Megan Walker

Bellevue University

**WEB 330 - Enterprise JavaScript II**

Professor Krasso

January 23, 2023

## Discussion 4.1 – Lambda Expressions

### What are Lambda expressions?

Lambda expressions, also called arrow functions, are a way to write functions in JavaScript using shorter and simpler code. They were first introduced in ECMAScript 6 and are used to make functions more concise and easy to read and understand.

### How are Lambda expressions used in JavaScript?

Lambda expressions are mostly used to create callbacks, pass functions as arguments, and create closures. A regular function to add two numbers together would look like this:

```
function add(a, b) {
  return a + b;
}
```

Using Lambda expression, it would look like this instead:

```
let add = (a, b) => a + b;
```

**Why would you use Lambda expressions over a traditional functional approach?**

Lambda expressions are shorter and simpler than regular functions. This makes code easier to read and understand. They do not require the "function" keyword, "return" statement, or curly braces in certain cases. This makes code more readable, especially when working with higher-order functions, callbacks, and closures.

**What are the potential pitfalls of using Lambda expressions?**

There are some downsides to using lambda expressions. They can't be used as constructors or with the "new" keyword, which makes them less flexible than regular functions. Also, they don't have their own "this" or "arguments" bindings, which can cause unexpected behavior when working with objects and methods.

References

*Arrow function expressions - JavaScript | MDN*. (2023, January 21). Mozilla.org. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Arrow_functions

*JavaScript Arrow Function*. (2015). W3schools.com. https://www.w3schools.com/js/js_arrow_function.asp