View Email

Things to do

1. Add data attribute to row div containing (i) email\_id (ii) boolean value for whether email has been read or not [Done]
2. Add new div in inbox.html called email-view [Note that is it NOT emails-view] [Done]
3. AddEventListener to div with row with click event. Event listener should do the following (i) create API call with event\_id (ii) If returns an error, then display an alert that shows an error (iii) If response is successful, then take the information from the API call and display the following (a) subject (b) sender (c) content. Each should be wrapped in a special div (iv) should also change data attribute to read (v) hide compose view and emails view and only display email-view [Done]
4. AddEventListner to div with row with event mouseover -> Basically the cursor should change when we hover over the div. [Done]
5. Add CSS for the divs created
6. With the new email-view need to make sure that it is hidden in compose, inbox, [Done]

What are HTML Data Attributes?

To store variables within a HTML attribute, use the data tag. For example:

<article

id="electric-cars"

data-columns="3"

data-index-number="12314"

data-parent="cars">

...

</article>

Data attribute meant to store extra information on standard HTML elements. Syntax is data-\* attributes. Any attribute on any element whose name starts with data- is a data attribute.

How to access in JavaScript?

2 ways

* getAttribute() with full HTML name to read the data. I.E. article.getAttribute(“data-columns”) would give 3.
* Use DOMStringMap to read out the dataset property. Dataset property is a read-only property of all HTML elements to provide read/write access to custom data attributes.

The 2nd method is the recommended method and example is shown below:

const article = document.querySelector('#electric-cars');

// The following would also work:

// const article = document.getElementById("electric-cars")

article.dataset.columns // "3"

article.dataset.indexNumber // "12314"

article.dataset.parent // "cars"

To get a data attribute through the dataset object, get the property by the part of the attribute name after data- (note that dashes are converted to camelCase).

**To read data attribute in HTML element**

elementName.dataset.[nameOfDataAttributeInCamelCase]

If you use elementName.dataset -> This would return a DOMStringMap which is basically just an JSON object that contains all the custom data attribute of an element.

Default value for data attributes without value is an empty string – “”.

**To write data attribute in HTML elements**

elementName.dataset.test = value

**Delete Data Attributes**

Delete div.dataset.dataAttributeName

Need to use the delete keyword. You can’t just set it to null or undefined as the data attribute will still exists just with a new value of null or undefined.

**Changing CSS with Javascript**

document.getElementById("lefteye")

document.getElementById("lefteye").style.backgroundColor = "purple";

Changing CSS with JS is similar to changing HTML with JS. Get the element with whatever select you want such as getElementById OR querySelector. After that you access any CSS property with the .style.cssName.

Note: When CSS property is just 1 word, use the same CSS name to change the style in JS. Examples are border or margin. However, when CSS property is 2 words, in CSS it is separated with a – such as margin-top. In JS, this property would be changed to camelCase. Hence, background-color becomes backgroundColor.