Discussion 4.1 Lambda Expressions

Shane Hingtgen

1/23/2023

Bellevue University

A Lambda expression is a way for writing a short function. Lambda expressions are not limited to JavaScript. It’s just a description of what they “do for functions what object-oriented programming does for objects. ” Although it may be different among other programming languages such as C++ or Python. . (*Lambda Expressions: A Guide*, n.d.) Also called anonymous functions or closures. (*Lambda Expressions: A Guide*, n.d.) Lambda Expression recognize that statements are different than expressions. (*Lambda Expressions: A Guide*, n.d.)

A way of writing a Lambda Expression is using an Arrow Function. For example, x => x + 1 is a shorthand way of writing a function that is also anonymous. Another Example:

Let z = (x => x +1);

There is no function name. You could use this in functions that are repeatedly applied throughout your code. (Ball, 2018) But according to zendev.com you should be careful when using lambda expression because they may cause problems and not to use them to replace all of your functions. (Ball, 2018)

According to zendev.com you would use a lambda expression in arrays that you are transforming with .map(). An example:

const words = [‘HELLO’, ‘World’];

const lowerCaseWord = words.map(word => word.toLowerCase()); (Ball, 2018) There are some problems with using Lambda Expressiosn. They are not meant to replace functions because they may process differently than normal functions. (Ball, 2018) The website explains that there was a trend for awhile to use “Class Properties and arrow functions as a away to create “auto-binding” methods”. (Ball, 2018) And it is recommended to just use a normal function in many cases.

References:

*Lambda Expressions: A Guide*. (n.d.). Joshdata. Retrieved January 23, 2023, from https://joshdata.me/lambda-expressions.html

Ball, K. (2018, October 1). *JavaScript Arrow Functions: How, Why, When (and WHEN NOT) to Use Them*. ZenDev, LLC. Retrieved January 23, 2023, from https://zendev.com/2018/10/01/javascript-arrow-functions-how-why-when.html