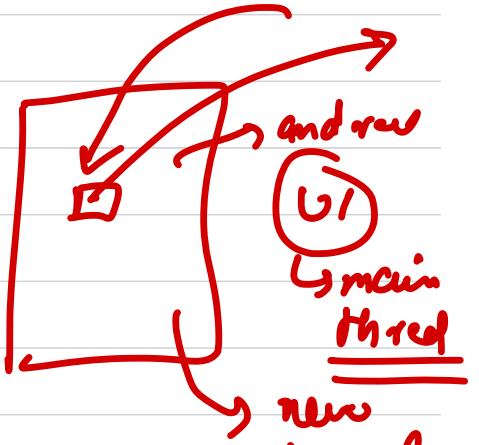
Imp features of JS

↳ JS is synchronous in nature (any piece of code understandable by JS will be executed line by line)

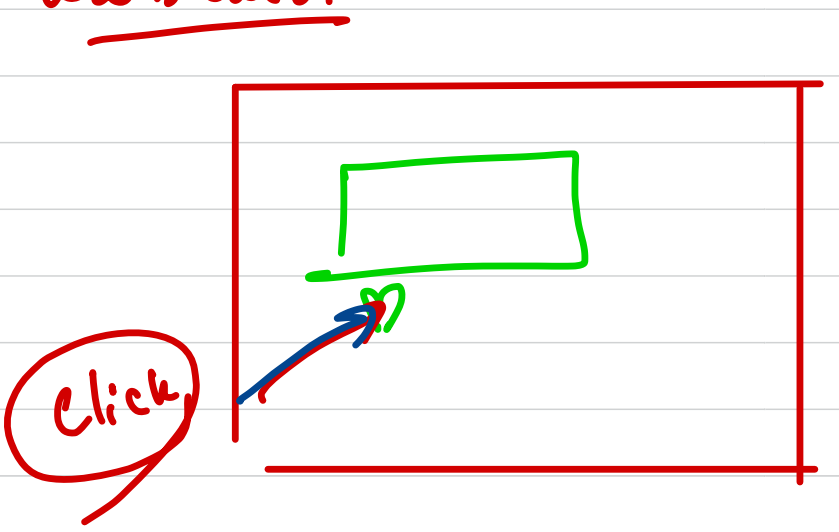
→
→
→
→



↳ JS is Single threaded.

Still gives non blocky nature
↳ for features apart from pure JS

Web browser

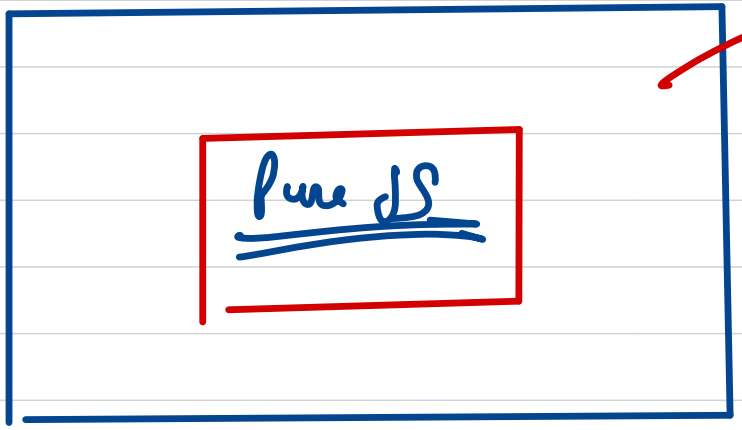


Internet slow

25s

Pure JS, doesn't know things like timers, counters, how to connect to internet, how render and control web pages etc etc

Runtime env provides extra features to JS



runtime env

engine

C++ & 2

Browser

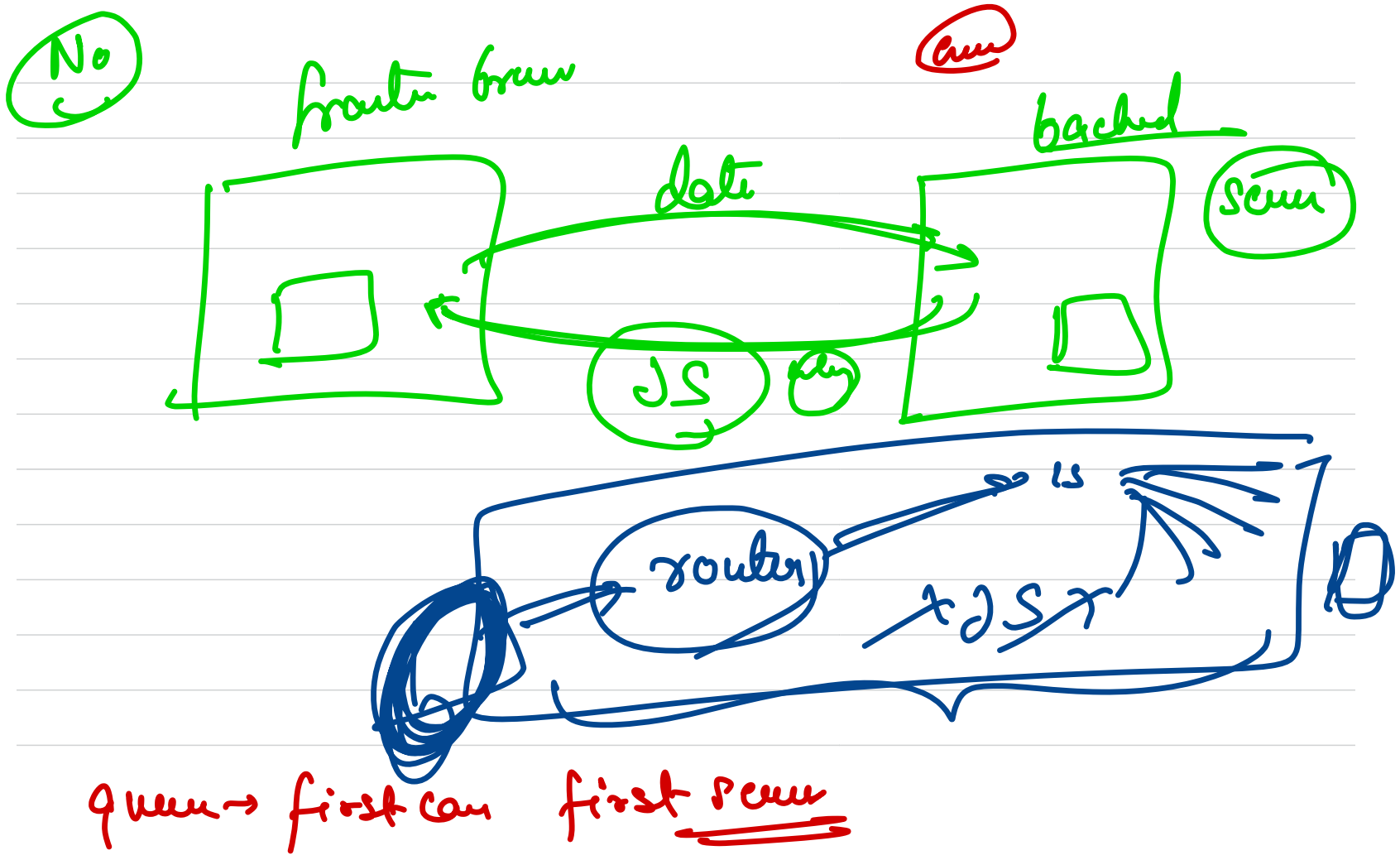
node

JS ⟷ browser

2009

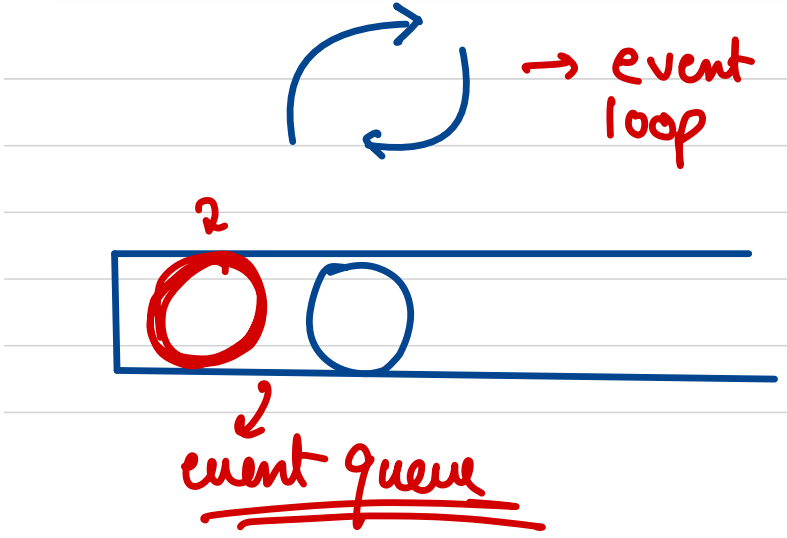
Ryan Dahl

node



queue → first come first serve

```
1   function printhello() {
2       console.log("hello");
3   }
4
5   function blockforfewsec() {
6       for(let i = 0; i < 1000000000; i++) {}
7       console.log("ending blocker");
8   }
9
10  setTimeout(printhello, 0);
11  blockforfewsec();
12  console.log("me first");
13  blockforfewsec();
```



JS will line by line execute the native code. The moment JS sees anything apart from native then it does the following -

→ JS just do one operation which is to send signal to the runtime, that a runtime feature is accessed.

→ Then it sends the function & arguments to the runtime and moves forward! It doesn't wait.

The moment runtime finishes it's func, we don't execute it then & there. We make it wait inside the event queue.

So JS will keep on execute the code & event queue will wait till the time our call stack / global code is yet to be completed.

Q) Given two strings, check if they are anagram of each other.

Ex

listen
silent

ans → true

(struck through:)
- l – 1 0
- i – 1 0
- s – 1 0
- t – 1 0
- e – 2 1 0
- n – 1 0

{ }

n → length of string

$n \le 10^6$

Both the strings should have same occurrence of char

DS → obj

key – value
char – freq

$O(n)$

$O(1)$

Matrix traversal

Quick Sort

more problem on recursion

Hashing