see, the CodePen environment pops up an alert box with a message in it.

3:54

The input tag is specified that the text of the message in the onclick event handler. The user will need to dismiss the alert box before more events can be handled. Let's take a look.

4:08

Here the user's clicking a button that causes the alert box to pop-up and then the user dismisses it, but what if the alert where in more complex set of instructions? Rather than using alert directly in the input tag, we'd like to be able to use JavaScript to handle the events generated by the button. We could do this by writing a JavaScript function, which is already done yourself We'll call this function dochange. In the body of the function, we'll show the alert with the text clicked button, then we can call the function for the onclick event. The HTML input creates the element that the user interacts with. And when the user clicks or presses the button, the event handler associated with the onclick event, the JavaScript is triggered or invoked. Here, the event handler is connected to the JavaScript code by calling the dochange function we just wrote. Let's see this in CodePen.

5:09

First, we write the function dochange. It has curly braces, because that's what we need in a function and then we'll copy and paste the alert code into that page.

5:22

That is the JavaScript function that we just wrote.

5:26

We'll change the text, so that we know it's a JavaScript function that's called. So in the onclick event handler, we'll call the function do change, but we'll edit the text in the alert box. Let's see what happens when the button is pressed.

5:45

Now, we can see that the event is different.

5:49

[This is a simple example of a good way to start.](#) We saw how to use an HTML element to allow a user to interact with a web page. We created a button, but we'll see later that there are other types of input. We saw how to create an event handler for the event of a user clicking on the button. Because JavaScript commands can sometimes be complex, we learned how to create a function that can be called by the event handler. You'll write JavaScript in the JS or JavaScript paneling CodePen, but you'll be able to use the knowledge you gain with HTML, CSS and JavaScript to create pages that you can be deployed, that you can deploy anywhere on the web not just within CodePen. And you'll be able to use your knowledge of loops, variables and its statements to interact and modify web pages. Have fun.

Downloads

Lecture Video mp4

Subtitles (English) WebVTT

Transcript (English) txt

Lecture Slides pdf

Would you like to [help us translate](#) the transcript and subtitles into additional languages?