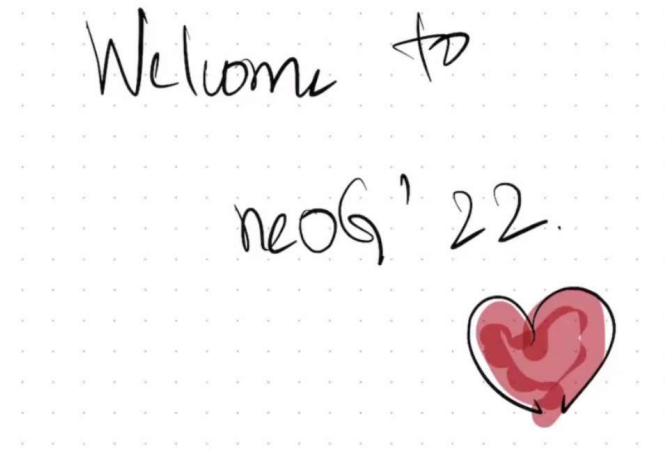
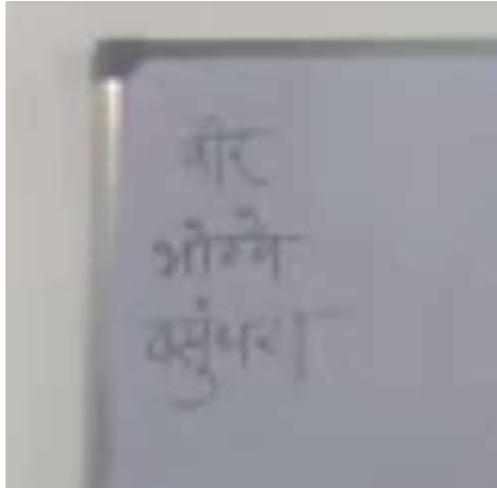


7-neog, callstack understanding



Blocking main thread, a bad idea!

Why do we need async?

If I am delegating my work to some people, how do I hear back?

Examples

Ex - 1

Ex-2

Questions

Why Event queue pushes only when Call stack is empty?

Does each tab has separate call stack, event queue?

We download assets as part of source along with JS, and JS hasn't executed yet so how come we initiate JS c...

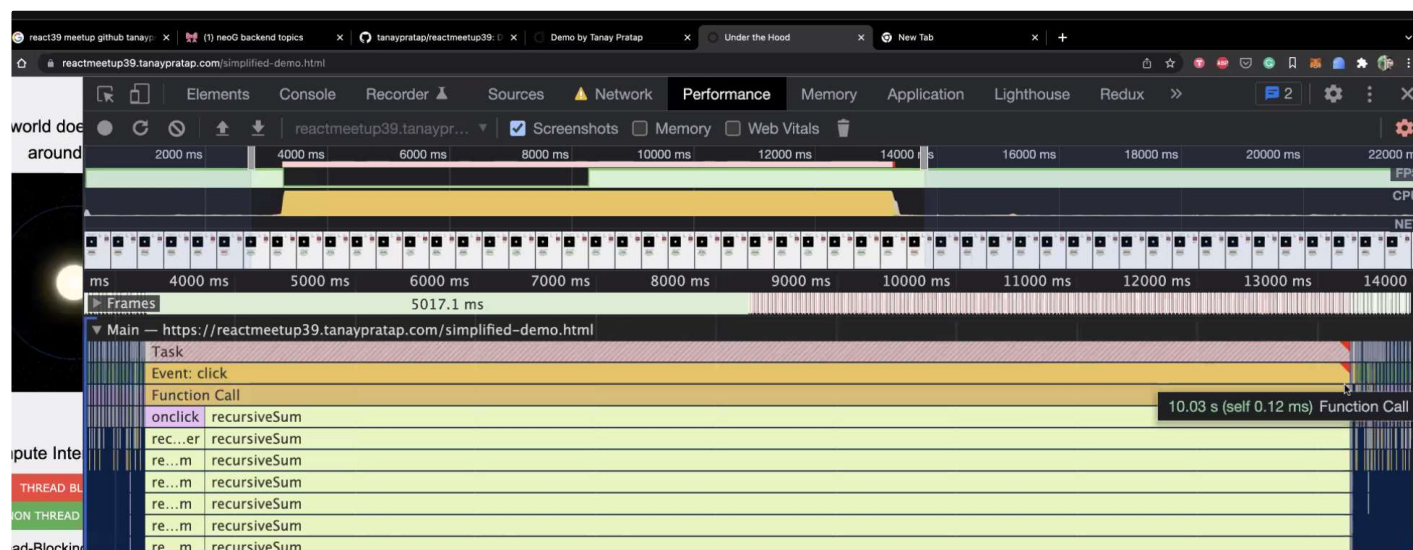
Tips

Websites mentioned

Homework

Blog Ideas

Blocking main thread, a bad idea!



- reactmeetup39.tanaypratap.com
- browser has a main thread, if we block the main thread, it will stop receiving user inputs, clicks and hence a bad ux.
- List

```
fetch() (LATER) .then(do smth with data) console.log("hey"); (NOW)
```

- multithreading, concurrency, async

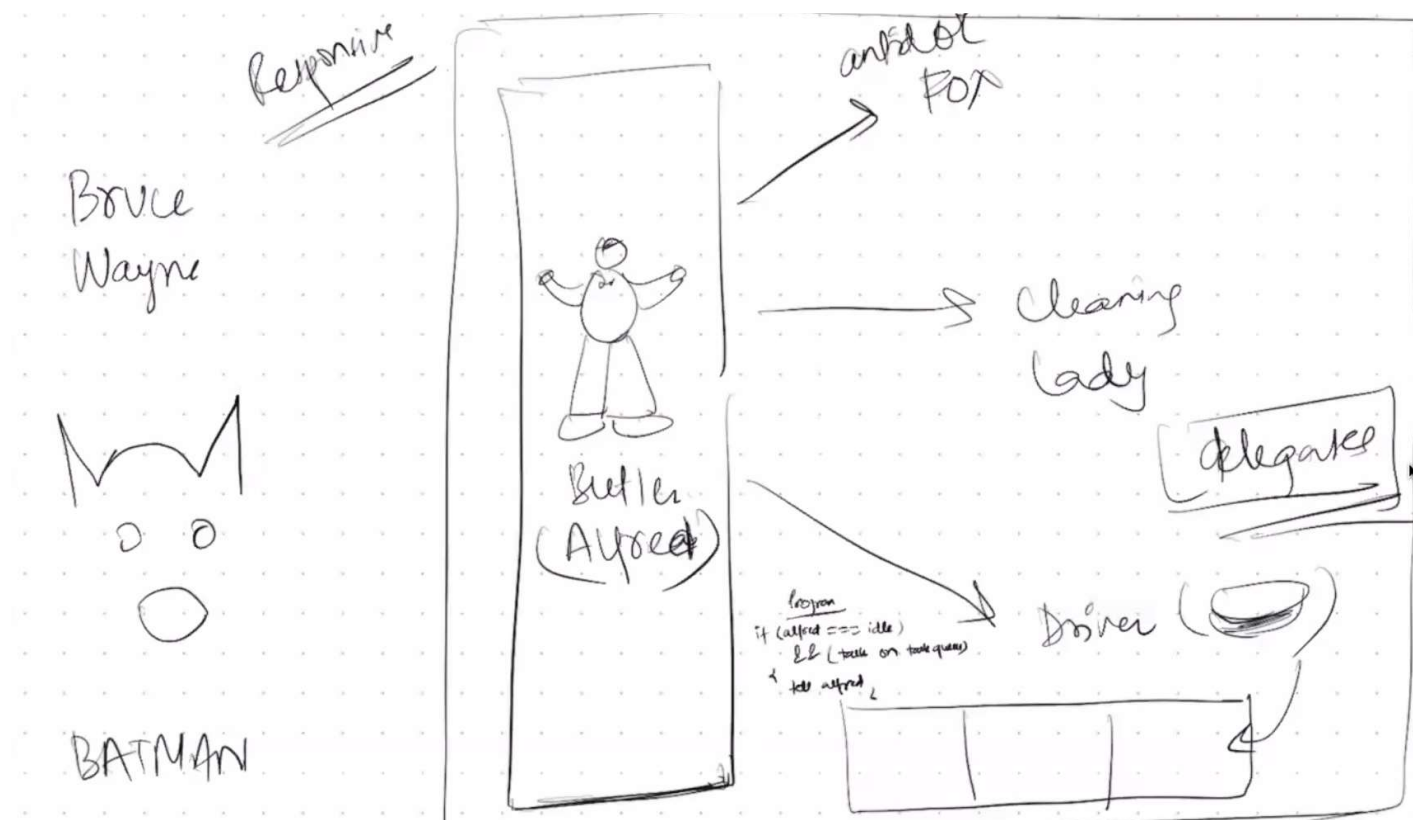
Why do we need async?

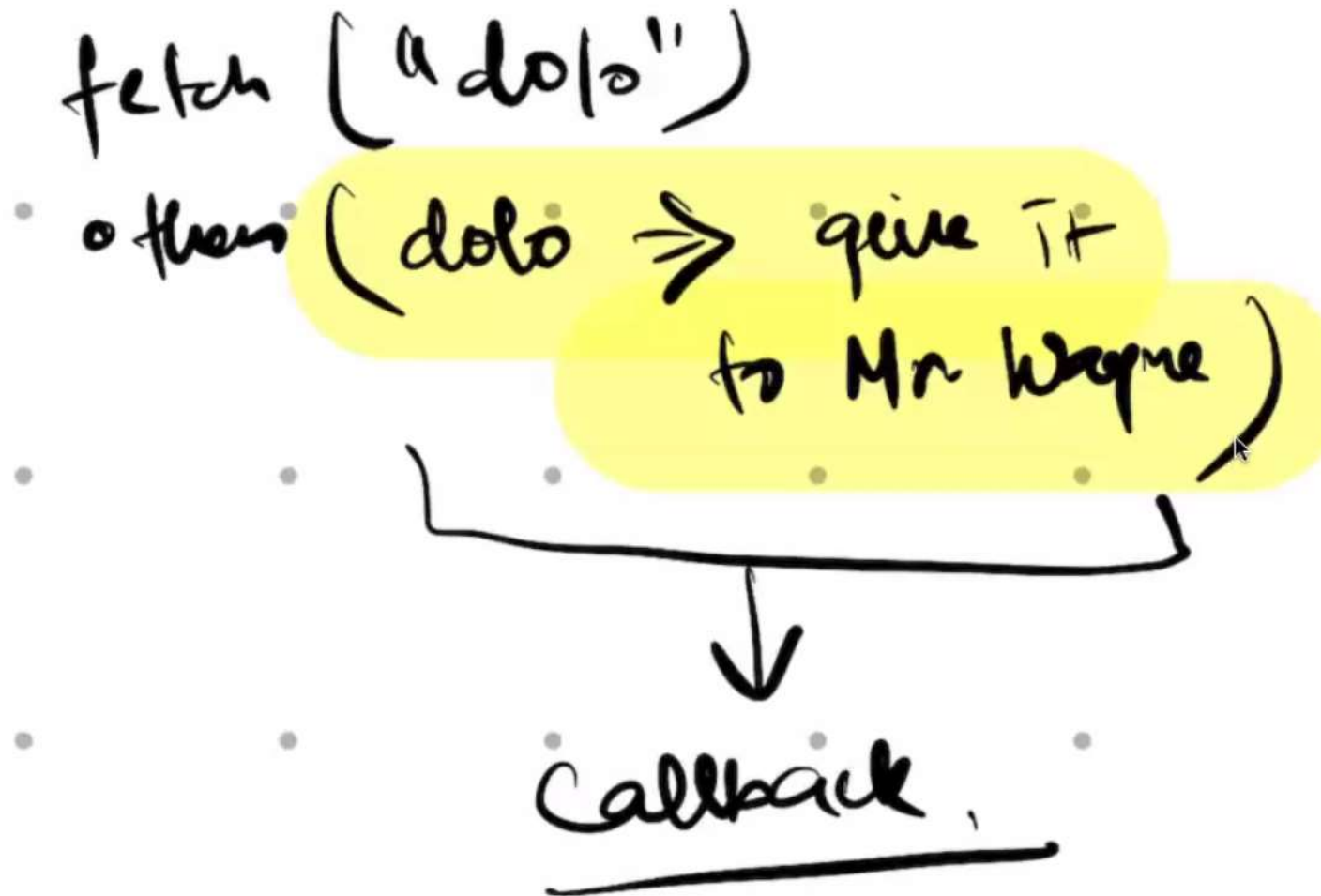
- We cannot make the user wait.
- by the very nature of browser, it has to wait for things. (example: send an email, searching in google)

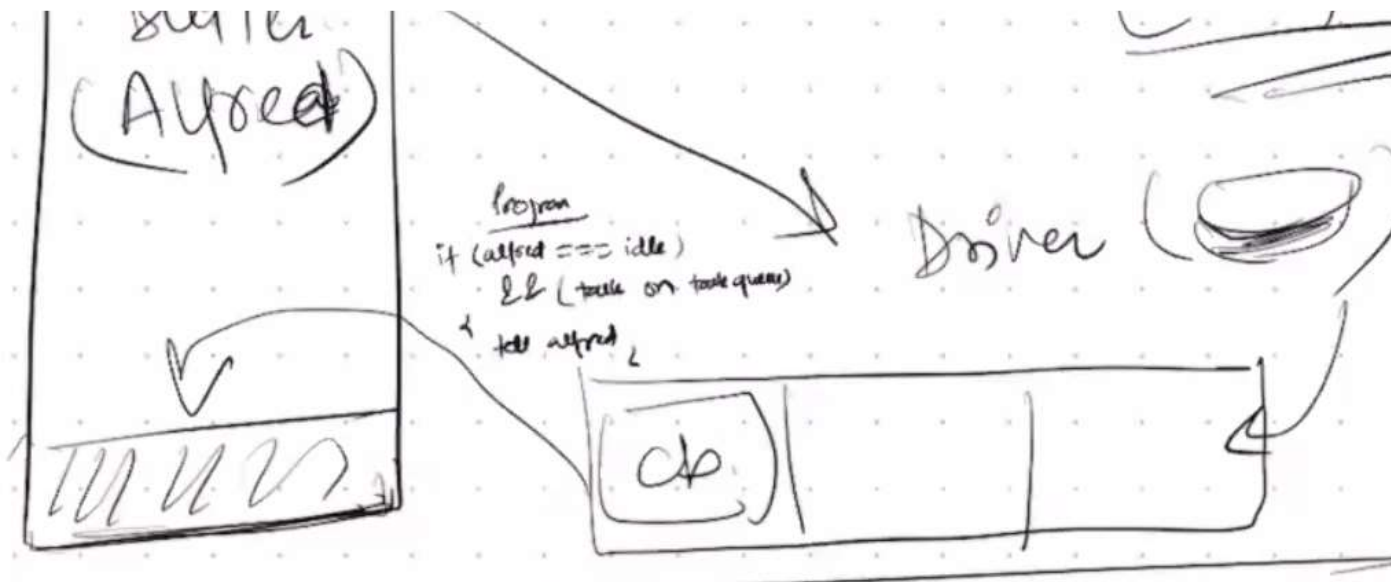
- List

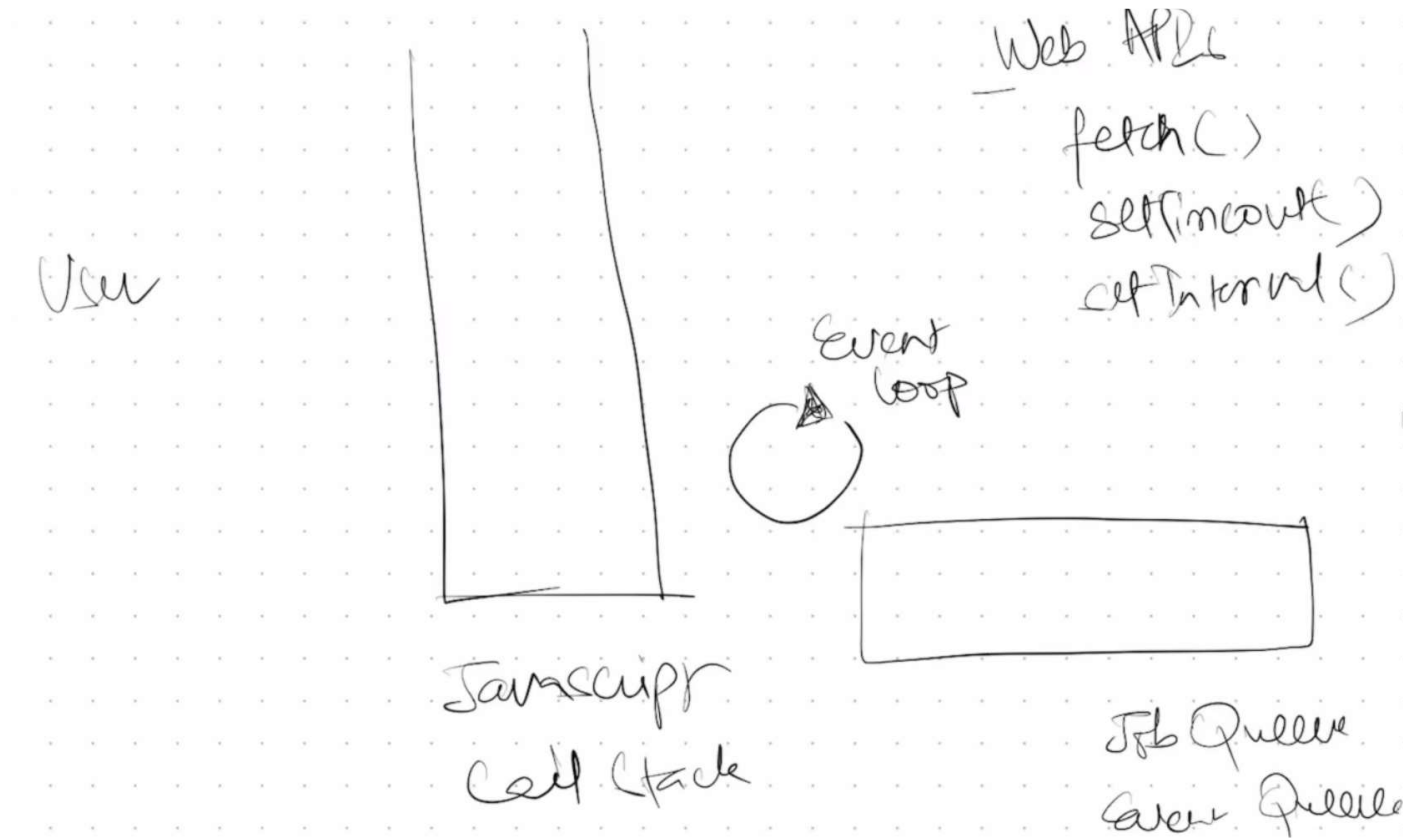
`fetch(url).then(callback)` when you are done, invoke the callback

If I am delegating my work to some people, how do I hear back?









Js call stack: running js

Web apis: once task will complete, it will put the CB in job queue then puts it in js call stack.

Examples

Ex - 1

```
fetch(url) .then(data => console.log("done"); console.log("hi");
```

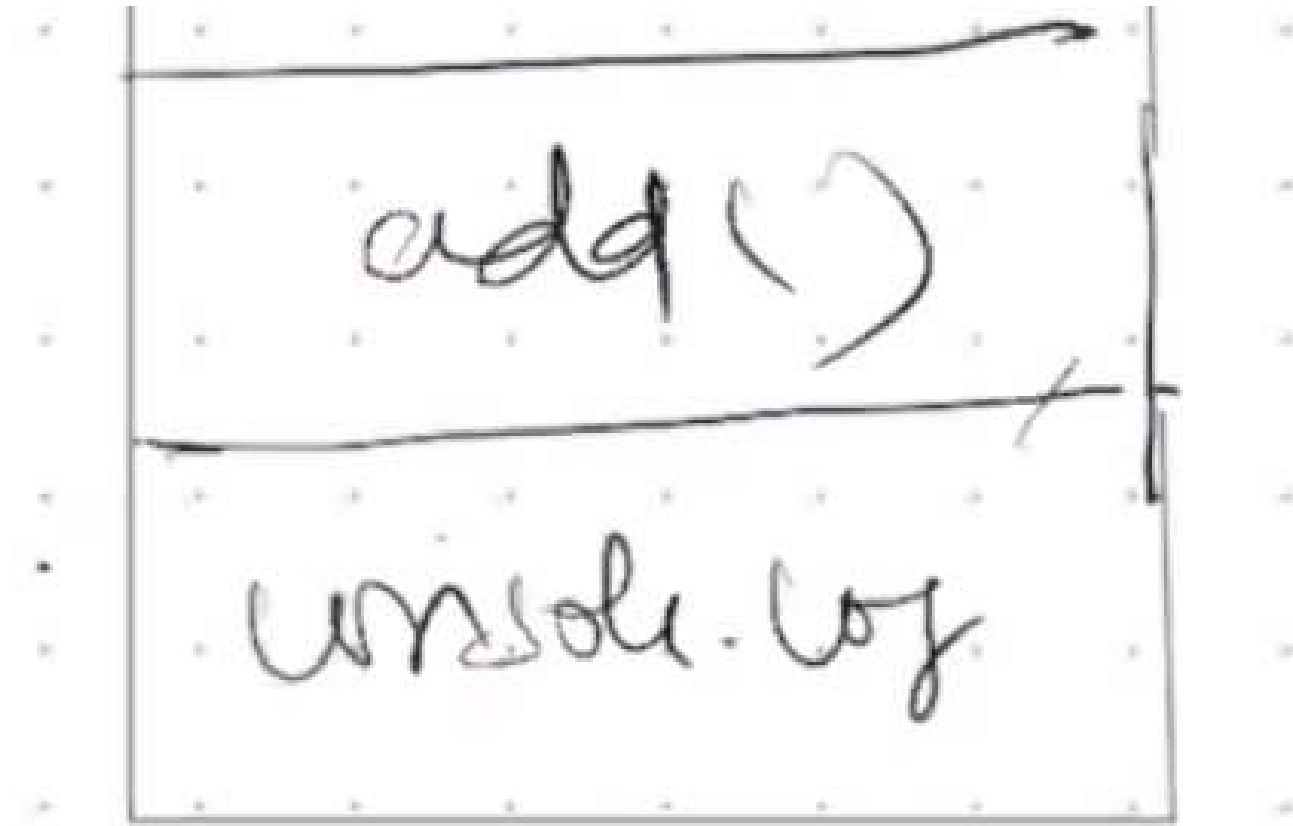
1. creates an execution context
2. browser registers the cb.
3. logs "hi".
4. call stack becomes empty
5. gets cb in event queue
6. Event loop will see call stack is empty and queue not empty
7. It will put the cb in call stack.
8. create a execution context of cb, and it logs "hello"

Ex-2

```
function add(num1, num2) { console.log(num1 + num2); } setTimeout(() =>  
console.log("hahaha"), 0); console.log(add(2,3));
```

1. creates an execution context, where in memory function add will be given space.
2. registers setTimeout CB.
3. As timer expires(0). It puts CB in event queue.

4. `console.log`, call `add(2, 3)` \Rightarrow so it will put `add(2, 3)` in call stack and creates a new execution context for function `add(2, 3)`. (like this)



5. Now it will print `2 + 3`.
6. Now from call stack exec context of `add` will be removed. Pops out `add()`.
7. Pops out `console.log()`

8. Now call stack is empty.
9. Event Loop notices call stack is empty, and puts CB in call stack.
10. Runs the CB, by creating an execution context.
11. logs the "hahahah"
12. Now call stack gets empty.

Questions

Why Event queue pushes only when Call stack is empty?

- Program will not be predictable.
- It is not good if it interrupts our program.
- If we have unpredictability, it is called a bug.

Does each tab has separate call stack, event queue?

Every tab operates in a separate VM, obviously for security. Tab is a one more instance.

We download assets as part of source along with JS, and JS hasn't executed yet so how come we initiate JS call stack to download images?

(need to watch recording, I missed something here)

- My first paint should happen as fast as possible. Start downloading images for future reference.
- You will open outlook, all emails should be seen in first 3 seconds. Start downloading header for first 20 emails.
- then content for next 20 when the call stack is empty. So how do we know that call stack is empty?
 - By using `setTimeout(0)`

Search for : `setTimeout(0)`, performance optimisation and lazy loading on web.dev,

Tips

- Should not do heavy calculations on main thread. That's when sites become unresponsive.
- So where should we do heavy working?
 - On web worker.
- How can we do heavy working on web worker?
 - Explore by ourselves

Websites mentioned

1. reactmeetup39.tanaypratap.com

2. js visualiser

- a. <https://www.jsv9000.app>
- b. <http://latentflip.com/loupe/?code=JC5vbignYnV0dG9uJywgJ2NsaWNrJywgZnVuY3Rpb24gb25DbGljaygpIHsKICAgIHNIIdFRpbWVvdXQoZnVuY3Rpb24gdGltZXl0KSB7CiAgICAgICAgY29uc29sZS5sb2coJ1lvdSBjbGlja2VkIHRob250b24hJyk7ICAgIAogICAgfSwgMjAwMCK7Cn0pOwoKY29uc29sZS5sb2colkhplSlpOwoKc2V0VGltZW91dChmdW5jdGlviB0aW1lb3V0KCkgewogICAgY29uc29sZS5sb2colkNsaWNrIHRob250b24hlik7Cn0sIDUwMDApOwoKY29uc29sZS5sb2colldlbGNvbWUgdG8gbG91cGUulik7!!!PGJ1dHRvbj5DbGljayBtZSE8L2J1dHRvbj4%3D>

Homework

Read reactmeetup39.

Blog Ideas

1. Understanding about callstack, event loop, event queue.
2. multithreading vs concurrency vs async
3. setTimeout(0)
4. Where should we do heavy working? What is web worker?

