Date: //
P. No:

Lecture 24 JS-DOM-III

18/02/23

[illegible]

→ Standard way to measure how long your code taken to run.
~~[illegible]~~ performance.now()

* **Reflow :-**
It is process of calculating the positions and geometrices of elements in the document, for the purpose of re-rendering part or all of the document

* **RePaint :-**
Show display pixel by pixel. It is faster than Reflow.

∴ When Reflow and RePaint is high then code optimization low whereas reflow / repaint is low code is faster.

* **Document fragment**
let element = document.CreateDocumentFragment('div');
∴ It is minimize / neglet Repaint and Reflow

* JS Single threading language
↳ processing one command at a time.
↳ ~~JS is synchronous language.~~

P.T.O

Date: 11
P. No:

[illegible]

*** Observation.**

(i) 'run-to-complete' nature to code

(II) JS does not execute multiple lines/function at a time.

```
function a()
{
  console.log('Hi');
}
function b()
{
  console.log("Hello");
}
a();
b();
```

start end end start

*** Callstack**

```
1  function a()
2  {
3     console.log("Hi")
4       b();
5  }
6  function b()
7  {
8     console.log("Hello");
9  }
10 a();
```

JS File

call stack

```
8 → c.log() × ②
4 → b() × ③
3 → c.log() × ①
10 → a() × ④
     main() × ⑤
```

→ when function calls ~~starts~~ add in call stack and when function finished entry remove from call stack.

*** Synchronous**

Occuring at the same time.

*** Async (Not-Synchronous)**

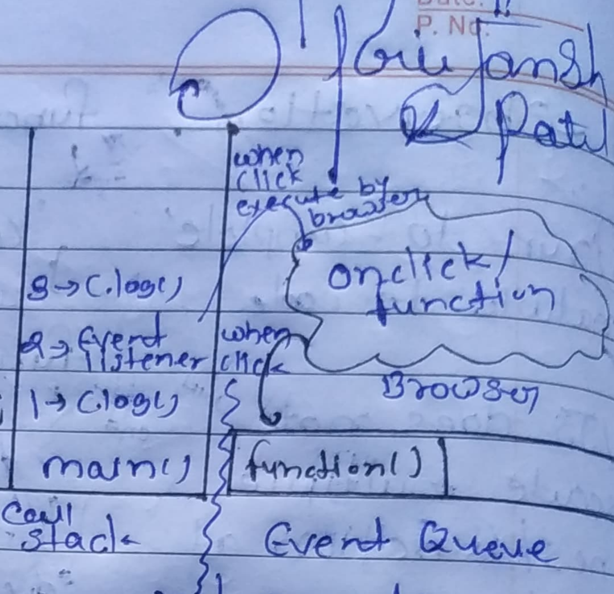
Date: !!
P. No.

Priyansh Patel

* Event Loop

Code :-

```
1) Console.log('Hi');
2) element.addEventListener
   ('click', function() {
        console.log('123') });
3) Console.log ('Hello');
```



not gurantee when click at that time it calls or not.

→ all entry comes and Remove in Call stack.

→ When event listener entry comes in Call Stack at that time it gives controll to browser and remove entry from call stack. ~~Broows~~

→ When clicked browser gives this function to event queue.

→ Event Queue executed this function when call stack is empty.

* Notes :-

(I) Async code uses JS Event loop
Ex → eventlistener, setTimeOut etc.

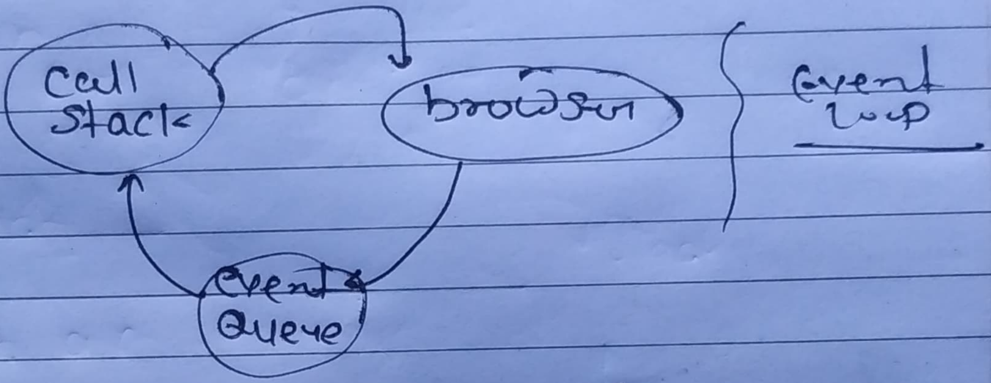
(II) Async code Handle by browser.

Date: !!
P. No:

[illegible]

(iii) Code comes browser to Event Queue.

(iv) When call stack empty event Queue send function in call stack to execute.



* Set Time Out() → [Async code]

```
SetTimeOut ( function () {
        Console.log(" JJP ");
    }, 4000 );
```

2 parameters
→ (i) function()
→ (ii) time in ms

→ in after that time function will execute and this is minimum time, time can be increased if call stack is not empty

mill second

minimum Time to be executed.