Let take a eg of function.

//created a func with 1000 milli sec=>i sec timer

function x() {

  var i = 1;

  setTimeout(() => {

    console.log(i);

  }, 1000);

}

x();

//it pritns value of i after one sec.

function x() {

  var i = 1;

  setTimeout(() => {

    console.log(i);

  }, 1000);

  console.log("namaste js");

}

x();

//the above code will first prints 'namaste js' then after 1 sec  and print i

How and y it is printing like this means bcoz, this function passed to settimeout forms a closure , it remembers a reference to I and forms a closure.

Wherever this fucn goes , it takes the ref of I along with it.

Settimeout will takes this callback func and stores it in some place. And attach a timer to it.

And js will proceeds with next lines.

Once the timer expires (1000 ms completed) then it takes the func and runs by putting it in callstack.

So js doesn’t wait any time .

**Situation :**

Suppose printing 1,2,3,4,5 after each and every sec.