Custom Attributes

* Follow the How_to_Run_HTML_JS_FILE Folder inside this folder HtmlJSFileRun.pdf is there Open the pdf file and you will see the instructions. How to run a project.

```
<element data-*="value"> -->
```

Two parts of the custom attributes:

Attribute Name – The attribute name must be at least one character long after the prefix “data-“.

It should not contain any upper-case letters.

Attribute Value – The value to be stored can be any string.

Example 1: In this example, we will read the values of these attributes with JavaScript is quite simple.

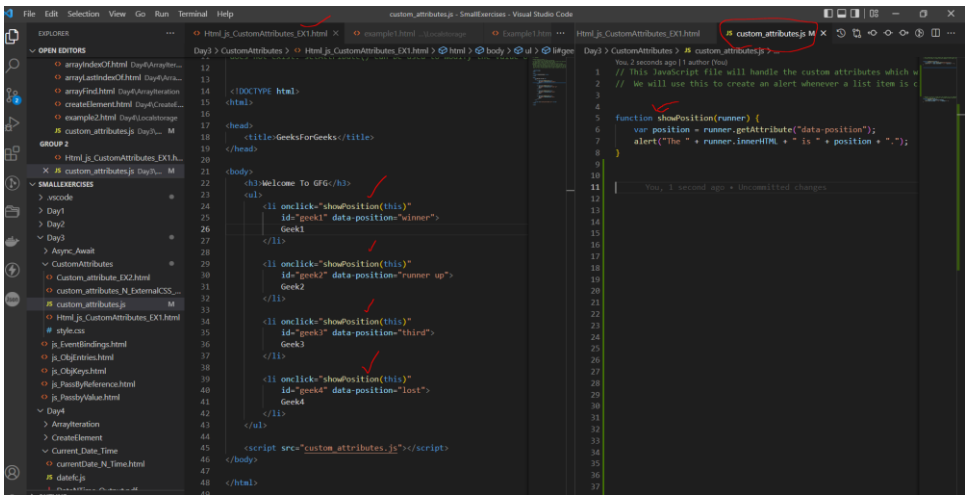
In fact there are more than one way to do so. One simpler way is using getAttribute() and setAttribute(). getAttribute()

can be used to get the stored data from the attribute. It will either return a null or an empty string if the asked attribute

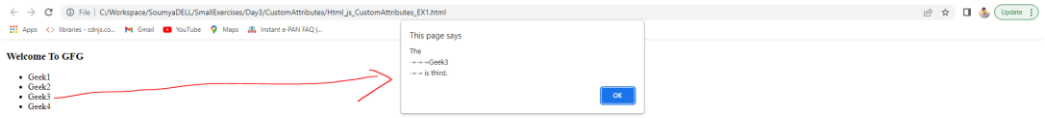
does not exist. setAttribute() can be used to modify the value of any existing attribute or to add a new attribute.

Example-1:

External js added. In this example when you click on any element like “Geek3” showPosition() method will call check the output pic.



Output:



Example-2:

In this example, we will see another way of accessing data attributes is by using dataset property.

This property returns a DOMStringMap object with one entry for each custom data attribute.

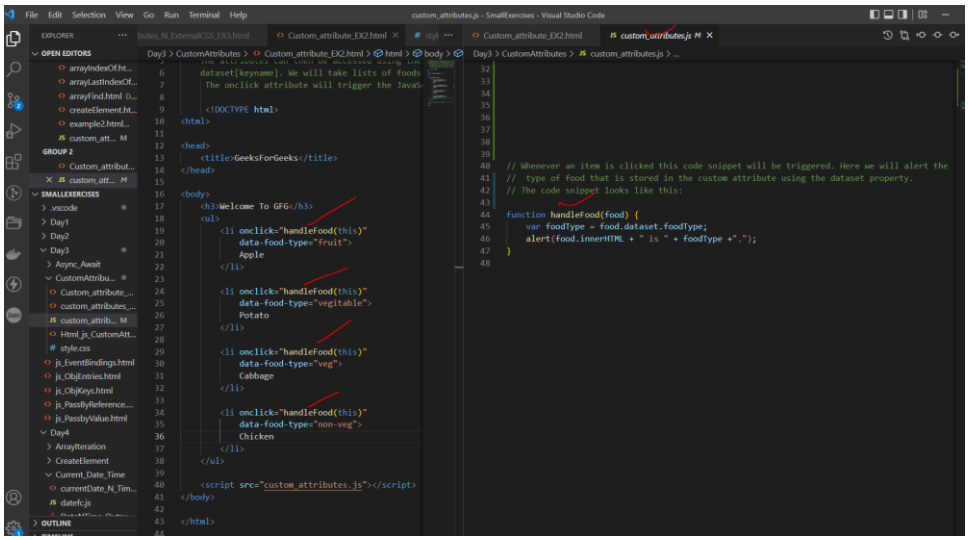
A DOMStringMap key is a transformed form of custom data attribute. The “data-” prefix is removed from the attribute name.

Any hyphen in the name is also removed. In this way, we get a camelCase name.

The attributes can then be accessed using the camelCase name stored in the object as a key like element.dataset.keyname or element.

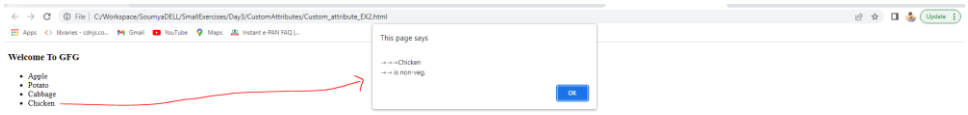
dataset[keyname]. We will take lists of foods. The custom attributes will contain food-type.

The onclick attribute will trigger the JavaScript when an item is clicked.



Output:

When you click any element then handleFood() method will call see the below image



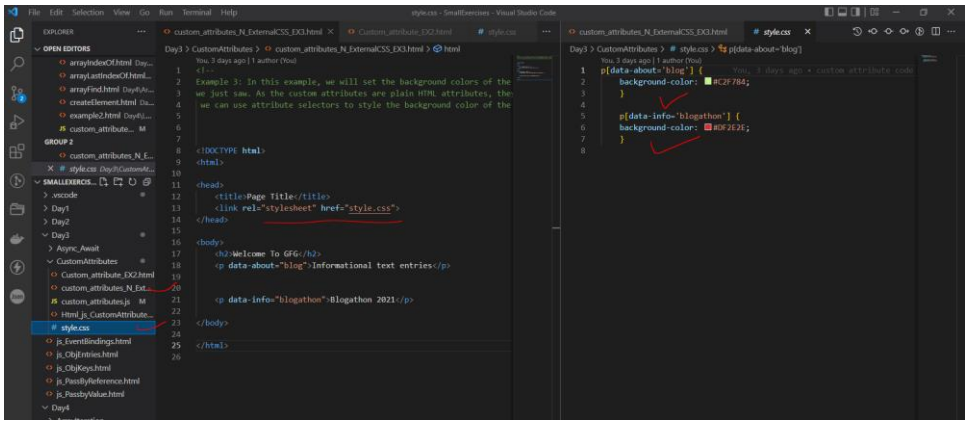
Example3:

In this example, we will set the background colors of the elements using the CSS access that

we just saw. As the custom attributes are plain HTML attributes, they can be accessed from CSS. For example,

we can use attribute selectors to style the background color of the element.

External CSS added:



Output: