

19 Video

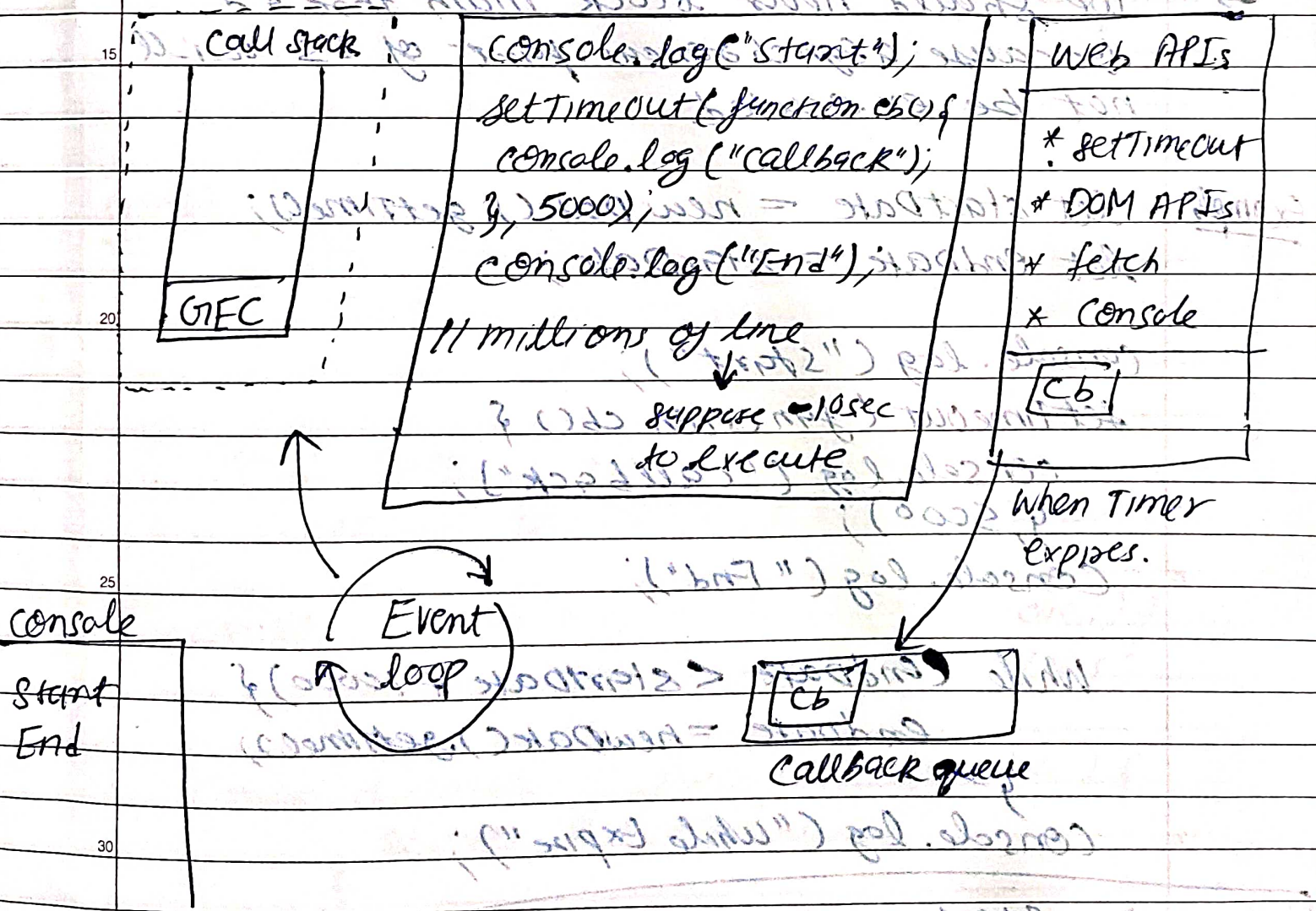
TRUST ISSUES With setTimeout()

```
function cb() {
  console.log("callback");
}
```

```
setTimeout(cb, 5000);
```

→ Does not always wait for 5 second. It may wait 6s, 8s. It completely / purely depend on call stack.

→ When call stack become empty then only methods inside microtask and callback queue get chance to executes



Concurrency Model

- Main thread / call stack become empty. Meanwhile timer will keep on running and will expire in 5sec. Then cb() have to wait for 5sec. more in callback queue. After that event loop will push it into call stack then cb() method will be executed.
- setTimeout will take 10sec instead of setting timer of 5sec.
- We should never block main thread because a synchronous part of JS will not be executed.

Example

```

let startDate = new Date().getTime();
let endDate = startDate;

console.log("Start");
setTimeout(function cb() {
  console.log("callback");
}, 5000);

console.log("End");

while (endDate < startDate + 10000) {
  endDate = new Date().getTime();
}

console.log("While Expire");
  
```

o/p	→	Start End while expires } After 10sec callback
-----	---	---