Q.1 What's difference between Synchronous and Asynchronous?

Synchronous: 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 : 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. This may not look like a big problem but when you see it in a bigger picture you realize that it may lead to delaying the User Interface.

Q.2 What are Web Apis ?

Web API is the enhanced form of the web application to provide services on different devices like laptop, mobile, and others.

API stands for Application Programming Interface.

A Web API is an application programming interface for the Web.

A Browser API can extend the functionality of a web browser.

A Server API can extend the functionality of a web server.

Q.3 Explain SetTimeOut and setInterval ?

SetTimeout() Method:

setTimeout() method executes a function, after waiting a specified number of milliseconds.

Syntax:

window.setTimeout(function, milliseconds);

There are two parameters accepted by setTimeout() method.

function:

The first parameter is a function to be executed

milliseconds:

Which indicates the number of milliseconds before the execution takes place.

We can even stop the execution of the setTimeout() function by using a method called as clearTimeout().

Syntax:

window.clearTimeout(value)

value:

The function whose execution is to be stopped.

setInterval() Method:

The setInterval() method repeats a given function at every given time interval.

Syntax:

window.setInterval(function, milliseconds);

There are two parameters that accepted by setInterval() method

function:

the first parameter is the function to be executed

milliseconds:

indicates the length of the time interval between each execution.

The setInterval() method executes the function infinitely hence there is a method called as clearInterval() to stop the execution of the setInterval().

Syntax:

window.clearInterval(value)

Q.4 how can you handle Async code in JavaScript ?

In JavaScript, there are two common ways to work with asynchronous operations, then/catch method chaining and async/await. Both methods can be used to handle promises, which are objects that represent the eventual completion (or failure) of an asynchronous operation.

then/catch method chaining is a more traditional way to handle asynchronous operations, while async/await is a newer syntax that provides a more concise and easier-to-read alternative

Q.5 What are Callbacks & Callback Hell ?

Callback:

A callback is a function that is passed as an argument to another function that executes the callback based on the result. They are basically functions that are executed only after a result is produced. Callbacks are an important part of asynchronous JavaScript.

Callback Hell:

Callback Hell is essentially nested callbacks stacked below one another forming a pyramid structure. Every callback depends/waits for the previous callback, thereby making a pyramid structure that affects the readability and maintainability of the code.

Q.6 What are Promises & Explain Some Three Methods of Promise

Promise:

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

There are 3 states of the Promise object-

Pending:

While a Promise object is "pending" (working), the result is undefined.

Resolved:

When a Promise object is "fulfilled", the result is a value.

Rejected:

When a Promise object is "rejected", the result is an error object.

Q.7 What’s async & await Keyword in JavaScript

Aysnc:

An async function declaration creates an AsyncFunction object. Each time when an async function is called, it returns a new Promise which will be resolved with the value returned by the async function, or rejected with an exception uncaught within the async function

Syntax

```
Async function myfirstfunction() {
return "Hello World"
}

  getData().then(data => console.log(data)); // Hello World
```

Await:

Await function is used to wait for the promise. It could be used within the async block only. It makes the code wait until the promise returns a result. It only makes the async block wait.

Syntax:

```
// Await function works only inside the async function
let value = await promise;
```

Q.8 Explain Purpose of Try and Catch Block & Why do we need it?

The try...catch statement is comprised of a try block and either a catch block, a finally block, or both. The code in the try block is executed first, and if it throws an exception, the code in the catch block

will be executed. The code in the finally block will always be executed before control flow exits the entire construct.

The catch block contains code that is executed if and when the exception handler is invoked. The runtime system invokes the exception handler when the handler is the first one in the call stack whose ExceptionType matches the type of the exception thrown.

Q.9 Explain fetch

fetch() method :

The fetch() method in JavaScript is used to request data from a server. The request can be of any type of API that returns the data in JSON or XML. The fetch() method requires one parameter, the URL to request, and returns a promise.

Syntax:

```
fetch('url')
  .then(response => response.json())
  .then(data => console.log(data));
```

Q.10 How do you define an asynchronous function in JavaScript using async/await?

The async function declaration creates a binding of a new async function to a given name. The await keyword is permitted within the function body, enabling asynchronous, promise-based behavior to be written in a cleaner style and avoiding the need to explicitly configure promise chains.