Lesson | Tuesday

Introduction to Programming (/introduction-to-programming) / Arrays and Looping (/introduction-to-programming/arrays-and-looping) / Separation of Concerns in Text Analyzer: boldPassage() UI Function

Text | Cheat sheet

Terminology

Separation of concerns: A key programming design pattern that dictates that each function should only be responsible for doing one thing, and not know about anything else in the application. In this context, a **concern** is a responsibility. So when we apply separation of concerns to our code, we're separating the functionality of our webpage into multiple different functions, each of which has a single responsibility. For instance, one function might be 'concerned' about one thing (adding two numbers together) while another function might be 'concerned' with returning those numbers to the user.

Using an `index` with `Array.prototype.forEach()`

We can pass in an `index` as the second parameter of the callback we pass into `Array.prototype.forEach()`. This allows us to get the index of the current iteration of the loop. The index always starts at 0. Here's an example:

```
const string = "I like cats!";
string.split(" ").forEach(function(element, index) {
  console.log(element, index);
});
```

Tests and Code for `boldPassage()`

Remember that we'll be writing tests for business logic only. The following tests for the user interface function `boldPassage()` is for TDD demonstration purposes only.

```
Describe: boldPassage()

Test: "It should return null if no word or text is entere
d."
Code:
const text = "";
const word = "";
boldPassage(word, text);
Expected Output: null

Test: "It should return a non-matching word in a p tag."
Code:
const word = "hello";
const text = "yo";
boldPassage(word, text);
Expected Output: <p>yo</p>

Test: "It should return a matching word in a strong tag."
Code:
const word = "hello";
const text = "hello";
boldPassage(word, text);
Expected Output: <p><strong>hello</strong></p>

Test: "It should wrap words that match in strong tags but n
ot words that don't."
Code:
const word = "hello";
const text = "hello there";
boldPassage(word, text);
Expected Output: <p><strong>hello</strong> there</p>
```

```
function boldPassage(word, text) {
  if ((text.trim().length === 0) || (word.trim().length ===
0)) {
    return null;
  }
  const p = document.createElement("p");
  let textArray = text.split(" ");
  textArray.forEach(function(element, index) {
    if (word === element) {
      const bold = document.createElement("strong");
      bold.append(element);
      p.append(bold);
    } else {
      p.append(element);
    }
    if (index !== (textArray.length - 1)) {
      p.append(" ");
    }
  });
  return p;
}
```

Previous (/introduction-to-programming/arrays-and-looping/separation-of-logic-fixing-a-bug-in-text-analyzer)
Next (/introduction-to-programming/arrays-and-looping/drying-code-and-completing-the-text-analyzer-ui)

Lesson 30 of 50
Last updated February 28, 2023

disable dark mode

(http://www.epicodus.com)