



RESEARCH ASSIGNMENT Week-04-ECMAScript_6_and_Intermediate_JavaScript

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?
5. What is your favorite thing you learned this week?

Student Answers.

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

“var” and “let” are used for slightly different reason but function in the same manner. “var” is used is global way and “let” is used in a more specific way, yet when used together to declare the value of a variable, there is no error signaled even if that value is changed. In the other end “const” is used to set a constant value to a variable. The value of variable declared by “const” cannot be changed, if the value is changed, an error will be returned or logged to the console.

2. What are the differences between callbacks and promises?

According to dev.to “Callbacks are functions passed as arguments into other functions to make sure mandatory variables are available within the callback-function's scope. Promises are placeholder objects for data that's available in the future” Promises are more than placeholder objects, the help the developer verify if the expected outcome of the code is successful or not by performing a assigned task if the expected outcome if successful or another task if the expected outcome is unsuccessful it also helps the developer write a much readable code.

Ref: <https://dev.to/tqbit/what-is-the-difference-between-callback-functions-promises-and-async-await-in-javascript-1c2k#:~:text=The%20TL%3ADR%20%2D%20version%3A,that's%20available%20in%20the%20future.>



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

At this level we have been introduced to the difference var, let and const; arrow functions and callbacks and promises. Through some web research, according to Luke Hoban GitHub repository <https://github.com/lukehoban/es6features>, the following features will also be new to students with ES6.

ES6 includes the following new features:

- [arrows](#)
- [classes](#)
- [enhanced object literals](#)
- [template strings](#)
- [destructuring](#)
- [default + rest + spread](#)
- [let + const](#)
- [iterators + for..of](#)
- [generators](#)
- [unicode](#)
- [modules](#)
- [module loaders](#)
- [map + set + weakmap + weakset](#)
- [proxies](#)
- [symbols](#)
- [subclassable built-ins](#)
- [promises](#)
- [math + number + string + array + object APIs](#)
- [binary and octal literals](#)
- [reflect api](#)
- [tail calls](#)

4. How does a promise work?

Like stated above a promise holds the place of an object with an unknown outcome and then returns the result of the outcome as successful or failure. A more technical explanation is found here: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise

*“The **Promise** object represents the eventual completion (or failure) of an asynchronous operation and its resulting value.”*



PROMINEO TECH

5. *Everything I learned and still learning this week is exciting but also very intimidating. It's too early to have a favorite at this point. Moving into the upcoming weeks will I hope hit my favorite string within me.*