

Note to grader: I made the decision to research all topics because if they were important enough to put in this assignment they are important enough for me to research and understand.

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

Prior to the advent of ES6, the use of VAR was all that JavaScript offered. There were no other options; however, there were issues with VAR so in the new version ES6 two other options were included; let and const. Var provided variables both locally and globally. Var works both inside and outside of a function block. Var has a particular issue that warranted the advent of let and const. The issue is most pronounced when code is very long and there is a chance you don't know that you are about to reuse some code. Var allows you to change any variable even if the intention was to make a variable constant throughout a program; therefore, const and let were created to solve this issue. Const created a variable that could in theory not be changed, at least not intentionally. To change this variable, you must go back to the original line and intentionally change the value. Let is now preferred over var. It is blocked-scoped, meaning that it is bound by a couple of curly braces {}. Var can be updated but not re-declared, practically eliminating an accidental change that could affect the file globally. One major difference between let and var is that var can be declared and initialized as undefined. If the same is attempted for let, the developer will receive a reference error. In conclusion, var and let were created to mitigate and minimize the possibility of accidental modification of variables leading to errors in code.

2. What are the differences between callbacks and promises?

Callbacks are functions that are passed inside a new function and then used by the function to perform a task; while promises, designed to solve callback issues, are JavaScript objects that link producing code and consuming code. Producing code is code that can take some time while consuming code is code that must wait for a result. Callbacks essentially allow JavaScript to be asynchronous by allowing the script to call back to an earlier place within the code. The problem with callbacks is that when there are too many, they become too difficult to read and maintain. It results in confusing code; therefore, promises were created in JavaScript. Promises are set up to have two results: resolve and reject. Resolve means the code was successful and reject means it was not. Whenever the request is successful, the promise is resolved, and the code goes on to do whatever we tell the code to do. If there is an error, then the promise is rejected. Essentially promises provide a cleaner more readable way to accomplish the same task.

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

Some of the new features that are used with ES6 are as follows:

- a. two additional variables are included: const and let created to resolve issues with var;
- b. promises were created to resolve some issues with callbacks;
- c. Arrow functions were created to simplify the use of functions;
- d. The ... operator is used to expand the iterable into more arguments for function calls; and
- e. The for/of loop was created to loop over an Array or a String.

4. How does a promise work?

A JavaScript promise is an object that will create one of the following four states:

1. Fulfilled: This is an action where the promise was successful.
2. Rejected: Another event type where the promise has failed.
3. Pending: An event type where the promise has not been completed and resulted in fulfilled or rejected.
4. Settled: An event type about where the promise is fulfilled or rejected.

The promise can be a present or future value. Promises, designed to solve callback issues, are JavaScript objects that link producing code and consuming code. Producing code is code that can take some time while consuming code is code that must wait for a result. The result to be produced is one of the four states above. The code will only move on once the promise has been settled. This allows the developer to callback to previous code without creating the infamous callback hell due to too many confusing callbacks.

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.

This assignment is graded based on participation.

Reminder: Create a .pdf file with your research findings (.pdf is the only accepted file type).

- To save as a .pdf, go to Save As or Export, then choose the file type .pdf. This will save a new version of your document as a .pdf.

- If you have a Mac and use Pages, please make sure to export the final version of the document to a .pdf before submitting.