

Synchronous vs Asynchronous

Tuesday, March 15, 2022 3:39 PM

- Two activities are synchronous if they depend on each other as cause and action
- If two activities are independent of each other and they don't need any interaction and it doesn't matter if the work one after other or parallel we can say they are asynchronous

```

14 const findPrimes = (min,max)=>{
15     let result=[];
16
17     for(let i=min;i<max;i++){
18         if(isPrime(i)){
19             result.push(i);
20         }
21     }
22     return result;
23 }
24
25 //console.log('findPrimes(20,40)',findPrimes(20,40));
26
27 console.warn("finding primes between 2 and 500000");
28 let p2_5l= findPrimes(2, 500000);
29 console.log('p2_5l',p2_5l.length);
30
31 console.warn("finding primes between 2 and 100");
32 let p2_100= findPrimes(2, 100);
33 console.log('p2_100',p2_100.length);
34
35 console.warn("finding primes between 2 and 2000000");
36 let p2_2l = findPrimes(2, 2000000);
37 console.log('p2_2l',p2_2l.length);
--

```

• Synchronous

- I can't print the result till the time I get the result

• Asynchronous

- Not dependent on each other
- They can work parallelly if possible
- They are often unrelated
 - If not unrelated at least independent.

A long running synchronous Task

A look at 'normal' synchronous program

- ▶ Consider a function that talks to external system
- ▶ It may be long running task.
- ▶ Here our service is fetching details from different external systems.
- ▶ We return the result only when we get all the results.
- ▶ We can't go to next step till this function returns;

```

function getFlightInfo( param ) {
    var result=[];
    //fetch details from different airlines.
    return result;
}
let result1 = getFlightInfo("param1");
console.log(result1);
let result2 = getFlightInfo( param2 );
console.log(result2);

```

Must wait for this to finish

Before executing this

Async using Callback model

Async using callback

IMPORTANT

- ▶ We don't return the result using standard 'return' statement
- ▶ We pass a callback function that the long running task would call to convey the result.
- ▶ The caller need to pass 'next action' as a call back.

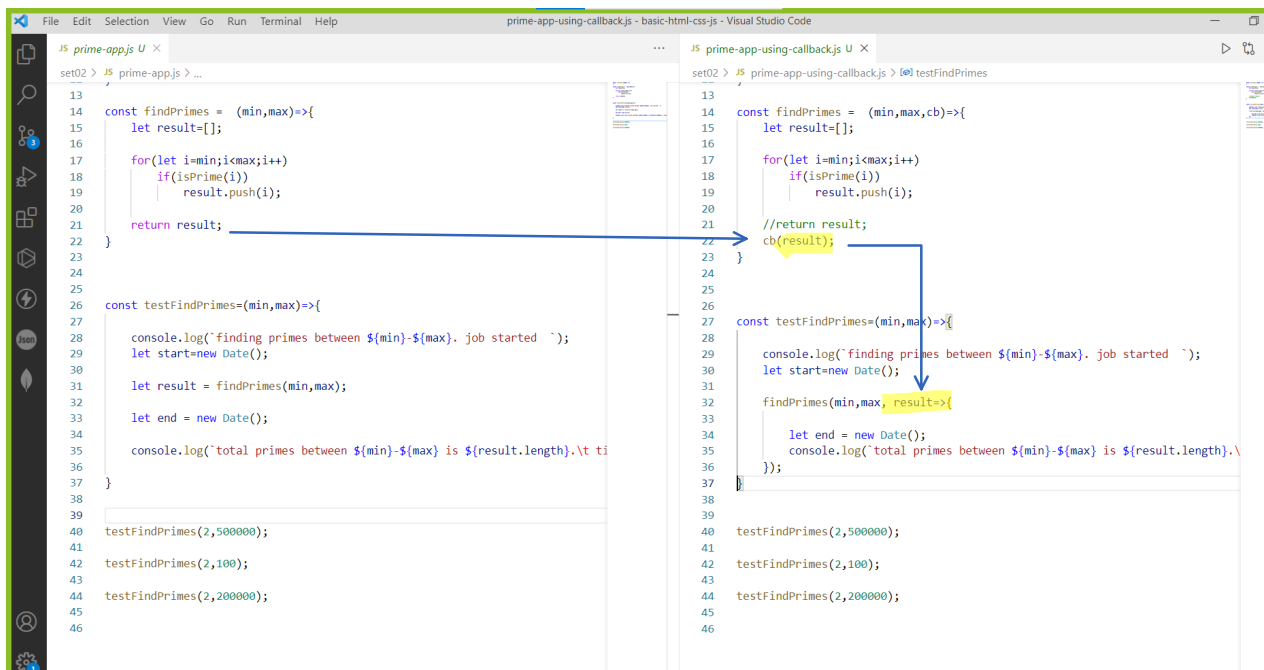
```
function getFlightInfo( param ,cb ) {
  var result=[];
  //fetch details from different airlines.

  cb (result);
}

getFlightInfo( param1 , result => console.log(result));
getFlightInfo( param2 , console.log);
```

Note
Both
callbacks
are same

- Simply using ; application as
- There should representing
- The api shoul
 - o Talk to
 - o Provide compoi
 - time/of



- The c
It is n