1.

Var: This is function based scoping

Let: Code Block scope

Const: code block scope for a unchangeable variable.

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

2.

Callbacks: Callbacks is a function that is passed into the argument of other function. This just makes sure the value is available prior to the primary function is generated. This makes sure there is the information for the function to execute that function as the callback is the basis for the function.

Promises: this is a placeholder object that will have a setting of pending until the state of the object changes which then makes it turn to state of resolved. At this point you are able to do a call of .then on the object which will use that object and then execute this code block if resolved.

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.

New features in ES6 include the following. The primary ones are let, const, arrow functions, map/sets and the major ability of having classes which gives us the ability to have this become an object oriented language.

- Arrow functions
- Let keyword
- Const keyword
- … Operator
- For/of
- Map Objects
- Set Objects
- Classes
- Promises
- Symbol
- Default Parameter
- Function Rest Parameter
- String.Includes()
- String.startsWith()
- String.endsWith()

- Array.from()
- Array.keys()
- Array.find()
- Array.findIndex()

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

4. This allows us to have a api call to a server asking for information whether it is a filter of data from a database or having us execute code that is housed on the server. This allows us to execute code async to this because the promise will activate a .then or .catch for (resolved, rejected) and then execute that code when it had returned from the async process that was executed.

https://www.youtube.com/watch?v=DHvZLI7Db8E