Callback: A callback is a function which is called when a task is completed, thus helps in preventing any kind of blocking. It allows other code to run in the mean time. Using this concept, node js can process large no of requests without waiting for any function to return the result. For example: In Node.js, when a function start reading file, it returns the control to execution environment immediately so that the next instruction can be executed. Once file I/O gets completed, callback function will get called to avoid blocking or wait for File I/O.

https://www.geeksforgeeks.org/node-js-callback-concept/#:~:text=js%20Callback%20Concept,-View%20Discussion&text=A%20callback%20is%20a%20function,asynchronous%20equivalent%20for%20a%20function.

**Synchronous JavaScript:** As the name suggests synchronous means to be in a sequence, i.e. every statement of the code gets executed one by one. So, basically a statement has to wait for the earlier statement to get executed.

**Asynchronous JavaScript:** Asynchronous code allows the program to be executed immediately where the synchronous code will block further execution of the remaining code until it finishes the current one.  it does not block the rest of the code from executing and after all the code its execution, it gets pushed to the call stack and then finally gets executed.

**Promises:** It is basically an advancement of call backs in Node. It is a js obj which is used to handle all the asynchronous data operations.