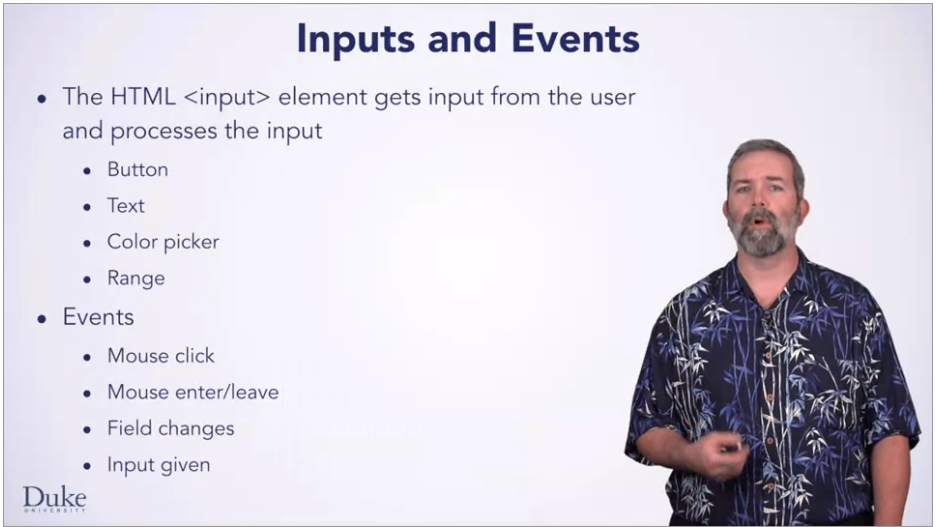
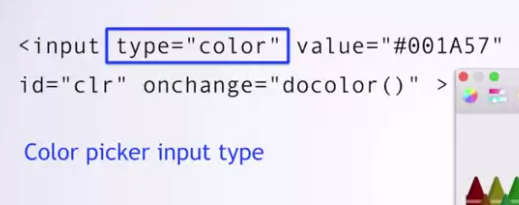
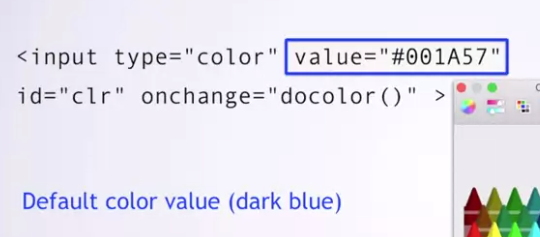
**Inputs and Events**

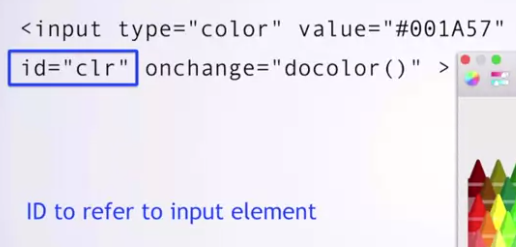


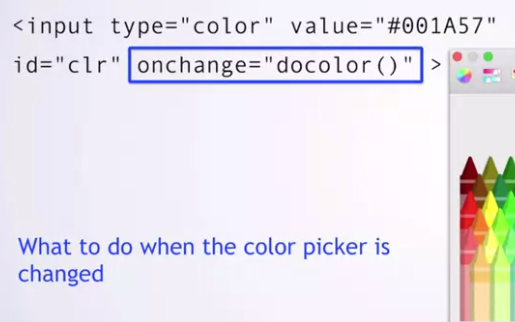
**Color Picker:**

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.

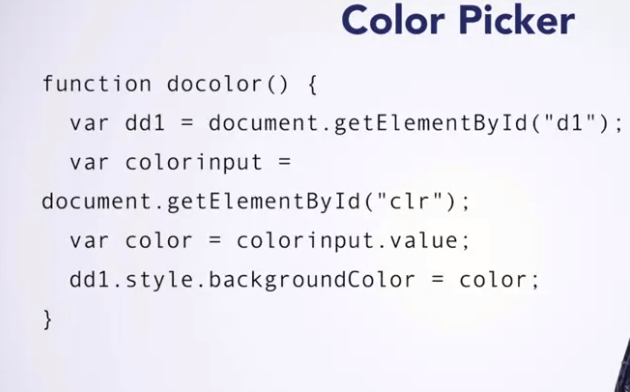








Onchange is an event-handler

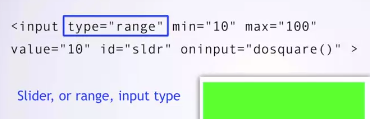


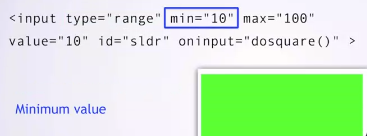
First, the canvas element is stored in variable dd1, as you have seen before.

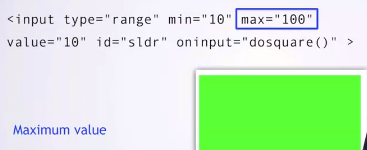
Then, the color picker element is stored in the color input variable, using the same technique to find the element in the document. As before, you can choose any name you want for your variable names, but the IDs given in the strings must match those IDs given in the HTML elements. The value of the color 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.

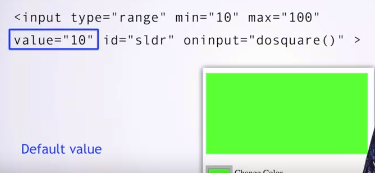
The basics are the same here as they are with a button input. Connect JavaScript code to an event using the HTML input element in the appropriate event handler.

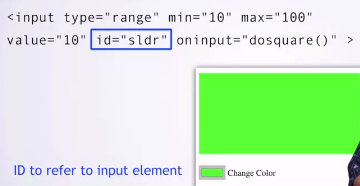
**Slider Input**

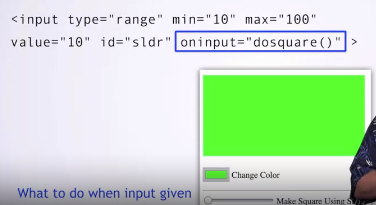


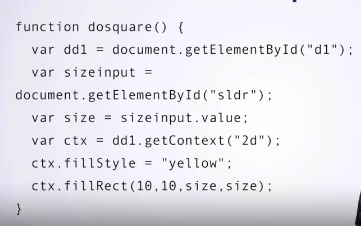


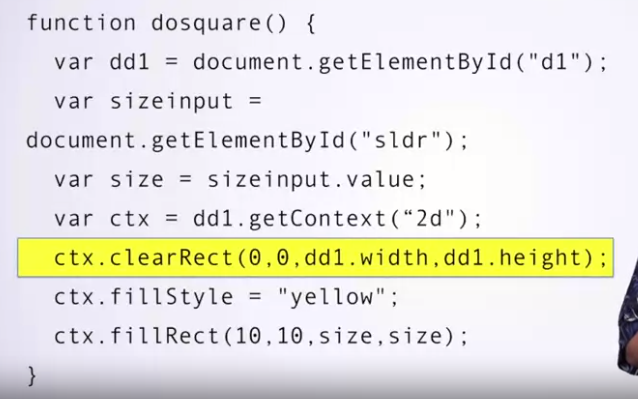












As before, we store our reference to the canvas in a variable.

Then, we use the ID for the slider element to create a variable sizeinput representing the slider.

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

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

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.

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.

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.

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.