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

While var, let and const are similar in that they are all used for declaring variables, there are a few major differences. Var specifically has a global scope when it is declared outside of a function, meaning it can be used anywhere in the code. Although when it is declared in the function, it can only be used inside that function. They can also be declared again and re-assigned within the same scope. Let on the other hand is blocked scoped, which means that if it is defined within a block, or a set of curly braces, whether it is a function, if/else statement, loop, etc., once that block is closed off that variable will no longer be defined. Let can be re-assigned, but not redeclared, unlike var, thus giving us less chance of accidentally re-using the same variable name, unless it is within a different scope. Const is similar to let in that it is block scoped. The difference between const and both let and var is that const cannot be re-assigned or redeclared. This is useful when you don’t want the user to change the value of a specific variable.

https://www.freecodecamp.org/news/var-let-and-const-whats-the-difference/

1. What are the differences between callbacks and promises?

Callbacks and promises work very similarly, in the sense that they both use a function as a parameter for another function. A big difference is in syntax, where callback looks similar to a function, using parameters, returns, braces, promises have a simpler syntax. Promises just take in one parameter, which is the function being called in, and the return value gets evaluated by .then or .catch. This makes a more legible code for promises where we can use multiple .then methods where callback syntax would create multiple blocks with braces making it harder to read.

EVALATE MOR EON THIS AND HOW THEY FUNCTION

1. What are some features that are new with ES6?
2. How does a promise work?