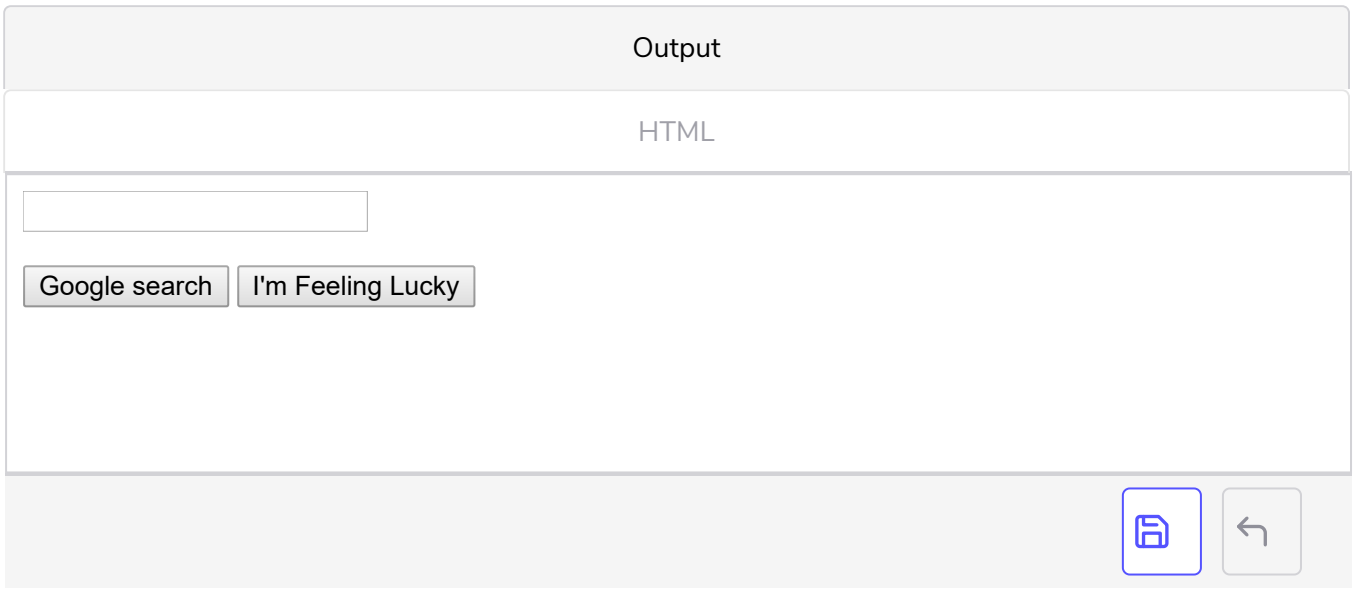
Displaying Autocomplete Suggestions

Let's connect the data from the backend and make the suggestions visible.

WE'LL COVER THE FOLLOWING

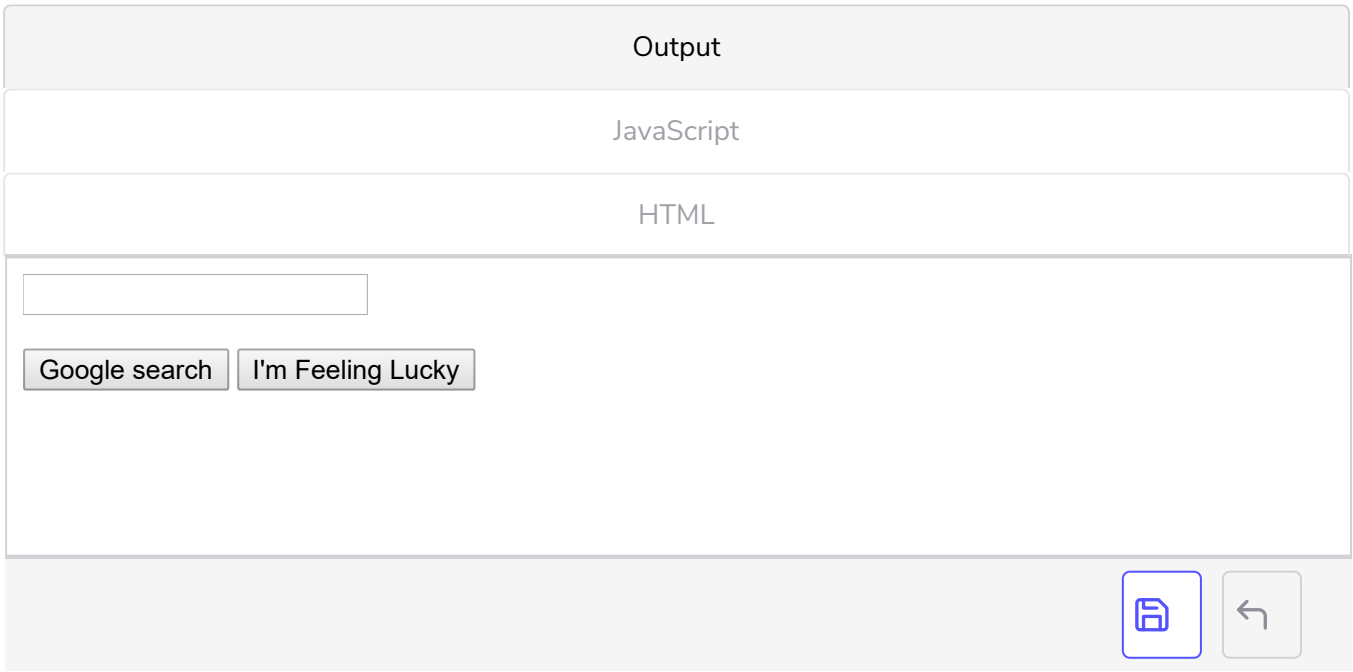
Input event listener

As you'll recall, we're working with just a basic input bar and a `ul` inside a `div` to show the list of results.



Since results show up every time the input changes, we'll add an event listener on the input element. Whenever it changes, whether that's a new character, a backspace or a copy-paste, this event listener should read the new value and get new results for us.

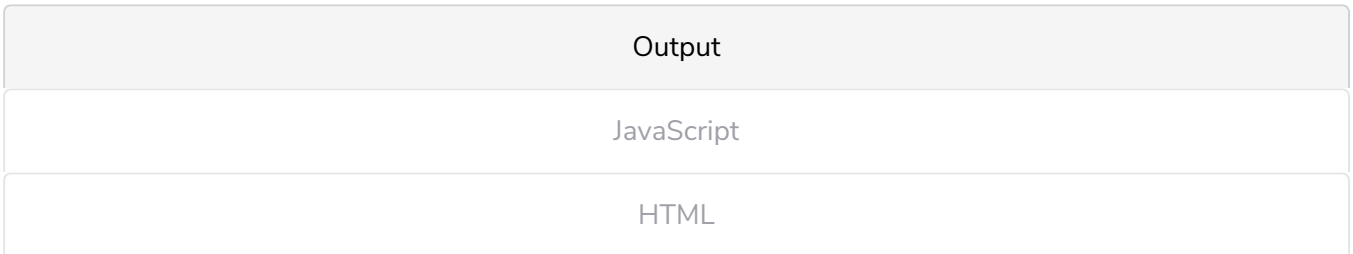
Let's add an event listener to the input and test that it works by just having it echo the results. Usually, I'd have the output be the console (part of the browser devtools mentioned earlier), but for demonstration purposes, writing HTML works too.

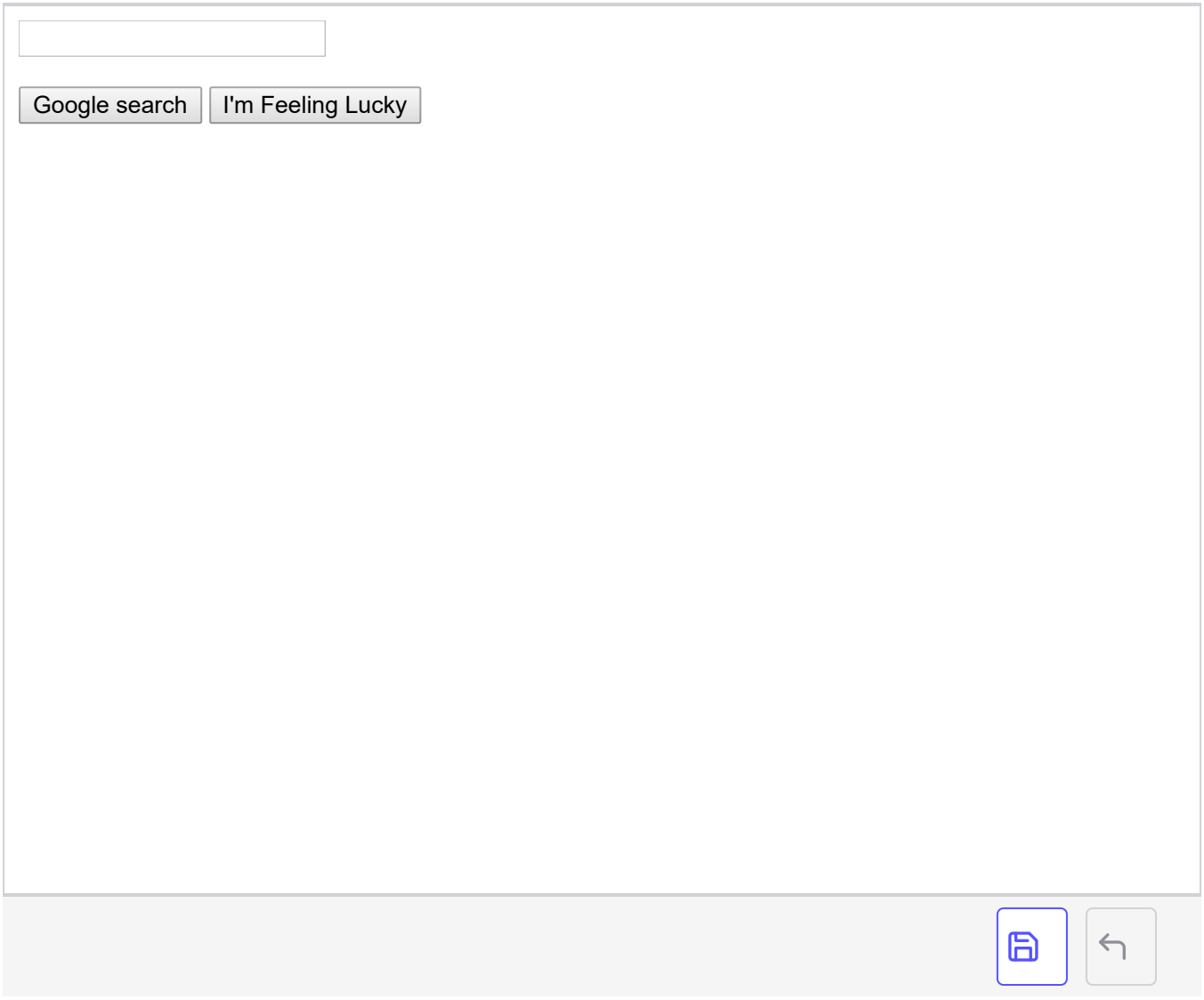


We're just doing basic DOM targeting and editing. I added the `<br>` as formatting to see more clearly that it's getting the right values. `<br>` is like HTML's newline.

Next, we'll bring in our server and make an API call on each input.

Getting server responses





I've copied and pasted the server I wrote from the last section, with the implementation of `getRandomString` and `getRandomInteger`. The code should hopefully be self-explanatory if you take some time to think about it. It's not important if you don't get it- these things you can always just Google. But, for completeness:

Random Integer

`Math.random()` gives random decimal from 0 to 1. It is a standard trick to do `* (max-min) + min` since that will just map it to our range. For example, if we have x = 0.5, max = 10, and min = 5, then this formula gives 7.5.

Random String

I manually defined a list of choices, create an empty array, and then just push random selections until the length is reached.

From here on out, I'll hide the server code since it won't be changing, and I want you to focus on the client code. You can always revisit the above code snippet as

you to focus on the client code. You can always revisit the above code snippet as needed.

But when we start typing, something unexpected happens! The output does not look like what we want. It's no doubt a minor issue that you'll run into again and again.

What do you think happened?

Q Why did we get this output?

COMPLETED 0%

1 of 1

Extracting text from the response objects

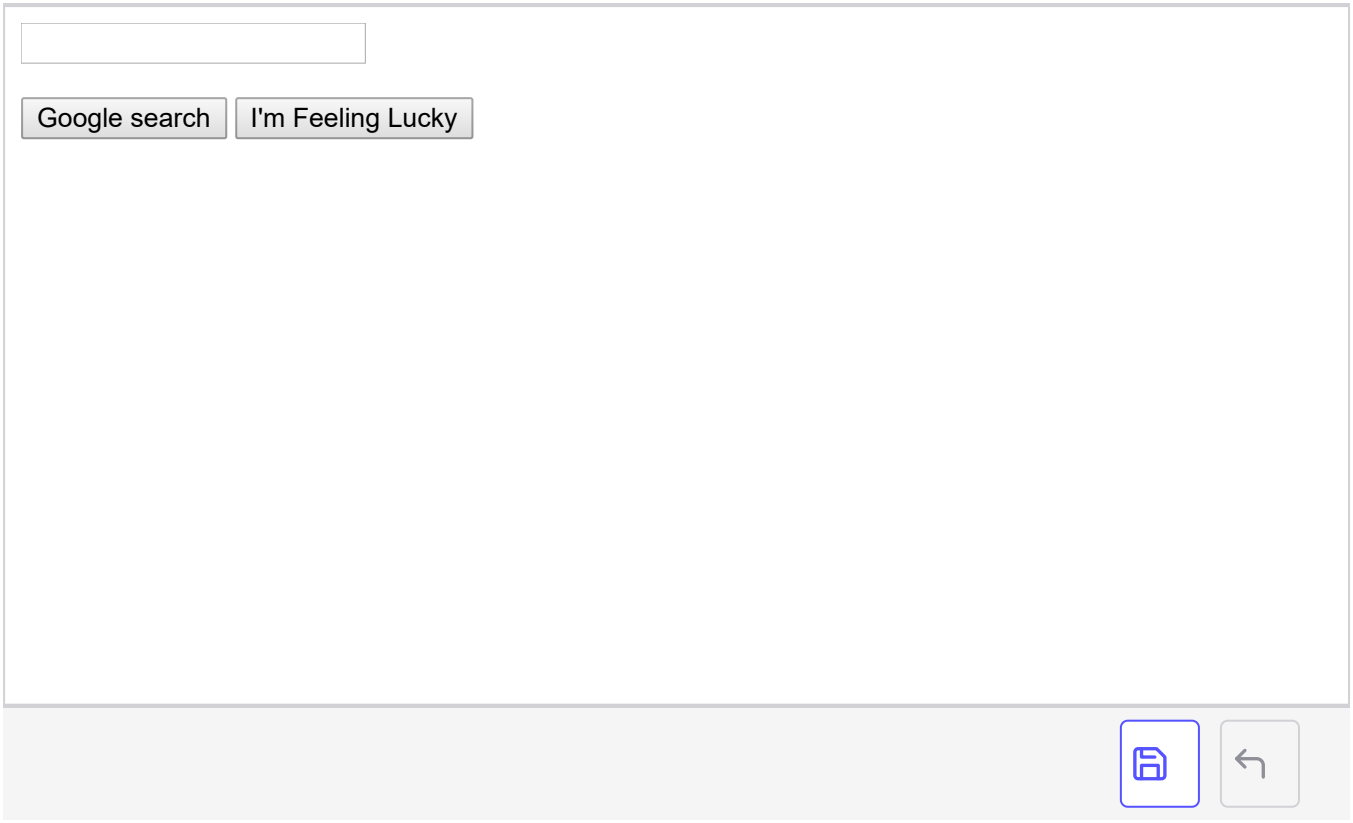
If you look back to the server code, you'll notice that it's returning a list of objects, because we need data about the suggestion as well as any auxiliary information. Since we're printing the response directly, we expected only to get the string representation of a list of objects — nothing mysterious here.

To correct this, we need to specifically target the suggestion inside the object to print.

Output

JavaScript

HTML



After typing for a little, it looks like this works!

Going down the checklist:

- Whenever auxiliary data shows up (non-empty after the `-`), there are multiple results of the same suggestion.
- There's a small set of results that are unrelated to our input, which simulates autocorrect.
- Some results are exactly the input.
- We have regular suggestions that have the input prefixed

We've gotten the suggestions to show up. In the next lesson, let's continue tweaking this and making the suggestions look somewhat like the way we want it to. This isn't all just static CSS though like the dropdown menu – we have to be smart about bolding suggestions and doing things like not generating results when there isn't input.