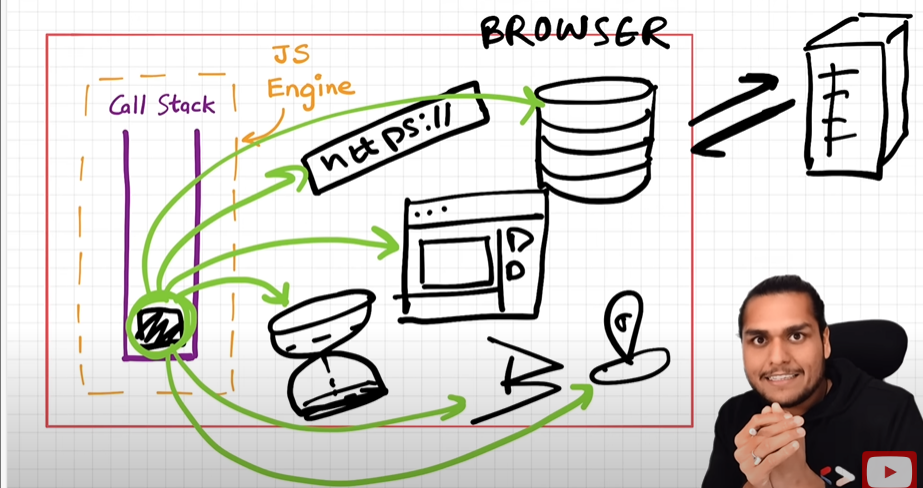
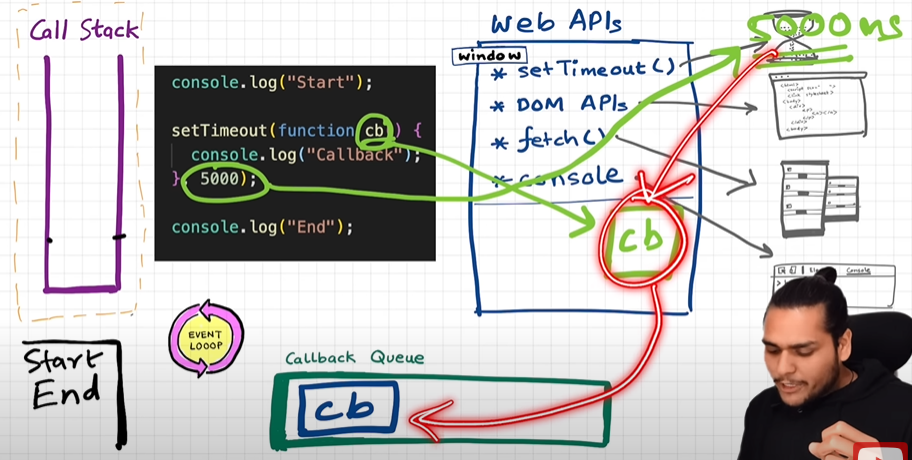
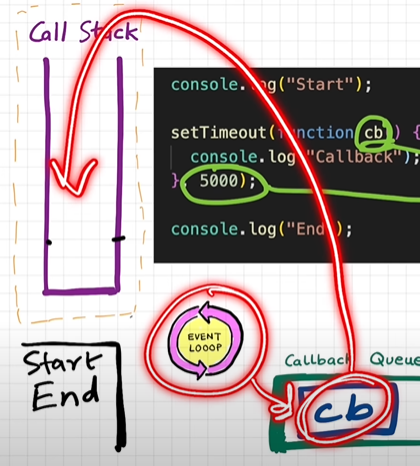
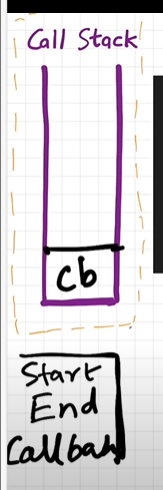
How JS Engine executes code.



Use throw global object window

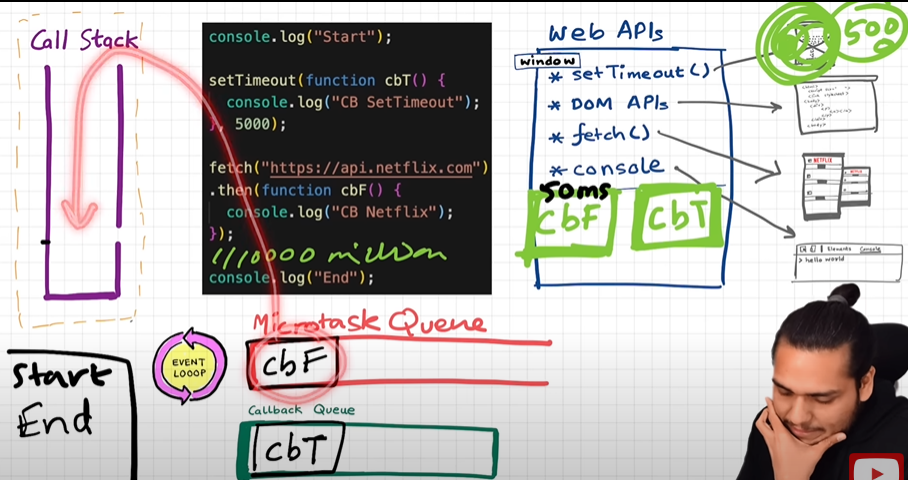




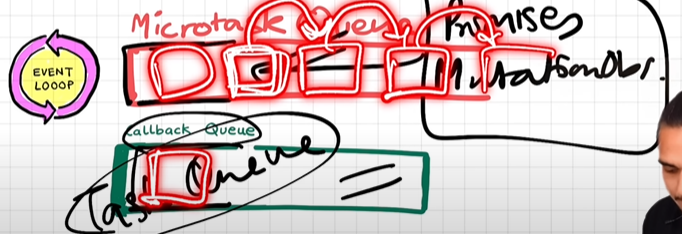
**Event loop** – continuously moniter call stack, when call stack is empty it start pushing from queues.

First the callback fun() is register inside web Apis

**Micro task Queue** is having higher priority (like Promises & MutationObjerver).

****

Starvation of Fun() in Callback Queue: never give the chance to callback queue to be executed.



Things learned:

1. Browser has superpowers that are lent to JS engine to execute some tasks, these superpowers include web API's such as console, location, DOM API, setTimeout, fetch, local storage.

2. Callback functions and event handers are first stored in Web API environment and then transferred to callback queue.

3. Promises and mutation observer are stored in API environment and then transferred to microtask queue.

4. Event loop continuously observes call stack and when it is empty it transfers task to call stack.

5. Micro task is given priority over callback tasks.

6. Too many micro tasks generated can cause Starvation (nit giving time to callback tasks to execute).