

6.5.6 Call the API Function When Buttons Are Clicked

We're almost finished. All that remains is making the call to the API every time the buttons are clicked and merging the project to the `master` branch on GitHub.

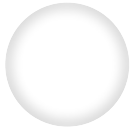
To do this, at the top of `homepage.js`, create a new variable called `languageButtonsEl`. Give it a value of the `<div>` with an `id` value of `language-buttons`.

HIDE HINT

Use the `querySelector()` method, and don't forget to include the `#` to indicate we're selecting an element by its `id` attribute!

Now we can add a click event listener to the `<div>` element that will call a new `buttonClickHandler()` function. Add the following at the bottom of the `homepage.js` file:

```
languageButtonsEl.addEventListener("click", buttonClickHandler);
```



REWIND

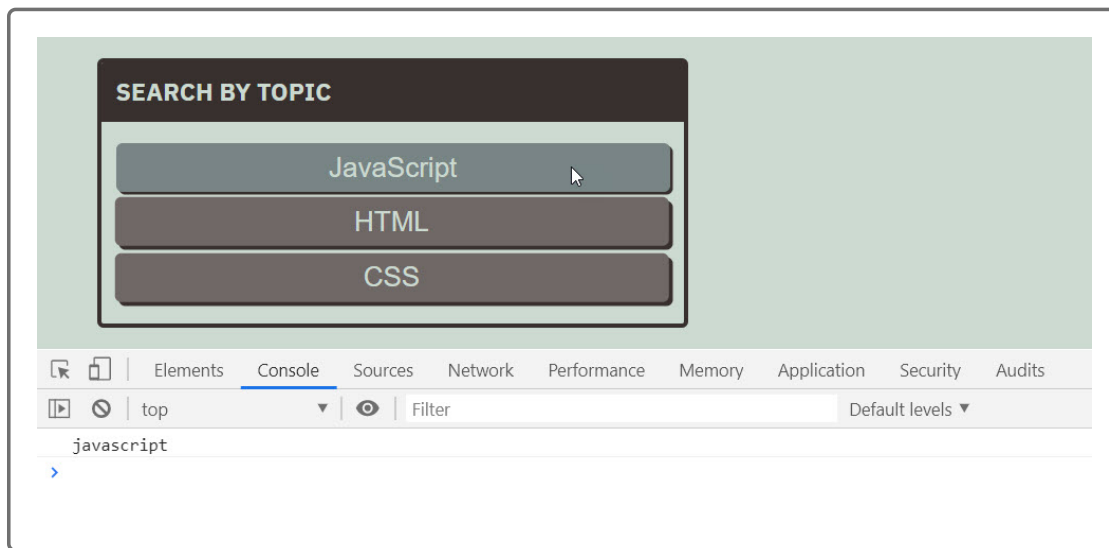
Why aren't we creating click listeners for each button? Think back to the concept of event delegation. Imagine that we had 15 language buttons instead of 3. That may add a lot of extra, repeated code. To keep the code DRY, we can delegate click handling on these elements to their parent elements.

Now let's create the `buttonClickHandler()` function somewhere above the `addEventListener()` expression to ensure things exist in the right order. This function should use event delegation to handle all clicks on buttons.

We'll start by creating a variable called `buttonClickHandler`. Give it a value of a function that accepts `event` as a parameter. Inside that function, create a variable called `language`, and give it a value of `event.target.getAttribute("data-language")`. Then use `console.log()` to log the value of `language` for testing purposes.

Remember, the browser's `event` object will have a `target` property that tells us exactly which HTML element was interacted with to create the event. Once we know which element we interacted with, we can use the `getAttribute()` method to read the `data-language` attribute's value assigned to the element.

The following image demonstrates the behavior you should see:



Now let's call the `getFeaturedRepos()` function and pass the value we just retrieved from `data-language` as the argument. Add the following `if` statement block to the `buttonClickHandler()` function right below the `language` variable declaration, as follows:

```
if (language) {  
  getFeaturedRepos(language);  
  
  // clear old content  
  repoContainerEl.textContent = "";  
}
```

Make sure to include `repoContainerEl.textContent = ""` to clear out any remaining text from the repo container. Even though this line comes after `getFeaturedRepos()`, it will always execute first, because `getFeaturedRepos()` is asynchronous and will take longer to get a response from GitHub's API.

PAUSE

The delegated event will still trigger on other elements that don't have a `data-*` attribute.

[Hide Answer](#)

Now that the code is complete, take a moment to go through the app and test the various functions. Try alternating between button clicks for HTML, CSS, and JavaScript. Try filling out a username and ensure that the form still works.

© 2020 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.