**Navachethan.M - 1NT18IS099**

**Web Assignment on Asynchronous function and await keyword.**

navachethan-async.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <h1>Asynchronous function and await </h1><br/>

    <h2>Asynchronous function implementation on webpage</h2>

    <button id="btn">submit</button>

    <p id="demo"></p>

    <h2>To resolve  promise using async and await</h2>

    <button id="btn1">Resolve</button>

    <button id="btn2">Resolve with setTimeOut</button>

    <p id="demo1"></p>

    <p id="demo2"></p>

    <h2>to print promise in alert</h2>

    <button id="btn3">Resolve</button>

    <button id="btn4">Resolve with SetTimeOut</button>

    <script src="navachethan-async.js"></script>

</body>

</html>

navachethan-async.js

// async function example

/\*async function f() {

console.log('Asynchronous  function output.');

return Promise.resolve(1);

}

f(); \*/

/\*async function f() {

console.log('Asynchronous function with promise.');

return Promise.resolve(1);

}

f().then(function(result) {

console.log(result)

});\*/

// a promise

/\*let promise = new Promise(function (resolve, reject) {

setTimeout(function () {

resolve('Promise resolved with time-stamp')}, 8000);

});

// async function

async function asyncFunc() {

// wait until the promise resolves

let result = await promise;

console.log(result);

console.log('hello javascript');

}

// calling the async function

asyncFunc();\*/

//The other way you can handle an error is by using try/catch block

// a promise

/\*let promise = new Promise(function (resolve, reject) {

    setTimeout(function () {

    resolve('Promise resolved with time-stamp')}, 8000);

    });

    // async function

    async function asyncFunc() {

    try {

    // wait until the promise resolves

    let result = await promise;

    console.log(result);

    }

    catch(error) {

    console.log(error);

    }

    }

    // calling the async function

    asyncFunc(); // Promise resolved \*/

    // implementation of async and await on webpage

    let fn\_btn = document.getElementById("btn");

    fn\_btn.addEventListener("click", ()=>{

    async function myDisplay() {

        let myPromise = new Promise(function(myResolve, myReject) {

          setTimeout(function() { myResolve("Implementation of asynchronous function"); }, 3000);

        });

        document.getElementById("demo").innerHTML = await myPromise;

      }

      myDisplay();

    });

    let fn\_btn1 = document.getElementById("btn1");

    fn\_btn1.addEventListener("click", ()=>{

    async function myDisplay() {

        let myPromise = new Promise(function(myResolve, myReject) {

          myResolve("Promise has been resolved successfully.");

        });

        document.getElementById("demo1").innerHTML = await myPromise;

      }

      myDisplay();

    });

    let fn\_btn2 = document.getElementById("btn2");

    fn\_btn2.addEventListener("click", ()=>{

    async function myDisplay() {

        let myPromise = new Promise(function(myResolve, myReject) {

          setTimeout(function() { myResolve("Promise resolved"); }, 3000);

        });

        try {

          // wait until the promise resolves

          let result = await promise;

          console.log(result);

          }

          catch(error) {

          console.log(error);

          }

          document.getElementById("demo2").innerHTML = await myPromise;

          }

      myDisplay();

    });

    let fn\_btn3 = document.getElementById("btn3");

    fn\_btn3.addEventListener("click", ()=>{

    async function myDisplay() {

        let myPromise = new Promise(function(myResolve, myReject) {

          myResolve(alert("Promise has been resolved successfully."));

        });

      }

      myDisplay();

    });

    let fn\_btn4 = document.getElementById("btn4");

    fn\_btn4.addEventListener("click", ()=>{

    async function myDisplay() {

        let myPromise = new Promise(function(myResolve, myReject) {

          setTimeout(function() { myResolve(alert("Promise resolved")); }, 3000);

        });

        try {

          // wait until the promise resolves

          let result = await promise;

          console.log(result);

          }

          catch(error) {

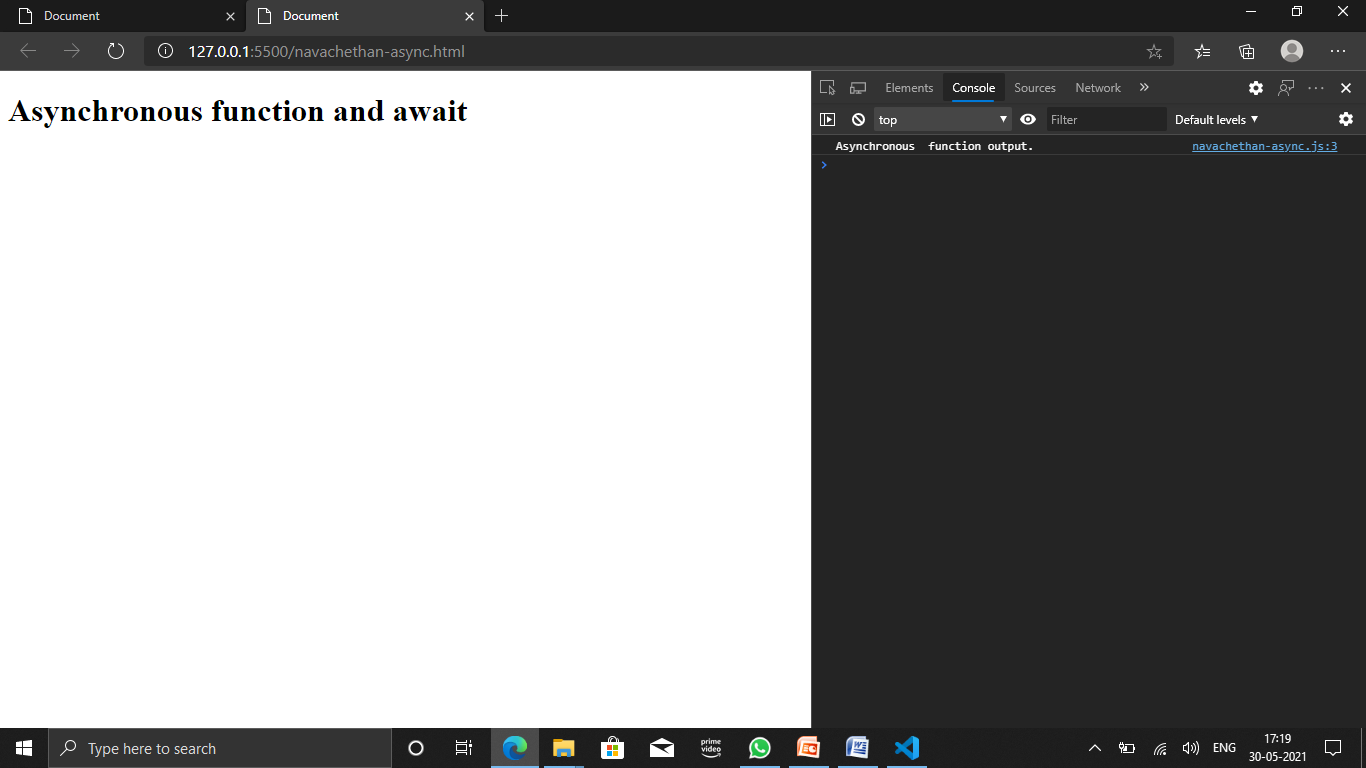
          console.log(error);

          }

          }

      myDisplay();

    });

Output:

