Rev. 08Aug2019

Exercise 22

MORE ON MAKING OUR OWN FUNCTIONS

Functions play a huge role in JavaScript, so let's spend more time working with them. We're going to create several functions and even have some functions call other functions. But we're going to do this as simply as possible.

First function: Adding two numbers.

I'll do the first one so that you can use it as a pattern reference.

```
let adder = (number1, number2) => {
  return number1 + number2
}
```

Second function: Concatenating two words

- Here's a bit of code to get you started:

```
let concatenator = (???) => {
  return ??? // concatenated text
}
```

What arguments do you need?

What do you need to do to concatenate (i.e. join) the arguments? Don't forget to place a space between the two words.

Third function: Placing text on a HTML element via dynamic id

The goal: your function accepts an `id` that it uses to locate a particular element and write to it.

```
let welcome = (???) => {
  let element = document.getElementById(???)
  let message = "Hello, my id is " + ???
  element.innerHTML = ???
}
???
```

I've provided HTML elements into which you can place the results

Remember that these functions won't do anything until they're called

You need to get three things from the person calling your function: first name, last name, and id to place the greeting

Fourth function: Concatenate first name and last name and place in given id dynamically

```
let greetByName = (???) => {
// find the element using getElementById
// create the greeting using the first name and last name
// set the element's innerHTML to the greeting
}
// call the function
```

The `//` followed by text is a JavaScript comment, which is ignored by the browser. Here, the comments provide instructions to you for writing the function body.

Fifth function: Add a class to a given id

The goal: accept a `class` name and an `id` and to dynamically add that `class` to the given element.

Remember to use classList.add

```
let classAdder = (???) => {
  let element = ???
  // add class to element
}
// call the function
```

Sixth function: Accept first name, last name, and class name and populate <p id="one"> with results

The goal: accept a first name, last name, and class name. Find the element with an `id` of `delegating-function` and place the concatenated first and last names in that element. Then add the given `class` name to your selected element.

You've got two functions already written you can use: `concatenator` and `classAdder`.

```
let delegatingFunction = () => {
  let concatenatedName = ??? // you have a function you can call to do this
  // get the element by id
  // place the concatenatedName in the element
  // call the classAdder, passing it the class name that was one of the arguments
  // for this function
}

// call the function
```

Yes, this one is a little harder, but don't be intimidated. Give it a good try. If you get stuck, check out my solution in `answer.js`.