

Functions and the DOM Lab

In this lab we'll wire up our page to respond to some user events and refactor our page to make use of JavaScript functions. Let's start with some raw events.

Wiring up the buttons

Let's tell the user what kind of graph he/she is looking at and change that with each button click.

1. If you don't already have one, add an `<h2>` to the page like so:
`<h2 id='graphType'></h2>`
2. Create a new function called `drawSummary()`. This function should set your `graphType`'s description to "Summary", and clear the graph.
3. Wire up the summary button to call `drawSummary` on click. Here are some steps to take:
 - a. Get a reference to the button using `getElementById`. (hint: "var b = ...")
 - b. Add a click event sort of like this:

```
b.addEventListener("click", function () {
    drawSummary();
});
```
4. Run and test. Make sure your button calls `drawSummary()`.
5. Create a new function called `drawTrends()`. This function should set your `graphType`'s description to "Trends" and clear the canvas by drawing a big, white rectangle.
6. Wire up the trends button to call `drawTrends` when the user clicks it.
7. Run and test. Make sure the trends button works as expected.
8. Finally, create a new function called `drawOverallScore()`. This function should set your `graphType`'s description and clear the canvas.
9. You guessed it! Wire up the overall score button to call `drawOverallScore()`.
10. Run and test, Make sure it works also. You should have three buttons that do something when clicked. When you do, you can move on.

Extracting repeated code

Note that in all three functions, you're clearing the canvas. It's never a good idea to code the same thing multiple times, so let's pull that duplication into its own function.

11. Create a new function called `clearCanvas()`. It should clear the canvas (duh). Call `clearCanvas()` from each of your three draw functions.
12. Run and test. Make sure everything is still working as expected.

Calling a function from two or more events

One more thing. Notice that when your page loads, you are drawing something. (Eventually this will be the full-blown graph). We want to be able to have that same graph drawn when the user hits the *trends* button.

13. Find where you're creating the graph. Put all of that functionality into your `drawTrends` function.

Now when you run the page, it will not draw the graph on page load (that's bad) but at least it will draw the trends graph when the user clicks the button (that's good). We want it to do both. Let's fix that.

14. Wire up the `documentDOMContentLoaded` event to call `drawTrends()`.
15. Run and test. Is the trend graph drawn on page load as well as on the button click?

If so, you've successfully re-used that same code in two events. Awesome! Go get a soda. You've earned it!