## **Week 2 Exploration**

* **Zainab Nusaiba**

## **Problem Statement:**

setTimeout(() => { console.log("Hello world!"); }, 2000);

for(var i=0; i < 10; i++) {

runIt(i);

}

function runIt(i) {

setTimeout(function(){

console.log("In-loop "+(i+1)+" Sec");

}, i \* 1000);

}

## **Objective:**

* Document the outcome of the above function.
* Are the sequences of events synchronous? If so, how would you make it asynchronous or vice-versa?

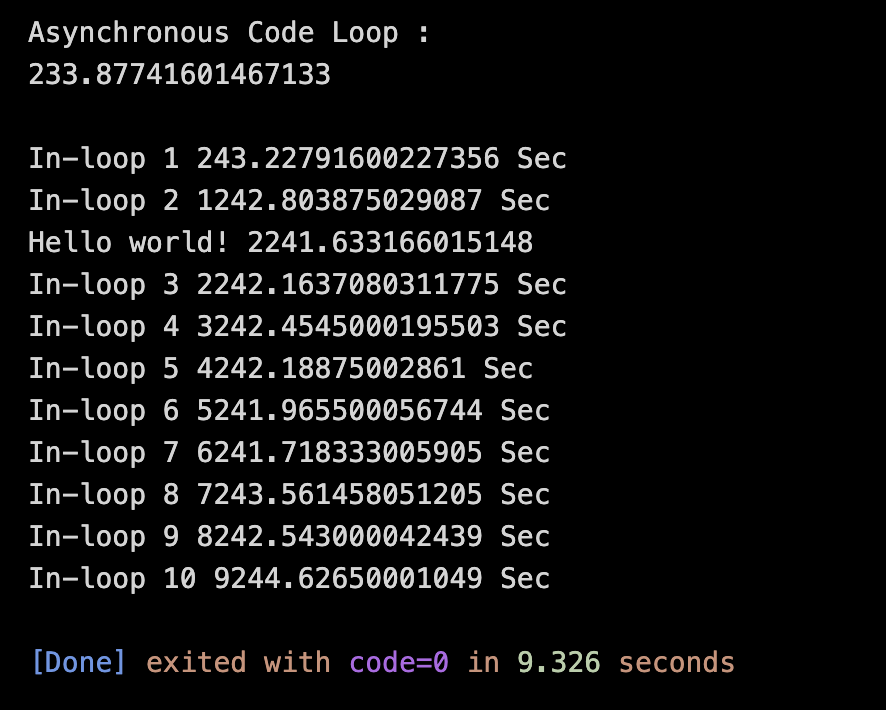
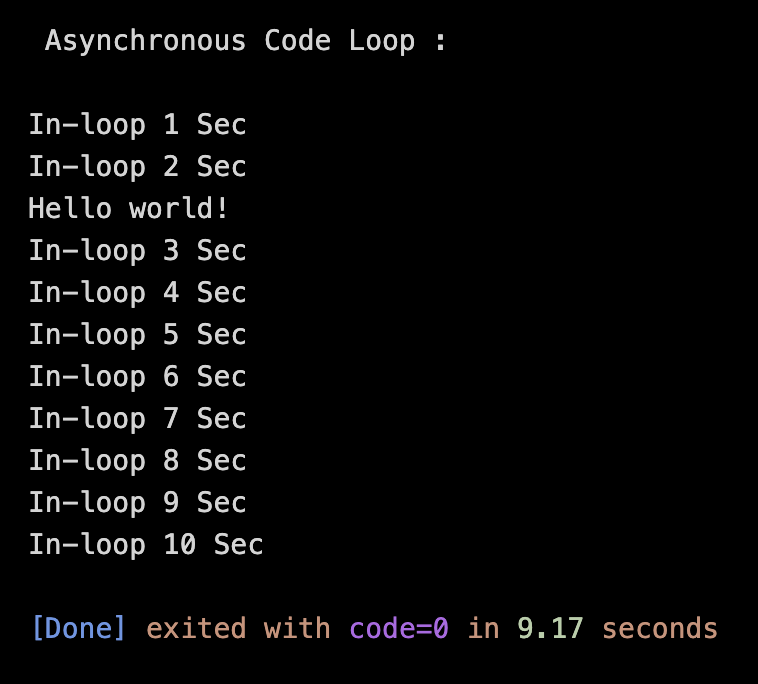
## **Outcome:**

The output of the above program is given below. Some of the functionings involved in the program are :

**setTimeout()**

An asynchronous method executes the given data or anonymous function after the timer. This shows that it consists of two parameters the anonymous function and time represented in milliseconds

Eg: setTimeout ( greet() , 2000 )



## **Are they Synchronous:**

No, the sequences of events are Asynchronous.

## **Making it Synchronous:**

It is asynchronous because setTimeout() is an asynchronous method. It also interrupts the program between the loop calls. And, setTimeout() can be written synchronously by using

**1. Callback**

This is done by passing a function into a timeout as an argument then the function will be executed after the timeout is executed.

**2. async, await, and promise**

Await is used along with the async function. Async makes the function asynchronous and when paired with the await, it waits for the acknowledgment and then executes after that. Promises are the microtasks that can be called. It is usually executed after all the normal callbacks in the program.

## **Synchronous Code:**

console.log("\nSynchronous Code Loop :\n");

function helloMssg() {

return new Promise((res, rej) => {

setTimeout(() => {

res("Hello world!");

rej("Rejected");

}, 2000);

});

}

async function loopMssg() {

await helloMssg()

.then((success) => {

console.log(success);

})

.catch((error) => {

console.log(error);

});

for (var i = 0; i < 10; i++) {

runIt(i);

}

function runIt(i) {

setTimeout(function () {

console.log("In-loop " + (i + 1) + " Sec");

}, i \* 1000);

}

}

loopMssg();

**Synchronous Outcome:**

