Cat Meme Emotion Picker:

Use of the for of loops and how to nest for loops inside of each other.

//old way using nested regular for loops…………… // function getEmotionsArray(cats){

//     const emotionsArray = []

//     for (let i = 0; i < cats.length; i++){

//         for (let j=0; j < cats[i].emotionTags.length; j++){

//                 emotionsArray.push(cats[i].emotionTags[j])

//         }

//     }

Challenge:

1. Set up a "for of" in getEmotionsArray to iterate

   over the data.

2. For now, just log out each cat object individually.

\*/

Const catsData = [

{

emotionTags: ["hungry"],

isGif: true,

image: "hungry.gif",

alt: "A cat looking hungry",

},

{

emotionTags: ["hungry"],

isGif: true,

image: "hungry2.gif",

alt: "A cat looking hungry",

},

] //end of catsData array of objects

The general syntax of the for loop:

for (let cat of cats){

console.log(cat)

}

Ex. Let’s say the cats is an actual array of objects from above and we want to access the emotionTags property (which happens to be an array)…we would need to do cat.emotionTags to access the object properties.

  for (let cat of cats) {

    for (let emotion of cat.emotionTags) {

      console.log(emotion);

    }

  }

}

The first for loop will iterate in the first cat object from the cats array

2. then it will access inside the object the property called emotionTags by doing cat.emotionTags..this is saying lets iterate the emotionTags array from each cat

So part 1 we have:

const catEmotionsArray = [];

function getEmotionsArray(cats) {

  for (let cat of cats) {

    //console.log(cat.alt);

    for (let emotion of cat.emotionTags) {

      //push each cat emotion to the catEmotions array

      catEmotionsArray.push(emotion);

    }

  }

  return catEmotionsArray;

} //end getEmotionsArray function

This is accessing the catsData array….which is a iterating out the catEmotios property which is an array

and it returns an array called catEmotionsArray whenever it’s called.

Part2: we have another function called renderEmotionsRadios

This is going to take in a single parameter called cats….

And inside the function we are setting a variable to hold the contends of the getEmotionsArray which will take in the cat parameter…..the cat parameter is actually the catsData object being passed in to the function renderEmotionsRadios which will then be passed inside the getEmotionsArray

function renderEmotionsRadios(cats) {

  const emotions = getEmotionsArray(cats);

  console.log(emotions);

}

renderEmotionsRadios(catsData);