1 ans: Synchronous, sometimes referred to as “sync,” and asynchronous, also known as “async,” are two different types of programming models.

Asynchronous is a multithreaded model that’s most applicable to networking and communications. Asynchronous is a non-blocking architecture, which means it doesn’t block further execution while one or more operations are in progress.

Synchronous is known as a blocking architecture and is ideal for programming reactive systems. As a single-thread model, it follows a strict set of sequences, which means that operations are performed one at a time, in perfect order.

2 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.

3 ans:The setTimeout() method is used to call a function after a certain period of time. The setInterval() Javascript method is used to call a function repeatedly at a specified interval of time. setTimeout() is cancelled by clearTimeout() method, and setInterval() is cancelled by clearInterval() method.

4 ans: 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.

5 ans: The callback function passed is used to sum up the elements of the array.

[Callback](https://www.geeksforgeeks.org/javascript-callbacks/)**:** 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.

6 ans: In JavaScript, a promise is a good way to handle asynchronous operations. It is used to find out if the asynchronous operation is successfully completed or not.

A promise may have one of three states.

* Pending
* Fulfilled
* Rejected

A promise starts in a pending state. That means the process is not complete. If the operation is successful, the process ends in a fulfilled state. And, if an error occurs, the process ends in a rejected state.

For example, when you request data from the server by using a promise, it will be in a pending state. When the data arrives successfully, it will be in a fulfilled state. If an error occurs, then it will be in a rejected state.

7 ans: Async and await are built on promises. The keyword “async” accompanies the function, indicating that it returns a promise. Within this function, the await keyword is applied to the promise being returned. The await keyword ensures that the function waits for the promise to resolve.

8 ans: A try block is the block of code (contains a set of statements) in which exceptions can occur; it's used to enclose the code that might throw an exception. The try block is always followed by a catch block, which handles the exception that occurs in the associated try block. A try block must be used within the method and it must be followed by a catch block(s) or finally block or both.

**Syntax of Try block**

1. try{
2. //code that may throw an exception
3. }catch(Exception) {
4. //code
5. }

The catch block catches and handles the try block exceptions by declaring the type of exception within the parameter. The catch block includes the code and it is executed if an exception inside the try block occurs. The catch block is where you handle the exceptions; so this block must be follow the try block.

Try catch in Java ensures that the flow of the program doesn’t break when the exception occurs during the running of program. For example, if there is a program that has a bunch of statements and an exception occurs mid way after executing certain statements of the application (program) then the statements after the exception will not execute and the program will terminate abruptly by handling the exception we make sure that all the statements execute and the flow of program doesn’t break during its runtime.

9 ans: The Fetch API is a modern interface that allows you to make HTTP requests to servers from web browsers.

If you have worked with XMLHttpRequest (XHR) object, the Fetch API can perform all the tasks as the XHR object does.

In addition, the Fetch API is much simpler and cleaner. It uses the Promise to deliver more flexible features to make requests to servers from the web browsers.

The fetch() method is available in the global scope that instructs the web browsers to send a request to a URL.

10 ans: The word “async” before a function means one simple thing: a function always returns a promise. Other values are wrapped in a resolved promise automatically.

For instance, this function returns a resolved promise with the result of 1;

async function f() {

return 1;

}

f().then(alert);

The keyword await makes JavaScript wait until that promise settles and returns its result.

Here’s an example with a promise that resolves in 1 second:

async function f() {

let promise = new Promise((resolve, reject) => {

setTimeout(() => resolve("done!"), 1000)

});

*let* result *=* *await* promise*;* *// wait until the promise resolves (\*)*

alert(result); // "done!"

}

f();