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

| **S. No.** | **Synchronous Transmission** | **Asynchronous Transmission** |
| --- | --- | --- |
| **1.** | In [Synchronous transmission](https://www.geeksforgeeks.org/synchronous-data-transfer-in-computer-organization/), data is sent in form of blocks or frames. | In [Asynchronous transmission](https://www.geeksforgeeks.org/asynchronous-serial-data-transfer/), data is sent in form of bytes or characters. |
| **2.** | Synchronous transmission is fast. | Asynchronous transmission is slow. |
| **3.** | Synchronous transmission is costly. | Asynchronous transmission is economical. |
| **4.** | In Synchronous transmission, the time interval of transmission is constant. | In Asynchronous transmission, the time interval of transmission is not constant, it is random. |
| **5.** | In this transmission, users have to wait till the transmission is complete before getting a response back from the server. | Here, users do not have to wait for the completion of transmission in order to get a response from the server. |
| **6.** | In Synchronous transmission, there is no gap present between data. | In Asynchronous transmission, there is a gap present between data. |
| **7.** | Efficient use of transmission lines is done in synchronous transmission. | While in Asynchronous transmission, the transmission line remains empty during a gap in character transmission. |
| **8.** | The start and stop bits are not used in transmitting data. | The start and stop bits are used in transmitting data that imposes extra overhead. |

**Q.2** What are Web Apis ?

**Answer:** A Web API is an application programming interface for the Web. It is a concept and not a technology. We can build Web API using different technologies such as Java, .NET etc. In particular, the most common categories of browser APIs you’ll use are: APIs for manipulating documents loaded into the browser. The most obvious example is the DOM (Document Object Model) APIs that fetch data from the server to update.

**Q.3** Explain Set Time Out and Set Interval?

**Answer:** set Timeout() and set Interval() are both methods of the HTML DOM Window object.

setTimeout() allows us to run a function once after the interval of time.

setInterval() allows us to run a function repeatedly, beginning after some time, then repeating continuously at the given interval.

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

**Answer:** To use async JavaScript, you need to:

* Use the async keyword before a function to make it return a promise.
* Use the await keyword inside a async function to pause the execution until the promise is resolved.
* Assign the resolved value of the promise to a variable using the await keyword.
* Use try and catch blocks to handle errors in async functions.
* Use top level await or then and catch methods to handle promises outside of async functions.

**Q.5** What are Callbacks & Callback Hell ?

**Answer:** A callback is a function that is passed as an argument to another function and is executed after its parent function has completed.

Callback Hell is a term used to describe the situation when callbacks are nested within other callbacks, making the code difficult to read and maintain.

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

**Answer:** A Promise is an object representing the eventual completion or failure of an asynchronous operations.

Some methods of Promise are:

1. Promise.all (): This method takes an array of promises and return a new promise that resolves when all of the promises in the array have resolved.
2. Promise.race(): This methods takes an array of promises and returns a new promise that resolves or rejects as soon as one of the promises in the array resolves or rejects.
3. Promise.any(): This method takes an array of promises and returns a new promise that resolves as soon as one of the promises in the array resolves.

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

**Answer:** The **async** and **await** keywords are used for working with asynchronous code in a synchronous like manner.

The **async** keyword is used to declare a function as asynchronous, which means that it will return a promise. Inside an async function, the **await** keyword can be used to pause the execution of the function until a promise is resolved.

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

**Answer:** The **try** and **catch** blocks are used for error handling in JavaScript.

The purpose of the **try**block is to enclose the code that might throw an exception. If an exception is thrown, the control is passed to the **catch** block.

The purpose of the **catch** block is to handle the exception that was thrown in the **try** block.

We need error handling because it helps us to gracefully handle errors that might occur during the execution of our code. It also helps us to debug our code by providing us with information about what went wrong.

**Q.9** Explain fetch ?

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

The **fetch()** method 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()** methods requiresone parameter, the URL to request, and returns a promise.

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

**Answer:** To define an asynchronous function in JavaScript using **async/await**, you can use the**async** keyword before the function declaration.

Here’s an example:

**async function myFunction() {**

**// code goes here**

**}**

The **async** keyword tells JavaScript that the function is asynchronous. You can then use the **await** keyword inside the function to wait for a promise to resolve before continuing execution.