**S7\_\_ AsyncronousProgramming & Async Await & TryCatchBlock**

**Asynchronous JavaScript:** 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. This may not look like a big problem but when you see it in a bigger picture you realize that it may lead to delaying the User Interface.

* **Async/Await :- its used against with the promise.**
* **Async: -** start with async keyword. It makes function return promise. It was synchronous and it check that we are not breaking the execution thread. it runs synchronously until it reaches first await expression at which point the method is suspended until awaited task is completed. If the method don’t contain await keyword it run asynchronously.It operates asynchronously via event loop. It always return a value. If not retruing js will auto return resolve as value.

**Await:-** is used to wait for a promise. It can only used inside and async function. If the promise is rejected tge await expression throws the rejected values.An await splits execution flow, allowing the caller of the async function to resume execution. If this await is the last expression executed by its function, execution continues by returning to the function's caller a pending Promise for completion of the await's function and resuming execution of that caller.

* **Exception Handling Statement: --**:-- exception handling is used to manage the exception thrown by the program. It has different types. Error will be handeled by CLR. If the catch block not found then it will throw the unhandled exception throw that method and CLR.
  + **Try:-** the all logical code written in it but if the error is occurred then it will send the error to catch block and catch block will handle that error. In the program error is occurred then it will break the program.
  + **Catch :-** if the error is thrown by try block it will caught in catch. We can write multiple catch block but when the try block got exception it will terminate the program. And not moved or check the next catch block. Trow keyword is used to catch the error. Or return the error.

Program

const objectData = [

    { name: "Imran", add: "Pune", age: "33" },

    { name: "Amit", add: "Pune", age: "33" },

    { name: "Ankit", add: "Pune", age: "39" },

    { name: "Anikesh", add: "Bombay", age: "33" },

];

function getUsers() {

    setTimeout(() => {

        let output = '';

        objectData.forEach((objectData, index) => {

            output += `<li>${objectData.name}</li>`; //Template Syntax

            output += `<li>${objectData.age}</li>`;

            output += `<li>${objectData.add}</li>`;

        });

        document.body.innerHTML = output;

    }, 1000);

}

function createUsers(userData) {

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

        setTimeout(() => {

            objectData.push(userData);

            const error = false;

            if (!error) {

                resolve();

            }

            else {

                reject("Something is not Good");

            }

        }, 2000);

    });

}

// createUsers({ name: "Abhilasha", add: "Bombay", age: "33" }).then(getUsers).catch(errorShow => console.log(errorShow));   //promise

//Async Await...

async function display() {

    await createUsers({ name: "Abhilasha", add: "Bombay", age: "33" });

    getUsers();

}

console.log("Calling Display ");

console.log(display());

//Try Catch

try {

    console.log("Try Block...");

    functionNames();

} catch (error) {

    console.log("Function not Define...");

    console.log(error.message);

}

finally{

    console.log("Always Execute");

}