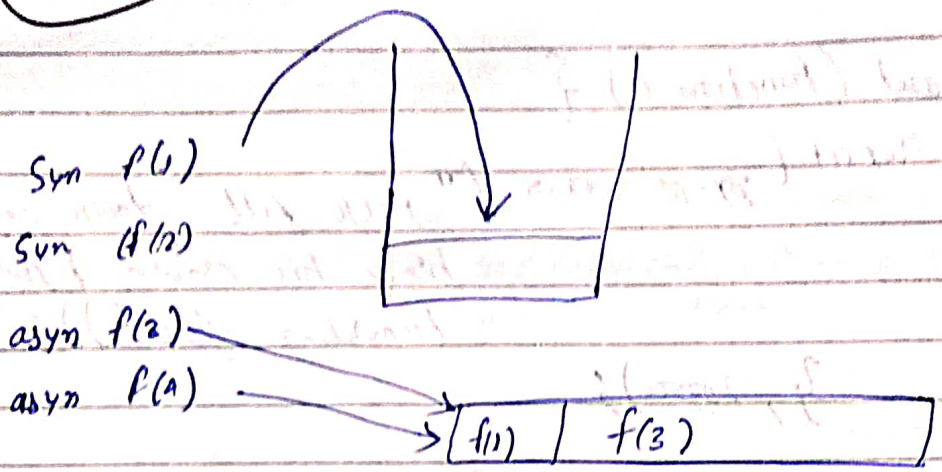
(1)

#12

01 Sep 2022

DATE : / /
PAGE NO. :



Event Loop :- also working background

f1() → f(3)

javascript Visualizer

1) Synchronous
2) Creating callback & micro task
3) once

Resolve :- Success fulfilled

Success → then
Failure → catch

(2)

DATE : / /
PAGE NO. :

async/wait

as

```
async f1() {
    return 1          // true condition
};
f1().then(console.log('success'));
```

asyn — alway return a promise (all type)

Why we use — Simplicity purpose (wrappes to of promise)

pupose — to return promise

await →

promise ^ executed settled then return.

Promise internal states

- Resolve
- Reject
- pending

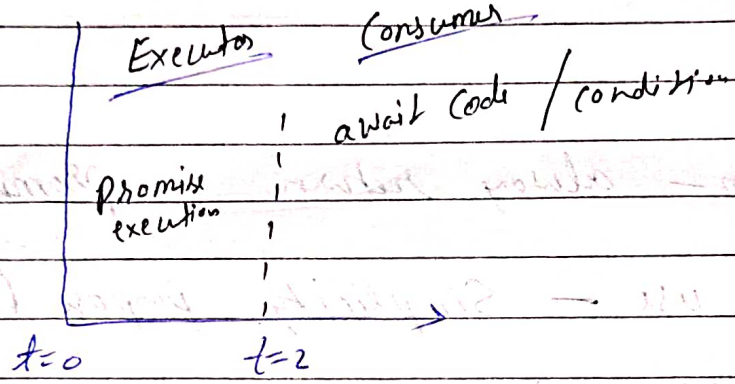
```
async function f() {
    let promise = new Promise((resolve, reject) =>
    {
        setTimeout(() => resolve("done"), 1000)
    });
```

(3)

DATE : / /
PAGE NO. :

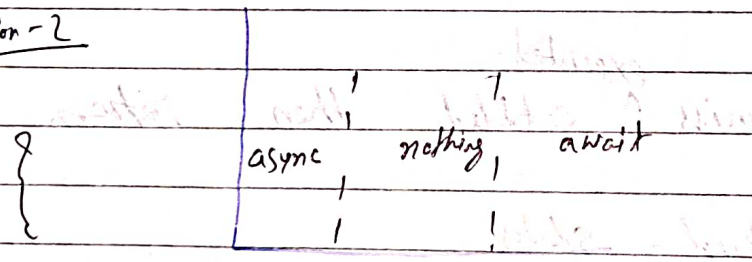
```
let result = await promise;
console.log('success');
}
f();
```

Condition ①

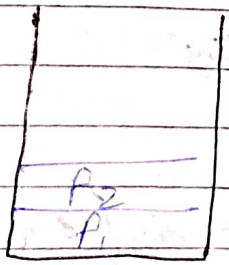


Promise code finished

Condition-2



Imp



Callstack

Syn f_1
Syn f_2
assign f_3
assign f_4

Callsync f_3

f_4

Micro Task

(4)

DATE : / /
PAGE NO. :

Priority

interpreter → promise condition

→ Let example

```
Const getTime clock1 = () => {
    return new Date().getHours()
    + ":"
    + new Date().getMinutes()
    + ":"
    + new Date().getSeconds() };

function getSynchronousMsg() { return
    "Hello"
}

function getHelloFromPromise() {
    return Promise.resolve("Hello-promise");
}

function getHelloWithDelay() {
    return new promise (function
    (resolve, reject) {
        setTimeout (function() {("hello");
    }, 4000);
```

Commitment, Focus, Dedicated, Hard Work, Discipline

DATE: / /

JS Auto Focus – Learning

@ Hyder Abbas

•

•

IT industry
Anything Possible

- Small small effort
- little-little doubt

putting effort

Var = "Alien"

Var a = 10

- Symbol

Var a = Symbol();

Var a = Symbol("Azmi"); → Identifier or description

It is type of Symbol

Var a = "Alien";
Var b = "Alien";
Console.log(a==b); true

Object prepare - convert JSON then server

DATE: / /

PAGE NO.:

(6)

Var a = Symbal ("Alien")
Var b = Symbal ("Alien");

console.log (a == = b); (False)

type

console.log (type of a); ⇒ Symbol
console.log (type of d); ⇒ String

(object create) let age = Symbal(); (Ages) ("age");

↓ identifier

let users = {
name : "hyder"
~~age~~ :
qualification : "BE"

Symbal reason — [age] : 28

};

console.log (users);

console.log (users.name) ⇒ hyder
console.log (users.age) ⇒ undefined
console.log (users [age]) ⇒ then ⇒ 28

• - All data type used except symbal
[] - Only symbals datatype

(7)

```
for (let key in user)
{
  console.log(key);
}
```

JSON `console.log (JSON.stringify (user));`

object to JSON conversion

Iterators & Generators

iterators → traversing or collection

Generators →

```
var a = [10, 20, 30, 40]
for (let i=0 ; i<a.length ; i++)
{
  console.log [a[i]];
}
```