Event-Driven Programming

Green Screen Web Page

Which of the following are event handlers that you saw used with a color picker in this video? Select all that are correct.

- [ ] onmouseover
  Un-selected is correct
- [x] onclick
  Correct
- [ ] alert
  Un-selected is correct
- [x] onchange
  Correct

Continue

Inputs and Events

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

Interactive Transcript

Search Transcript English

0:03
In this lesson, you're going to learn about new HTML input types and events. You've already seen the button input type with the onclick event. And now you will learn how to use additional input types of events to make your web pages even more interactive.

0:18
As you have seen, the HTML input tag is used to create an HTML element, like a button that gets input from the user and processes that input with JavaScript code.

0:30
You have already used a button to get input from the user. But you can also create web pages that ask for text input.

0:37
You've likely seen web pages in which you enter a name or a password or a URL. You can present the user with a color picker tool such as this one.

0:48
Or a range, which gets information from the slider. Both of which you'll see in this lesson.

0:55
There are many other options you can find in the documentation for the HTML input element.

1:02
In the case of a button, the input is the user clicking on the button. The onclick event. But as you'll see here, there are other events that you can connect to a button or another type of input. For example, here is an article on the Duke Today website. The content highlights when the mouse rolls over it. The on mouse enter and on mouse leave events are used to have the browser execute any function you connect when the mouse enters or leaves an object.

1:33
Changes to a field or input can also be useful events.

1:38
Here is an example of a search function executing on the Duke Today website, as user input is typed in to look for articles about Duke's Kunshan University. A color picker, or color chooser, allows the user to choose colors as part of interacting with the web page. The color can be specified by name, by using sliders, or selecting RGB values or more.

2:06
Note the color was introduced with HTML5. And new input types that are not supported by older web browsers, will behave simply as input text types.

2:19
Let's look more closely at how a color picker, or chooser, works.

2:24
Here is the input tag, which you have seen before. The type for our color chooser is color. And here we have specified a default value of dark blue.

2:36
This color picker element has an ID, so that we can refer to it in our JavaScript code, and represent it's input with a variable.

2:44
Many inputs require connecting events other than mouse clicks or moves. A color chooser uses an onchange event. This may be connected to a mouse but isn't based on clicks. Just when the color chooser's value changes. This event is connected to JavaScript through the onchange event handler.

3:05
Now let's look at the function docolor.

3:09
This function can be named anything but must match the name you specified in the HTML input element. Accessing the color picker value in the JavaScript code uses functions and concepts that you may recognize.

3:22
First, the canvas element is stored in variable dd1, as you have seen before.

3:29
Then, the color picker element is stored in the color input variable, using the same technique to find the element in the document. As before, you can choose any name you want for your variable names, but the IDs given in the strings must match those IDs given in the HTML elements. The value of the color chooser is accessed using the .value attribute, or field of the color chooser, HTML element stored in the variable color input. Next we use the .style attribute to set the background color of the canvas to whatever color the color picker is currently set to.

4:05
The basics are the same here as they are with a button input. Connect JavaScript code to an event using the HTML input element in the appropriate event handler.

4:17
Let's see the color picker in action. The make line button still works. Now, clicking or moving the pressed mouse button over the color options will change the color shown in the color chooser dialogue. Pressing down on the mouse will change the active color in the color chooser and generate an onchange event. This event is also generated if you keep the mouse pressed down and move it over the colored pencils.

4:44
When the mouse is down, a color change event can be generated.

4:49
What happens if we change the event we were responding to from onchange to onclick?

4:55
Now, clicking on the color picker with its default of dark blue changes the canvas background. Whereas it did not for the onchange event because the element had not been changed. In this case, changing the color chooser does not change the canvas color until we click on the color picker element itself. This is the difference between the onchange and onclick events for an HTML input element.

5:22
Let's look at one more type of HTML input and the events it uses. The range or slider input. Here, the slider is horizontal and has a label of text indicating its purpose is to make a square using a slider.

5:38
As a slider is moved, a square is drawn with side lengths determined by the value of the slider. The square can be made larger or smaller as the user drags the slider right or left.

5:51
The HTML to create this slider has an input type of range with three parameters you specify when creating the HTML element. The minimum value of the range. Here, that's 10. The maximum value of the range. Here, that's 100. And the current value when the slider is first rendered in a web page. Here that's 10, which happens to be the minimum value. So the slider will be first rendered all the way to the left.

6:21
The ID of the input element is important for accessing its value to use in the function dosquare. When you click and drag the slider, you generate an oninput event that you can use to connect the web page with your JavaScript. Let's look more closely at the function dosquare now.

6:40
This is the function dosquare which draws a square with a side length given by the range input.

6:47
As before, we store our reference to the canvas in a variable. Then, we use the ID for the slider element to create a variable sizeinput representing the slider.

6:59
We get the value of the slider element and store it in the variable size.

7:04
As you have seen before, we need to get the context of the canvas in order to draw in it.

7:09
Finally, we use the size input from the slider to draw a yellow rectangle starting at the coordinates 10, 10, whose side lengths are determined by the size variable. Let's see what this code does.

7:29
Hm, the square can be made larger with a slider but it doesn't seem to get smaller. This is because each time the oninput event handler calls the dosquare function, it draws another square, one on top of the other. But the old ones are not cleared.

7:48
Luckily, there is an attribute for the context that can clear our previous drawing. .clearRect will clear a rectangle, given four parameters. Two for the top left coordinate of the rectangle and then its width and height. For simplicity here, we will clear the entire canvas each time dosquare is called. That's better. Now the square resizes with the user input from the slider.

8:13
Now you know how to use new types of input, the color chooser, and the slider. And you know new types of events to use with them, onchange and oninput. Have fun as you continue to make interactive web pages.

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?