Assignment Questions

1. Synchronous and Asynchronous have a very basic common difference and that lies in there waiting time which they take to complete a task for instance in case of Synchronous Programming Languages they wait until the previous process is executed whereas in the case of Asynchronous Programming languages if the time taken by a process is large then the control would shift to the next line and it will get executed and once it is done the line which had been taking time then that is going to be executed.
2. Web API’s (Application Programming Interfaces) are interfaces provided by web browsers that allow developers to interact with various web technologies and access features like manipulating the DOM, making HTTP requests, handling timers, working with local storage.
3. setTimeout and setInterval.
   1. setTimeOut is a function that schedules the execution of a callback function after a specified delay in milliseconds.
   2. setInterval is similar to ‘setTimeout’, but it repeatedly executes the callback function at a specified interval until cleared.
4. ECMAScript 6 also commonly known as ES6 was introduced in 2015 and some of its ground breaking features include block scoped variables which are **let** and **var,** *arrow functions*, *classes, modules, templates, literals, destructuring assignments, spread operators and enhanced object literals.*
5. The major difference between **let** and **const** is in their reassignment behaviour. That simply means that any value that is assigned to a variable declared by **let** can be reassigned. But, in case of **const** if something is given some certain value during initialisation cannot be reassigned with some other value. But, It is also important to note that **const** variables are not immutable, which means that their properties can still be modified if they are object or arrays.
6. Template literals , also known as template strings *are a feature in ES6* that allows for more flexible string formatting. They are enclosed within backticks(`) instead of single or double quotes. **They also support multiline strings, interpolation of variables using “***${}***”** and the ability to include expressions or function calls directly within the string.

Example:

const name = ‘Anurag’;

const age = 21;

const greeting = `Hello, my name is ${name} and I am ${age} years old.`

console.log(greeting);

1. Both ‘map’and ‘forEach’ are array methods/functions in JavaScript. The main difference between them is that **map** creates a new array with the results of calling a provided function on every element, while **forEach** executes a provided function for each array element, but does not return a new array.

Example usage of **map:**

const numbers = [1,2,3,4];

const doubledNumbers = numbers.map(num=> num\*2);

console.log(doubledNumbers);

Example usage of **forEach:**

const numbers = [1,2,3,4];

numbers.forEach(num=> console.log(num));

1. Destructuring assignment is a feature of ES6 that allows you to extract values from objects into distinct variables.

Example usage with Objects:

const person = {name: ‘John’, age:25, city: ‘New York’};

const {name, age} = person;

console.log(name, age); // John 25

Example usage with arrays:

const numbers = [1,2,3];

const[first, second] = numbers;

console.log(first, second); // 1 2

1. ES6 introduced the ability to set default values for function parameters. If a parameter is not passed or is explicitly set as ‘**undefined**’**,**  the default value will be used.

Example:

Function greeting(name = ‘Anonymous’){

console.log(`Hello, ${name}!`);

}

greeting();

greeting(‘John’);

1. The purpose of the spread operator (…) is to expand elements of an array or properties of an object. It allows you to create copies or merge arrays/objects easily.

We can even say it works similar to concatenation but for arrays and objects in JavaScript.

Example usage with arrays:

const arr1 = [1,2,3];

const arr2 = […arr1,4,5,6];

console.log(arr2); // [1,2,3,4,5,6]

Example usage with objects:

const obj1 = {x: 1, y: 2};

const obj2 = {…obj1, z: 3};

console.log(obj2); // {x: 1, y: 2, z: 3}

The spread operator is versatile and can be used in various scenarios, such as function arguments, array manipulation, Object Cloning and much more.