In this essay I want to discuss JavaScript Lambda expressions (a.k.a. arrow functions). I want to discuss how Lambda expressions are used in JavaScript and will state reasons as to why one would use Lambda expressions over a traditional function approach. As well, I will discuss the potential pitfalls of using lambda expressions in JavaScript. So, what are lambda expressions?

Lambda expressions are “a short block of code which takes in parameters and returns a value. Lambda expressions are like methods, but they do not need a name and they can be implemented right in the body of a method.” (*Lambda Expressions*, 2022). Lambda expressions are used in JavaScript to form compact alternatives to traditional function expressions. They are a kind of anonymous function that reduces syntax and can be clutter free. Lets consider the example below of an anonymous function that we can then rewrite as a lambda expression:

Graphical user interface

Description automatically generated with medium confidence

**As a result of using the lambda expression, the 8-character “function” is now a 2-character “=>”. (2021).**

**The following are reasons to use lambda functions over traditional functions:**

* Lambda expressions save time and space.
* Lambda expressions are structured in a way that they are extremely readable.
* Since they are readable, they are very concise.
* Since they are anonymous, they can be reused.

The potential pitfalls of using Lambda expressions are:

* The expressions themselves can be unfamiliar to other developers trying to read your code.
* They lack names and documentation, meaning the only to know that they do is to read the code.
* Lambda expressions do not have their own bindings to this, arguments, or super, and should not be used as methods.
* They can not be used as constructors.

(*Arrow Function Expressions - JavaScript | MDN*, 2022)

References:

*Java Lambda Expressions*. (2022). W3schools.Com. Retrieved 20222–08-29, from <https://www.w3schools.com/java/java_lambda.asp>

み. (2021, December 28). *Introduction to lambda expression with JavaScript [Easy-to-understand explanation for beginners]*. Midolog.Net. <https://midolog.net/en/javascript-lambda-expression-tutorial/>

C., & C. (2019, December 5). *Benefits of Lambda Expressions*. TechStack. Retrieved August 29, 2022, from <https://techstack.tech.blog/2019/12/05/benefits-of-lambda-expressions/>

*Arrow function expressions - JavaScript | MDN*. (2022, August 7). Mdn Web Docs. Retrieved August 29, 2022, from <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Arrow_functions>