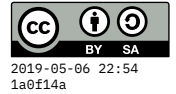


Project 3: jQuery color picker



Starting point

In class, I've been developing a simple color tool using jQuery. The latest code is in:

- jqdemo/colors.html
- jqdemo/colors.css
- jqdemo/colors.js

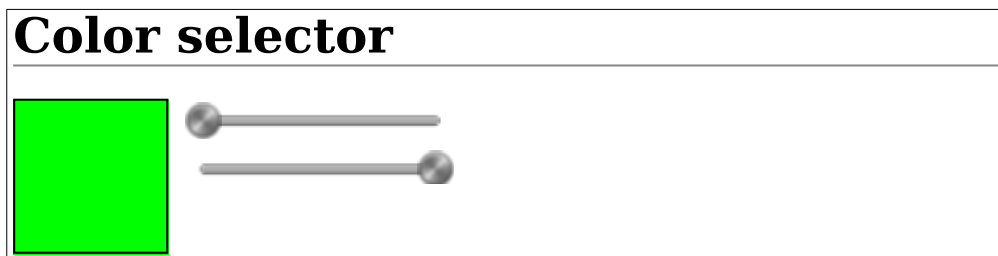


Figure 1: Initial color selector only has sliders for red/green

If you have previously cloned the `cs120pub`¹ folder, you can go to that directory in your terminal and request it to download any updates:

```
cd ~/Desktop/cs120pub
git pull
```

But make sure you make any changes in *your own* `cs120` folder, not in `cs120pub`.



¹gitlab.liu.edu/cs120s19/cs120pub

Requirements

Your task for this assignment is to extend the color selector as follows:

1. Add the slider corresponding to the blue channel, and make sure it works.
2. Make the `rgb` specification appear on the page itself, not just in the console.
3. Provide a palette of several predefined colors. Here is how I recommend starting it:

```
<div id="palette">
  <div style="background:#000000"></div>
  <div style="background:#ffffff"></div>
  <div style="background:#c25a5a"></div>
  <div style="background:#c2c25a"></div>
</div>
```

Then you can use CSS rules to make sure the child divs pile up left-to-right (float them to the left) and have a fixed width and height. The selector for that rule can be `div#palette > div`.

4. When our user clicks on a color in the palette, it *loads* that predefined color into the swatch (the larger box), **and** the sliders, **and** the `rgb(...)` display. One trick I used for this is to use `.css` to grab the background color property (which conveniently is already in the `rgb(...)` format) and then convert it from a string into a record containing the three components.

```
var rgb = getRGB($(this).css("background-color"))
red = rgb.red
green = rgb.green
blue = rgb.blue
```

where `getRGB` was from this StackOverflow answer²:

```
// https://stackoverflow.com/questions/34980574/how-to-extract-color-values
function getRGB(str){
  var match = str.match(
    /rgba?\((\d{1,3}), ?(\d{1,3}), ?(\d{1,3})\)(?:, ?(\d(?:\.\d?)\)|$)/
  );
  return match ? {
    red: match[1],
    green: match[2],
    blue: match[3]
  } : {};
}
```

5. Finally, when our user clicks on the swatch (the larger box), it **adds** that color to the palette! To construct a brand new element dynamically with jQuery, you can use syntax like this:

```
var elt = $("

</div>")
elt.css("background-color", "...")


```

Unfortunately, it will not pick up the click handler specified for previously-existing elements, so you will need to specify the click handler separately:

```
elt.click(...)
```

And then you can add it to the end of a parent element using:

```
$("#div#palette").append(elt)
```

Below is a screenshot of my solution.



²stackoverflow.com/a/34980657/288195

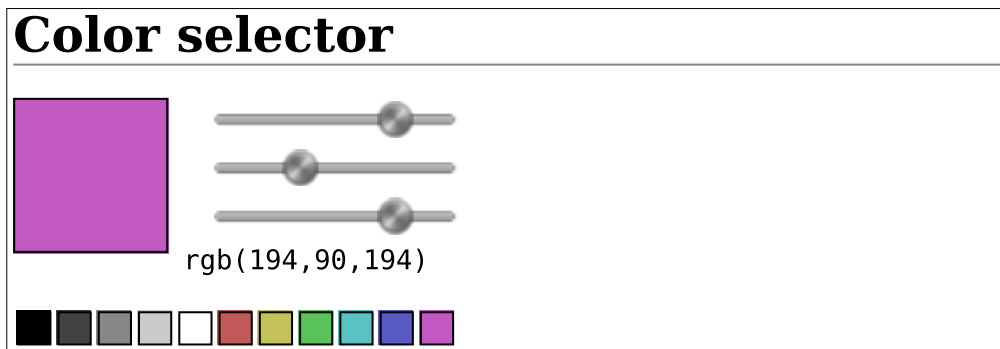


Figure 2: Color selector with palette