

Back to Week 3

Lessons

PrevNext

Coursera

Catalog

Search catalog

Q

For Enterprise

Java

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

Two <div> ID's

```
<div id="d1" >
  Hello
</div>
<div id="d2" >
  Goodbye
</div>
<p>
<input type="button" value="color change"
  onclick = "changecolor()" >
</p>
```

Duke University

to execute the JavaScript code that follows.

Changing Pages Interactively

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

Interactive Transcript

Search Transcript

English

0:03

Let's look at how to change a web page interactively using JavaScript. This interactivity will be the basis for programming a green screen algorithm in a web page and for being creative in designing your own web page. To change a page, you'll need to write JavaScript code to access elements like h1 tags or div elements or other tagged elements in a page. It's possible to access all the elements with an h1 tag, for example, or with an li tag or any other HTML tag. It's also possible to access individual elements by an ID or a group of elements by a class. We'll mostly use IDs to access HTML elements since IDs are distinct and every element must have a distinct ID. In contrast, many elements can share the same class. In our examples, you'll see how to add a CSS class to an HTML element like a div. This will allow you to change the background color or the width or the height of an element. It's also possible to change the text displayed in an element or other features. Let's look at a simple example of changing a web page using interactive JavaScript. We'll use two CSS classes, each specify a different background color for a div. By clicking a button, you'll use JavaScript to assign a CSS class to each div, making the divs change appearance, and go from no background color to different background colors. The basic idea is to programmatically access each div using the IDs, and change the CSS associated with the div. We'll explore this at a high level and then look at the details. Here you saw the divs change color.

1:46

You may remember that each div has a different ID. One, which displays Hello, has the ID d1 as you can see here. The other, which displays Goodbye, has the ID d2.

2:00

Recall that the onclick event attribute tells the page to create an event handler [to execute the JavaScript code that follows](#). You'll be able to use JavaScript to access the div or whatever other element you tagged with the ID d1. That makes it possible to change the features of an element as we'll see. In this example, the JavaScript function is called changecolor. Let's take a look at that function by looking at the JavaScript.

2:29

The JavaScript function changecolor is shown here.

2:34

Remember that it's called when a user clicks on the button, that is, interacts with a web page. The JavaScript function is changecolor.

2:45

Remember that it's called when the user interacts with the web page because its function is called or invoked by the onclick event handler, which calls the function changecolor. The JavaScript method getElementById is used to access an HTML element using its associated ID. You may remember that a method performs an operation on an object and is proceeded by a dot. In this case, the method getElementById uses the HTML document to access its element. The label document refers to the entire HTML web page. The getElementById method has one parameter. Here, it's d1, the ID label associated with a particular HTML element.

3:33

The getElementById method returns this HTML element with the ID label that was passed as a parameter. That's the div element with the word Hello because the associated ID label is d1.

3:46

We need to create a variable to store this returned HTML element. Here, that's shown by var dd1. Now that variable can be used to access the element and change its color. The next line creates a similar variable called dd2. This stores the HTML element returned by document.getElementById when it's passed the parameter d2.

4:10

Next, to change the background colors used in this page, we need to set the background color for each div.

4:17

We'll do this by setting the CSS class for the div. The line of code dd1.className gets blueback is what changes the color of the div with the ID d1.

4:31

Remember, this div was stored in the variable dd1.

4:35

The attribute className is one of the characteristics of an HTML element that can be accessed in JavaScript. As we'll see in later examples, there are many other attributes of HTML elements that you can also change by accessing them with JavaScript. These attributes are sometimes called fields. They're accessed using the dot notation in a JavaScript program. The code assigns the className blueback as the class of the div stored in variable dd1.

5:05

As you can see in the CSS panel of the CodePen page that we've shown here, the class blueback has a specific background color. If other CSS characteristics were part of this class blueback, these characteristics would also be part of the div labeled d1 as a result of assigning the class blueback to the variable dd1 in JavaScript. The next line similarly assigns the class yellowback to the variable dd2, which is storing the HTML element with ID d2.

5:38

Again, in action, using the onclick event handler to call the JavaScript function changecolor assigns these CSS classes to the chosen HTML elements, and we see the background change. It's also possible to change other attributes of HTML elements like the text associated with an element. This function changetext uses a very similar approach to what we just saw with changecolor. The first two lines create variables to store HTML elements, and these are exactly the same lines as we just saw with changecolor. Just as before, to access or access attributes of an HTML element, you'll use the dot syntax and the name of an attribute or field that will be used in JavaScript. The attribute shown here .innerHTML accesses the HTML content within the element, in this case, that's everything inside the div, which is text. This line of code dd1.innerHTML gets Bonjour changes the text of the HTML element accessed by the variable dd1.

6:47

That's the same HTML element whose ID is d1 which used to show the word Hello before the code executed and will now show Bonjour after the code runs. It's worth pointing out that while dd1 is an HTML element, a div specifically in this case, in general, it's just an object like a CSS style or simple image are objects. You can always use documentation to find out what methods operate on different objects. In action, here you can see what happens when you click on the changetext button. An onclick event handler calls changetext, the JavaScript function we just saw, which accesses the HTML inside the divs and sets it to new text, Bonjour and Sayonara. We'll provide some resources to show how you access other attributes using JavaScript as you practice making web pages and changing them interactively. Bonjour and sayonara.

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?