Assignment-5

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

Ans.

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 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.

Q.2 What are Web Apis ?

Ans.

- API stands for **A**pplication **P**rogramming **I**nterface.
- 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.

- The Web API has methods and properties that can extend the functionality of the browser.

Q.3 Explain SetTimeOut and setInterval ?

Ans. Both setTimeout and setInterval are built-in functions in JavaScript used for executing code after a specific delay. They are often used to introduce time-based behavior in JavaScript applications, such as delaying an action or repeatedly executing code at a certain interval.

Eg: Set time out

```
setTimeout(function() {
  console.log('Delayed message');
}, 2000);
```

Eg: SetInterval

```
setInterval(function() {
  console.log('Interval message');
}, 1000);
```

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

Ans.

- We can handle async function by using promise, async/await,
- In Promise there is a concept of resolve and reject.

Eg;

```
function fetchData() {
  return new Promise(function(resolve, reject) {
    // Asynchronous operation
    setTimeout(function() {
      const data = 'Some data';
      resolve(data);
    }, 2000);
  });
}
```

```
fetchData()
  .then(function(data) {
    console.log(data);
  })
  .catch(function(error) {
    console.error(error);
  });
```

Q.5 What are Callbacks & Callback Hell ?

Ans.

- 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.

On the other hand

- 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

Ans.

- Because of callback hell we need to use the promise.
- A Promise is created when we are unsure of whether or not the assigned task will be completed or not.
- The Promise object represents the eventual completion or failure of an async(asynchronous) operation and its resulting value.
- As the name suggests from real life itself, a Promise is either kept or broken. A Promise is always in one of the following states:

fulfilled: Action related to the promise succeeded.
rejected: Action related to the promise failed.
pending: Promise is still pending i.e not fulfilled or rejected yet.
settled: Promise has been fulfilled or rejected

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

Ans.

- async: The async keyword is used to declare an asynchronous function. It allows the function to use the await keyword inside it and ensures that the function always returns a Promise.

- The await keyword is used inside an async function to pause the execution of the function until a Promise is resolved. It can only be used inside an async function.

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

Ans.

- The try and catch blocks are used in JavaScript for error handling.
- They provide a way to handle and recover from errors that may occur during the execution of code.
- The try block contains the code that might throw an error, while the catch block is used to handle and respond to any errors that occur.
- Prevent Program Termination: By wrapping potentially error-prone code in a try block and providing error handling in the catch block, you can prevent your program from terminating abruptly when an error occurs.

Q.9 Explain fetch

Ans

- fetch is a built-in JavaScript function introduced in ES6 (ECMAScript 2015) that provides an easy and modern way to make HTTP requests and handle responses asynchronously.
- It allows you to send network requests to a server and handle the returned data.
- Fetch is based on async and await
- The fetch() method starts the process of fetching a resource from a server.
- The fetch() method returns a Promise that resolves to a Response object.

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

Ans.

- async: The async keyword is used to declare an asynchronous function. It allows the function to use the await keyword inside it and ensures that the function always returns a Promise.

- The await keyword is used inside an async function to pause the execution of the function until a Promise is resolved. It can only be used inside an async function.

- The async keyword is used to define an asynchronous function. It allows you to use the await keyword inside that function. When an async function is called, it always returns a Promise. This makes it easier to work with asynchronous code, as the function's return value can be handled with .then() or await.