Prompts

1. What are the differences between var, let, and const?

2. What are the differences between callbacks and promises?

3. What are some features that are new with ES6?

4. How does a promise work?

Instructions

As developers, research is a constant part of our job. A common saying is that 90% of software development is Googling, and while that is an exaggeration, Google is a highly used tool in the role.

This Research assignment is meant to go beyond the course curriculum and increase your understanding of relevant topics while exposing you to online resources you'll frequently use on the job.

Please write a paragraph for **two (2)** of the above prompts and include URLs from where you found the information to cite your sources. Do not copy and paste text from the internet or any other source; use the information you find in your research, summarize, in your own words, the concepts. Plagiarism will result in a zero for the assignment as well as disciplinary actions.

Prompts: 3 and 4

Some of the new features with ES6 include the let and const keyword, which allows you to declare a variable with block scope, except for the latter's value not being able to be redeclared or reassigned. Arrow functions, allows for a more concise code that includes shorter lines of code with the same functionality; except for the arrow function must be defined before it can be used. Map and promises are objects in JavaScript ES6, with map, uses a method that will result from invoking a function on every element in the original array and returning a new array with those new elements. A promise is great for handling asynchronous operations and is used to discover if something happening in the future is successful or a failure. [1,2] A promise works sort of like a placeholder for a value not known yet and is typically provided by an external source. It is an object that results in a future value with three possible results: pending (default state of a defined promise), fulfilled (state of a successful promise), and rejected (state of a failed promise). [3]

1 https://www.w3schools.com/js/js_es6.asp

2 https://www.w3schools.com/js/js_object_maps.asp

3 https://www.freecodecamp.org/news/guide-to-javascript-promises/#:~:text=In%20JavaScript%2C%20a%20Promise%20is,of%20promises%20in%20real%20life.