**Step 1** move the array of objects (tweets) into data.js

Do the imports/exports and add type=”module” to the html in the js script section

**Step 2**. In the useless chatbox example we added a textarea with id….we took the id called chatInput and called the .value….chatInput.value…this way we can get any text being typed….we also cleared out the text to an empty string after the button is clicked…..chatInput.value = ‘’

Also inside the css we set the resize value to none in the textarea property…this way users cannot physically resize the textarea box.

**Step 3**. Lets take the data from the data.js file and pass it via a for of loop and into an html string. We’ll also call the getFeedHtml() and do a console.log(feedHtml) to make sure that we are receiving the data in a string…NOTE that the feedHtml has a **+=** ….this is b/c we have multiple tweets to pass over and over

We begin by setting a let variable of feedHtml to an empty string, then we created our for loop, next we took our feedHtml variable and set template strings `` and inside we passed the html to be rendred with the variables replaced and the array set to .length for the replies variable as we want the length of the replies…(some may have 0 tweets other may have hundreds….so we need .length)

function getFeedHtml() {

  //Job of fuction to iterate through data.js and create HTML from each tweet with the boilerplate code which we will replace using template literals

  let feedHtml = "";

  for (let feed of tweetsData) {

\*/

    feedHtml += `

<div class="tweet">

    <div class="tweet-inner">

        <img src="${feed.profilePic}" class="profile-pic">

        <div>

            <p class="handle">${feed.handle}</p>

            <p class="tweet-text">${feed.tweetText}</p>

            <div class="tweet-details">

                <span class="tweet-detail">

                    ${feed.replies.length}

                </span>

                <span class="tweet-detail">

                    ${feed.likes}

                </span>

                <span class="tweet-detail">

                    ${feed.retweets}

                </span>

            </div>

        </div>

    </div>

</div>

`;

  } //END for of function

//   console.log(feedHtml); we did a test with console now we will just return feedHtml

return feedHtml

} //end getFeedHtml function

//STEP 1 pass tweetsData inside getFeedHtml function

getFeedHtml();

DIFFERENCE WHEN USING += AND = FOR THE FOR OF LOOP

Remember in our cat emotions app we used the for of loop when rendering a single cat image inside the modal and we used the equal sign alone b/c we are only passing in one pic

function renderCat() {

  const catObject = getSingleCatObject();

  memeModalInner.innerHTML = `<img

        class="cat-img"

        src="./images/${catObject.image}"

        alt=${catObject.image}

        >`;

  memeModal.style.display = "flex";