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

Ans:- Synchronous- happens at the same time. Stops Execution of further code until this is done because of this stopping further execution. Synchronous code is called blocking. Blocking in sense of no other code will be executed.

Asynchronous- doesn't happen at the same time. Asynchronous code is called non-blocking because it doesn't block further code from running.

Q.2 What are Web Apis ?

Ans:- A Web API is an application programming interface for the Web. A Browser API can extend the functionality of a web browser.

Q.3 Explain SetTimeOut and setInterval ?

Ans:=setTimeout allows us to run a function once after the interval of time.

setTimeout(() => {

    console.log("welcome to js course");

}, 3000);

setInterval allows us to run a function repeatedly, starting after the interval of time, then repeating continuously at that interval.

setInterval(() => {

   console.log("hello");

}, 3000);

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

Ans:- JavaScript provides three methods of handling asynchronous code: callbacks, which allow you to provide functions to call once the asynchronous method has finished running; promises, which allow you to chain methods together; and async/await keywords, which are just some syntactic sugar over promises.

Q.5 What are Callbacks & Callback Hell ?

Ans:- 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.

Callback Hell: Callback Hell is essentially nested callbacks stacked below one another forming a pyramid structure.

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

Ans:- Promises are used to handle asynchronous operations in JavaScript.

Three Methods of Promise:-

1. then() :- The then() method is used with the callback when the promise is successfully fulfilled or resolved.
2. catch():- The catch() method is used with the callback when the promise is rejected or if an error occurs. For example,
3. finally():-The finally() method gets executed when the promise is either resolved successfully or rejected.

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

Ans:- The async keyword is used to define an asynchronous function, which returns a AsyncFunction object. The await keyword is used to pause async function execution until a Promise is fulfilled, that is resolved or rejected, and to resume execution of the async function after fulfillment.

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

Ans:- The try statement allows you to define a block of code to be tested for errors while it is being executed. The catch statement allows you to define a block of code to be executed, if an error occurs in the try block.

Q.9 Explain fetch

Ans:- The fetch() method in JavaScript is used to request data from a server.

const url="https://restcountries.com/v2/all";

fetch(url)

.then((result)=>result.json())

.then((data)=>{

  console.log(data);

})

.catch((error)=>{

  console.error(error)

})

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

Ans:- An asynchronous JavaScript function can be created with the async keyword before the function name, or before () when using the arrow function syntax. An async function returns a promise.

const userOne=()=>{

    return "User One";

}

const userTwo=()=>{

    return new Promise((resolve,reject)=>{

        setTimeout(()=>{

            resolve("User Foud")

        },2000);

    })

}

const userThree=()=>{

    return "User Three"

}

const wait=async()=>{

    let guestOne = userOne();

    console.log(guestOne);

    let guestTwo=await userTwo();

    console.log(guestTwo);

    let guestThree=userThree();

    console.log(guestThree);

}

wait()