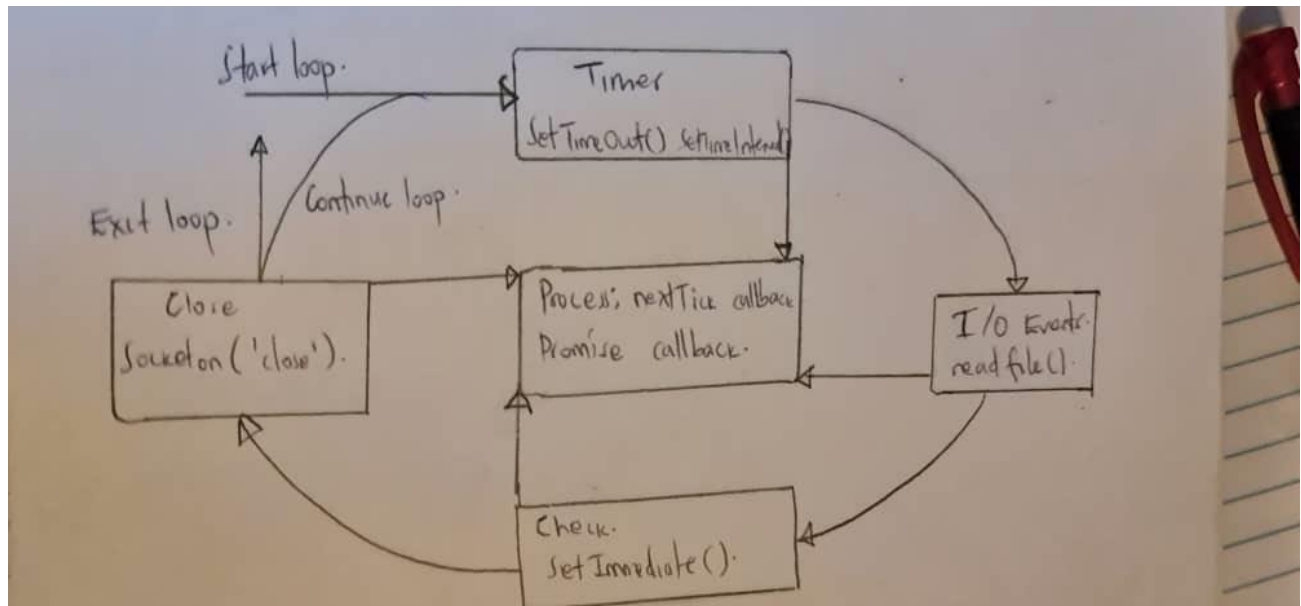# Lab10 - Node.js Intro & Module

1. There are 6 callback queues in NodeJS and they are described as below;

   1. Microtask Queue:

   The microtask queue is the highest priority queue and holds two types of callbacks:

   • process.nextTick() callbacks: These callbacks are scheduled to run in the current iteration of the event loop after all synchronous code has finished executing but before the next phase of the event loop begins.

   • Promise callbacks: When a promise resolves or rejects, its associated callback is enqueued in the microtask queue.

   2. Timer Queue:
      The timer queue holds callbacks scheduled to run after a specific delay using setTimeout () or setInterval(). These callbacks are prioritized based on their scheduled delay time.

   3. I/O Queue:
      The I/O queue holds callbacks associated with I/O operations, such as file system operations or network requests. These callbacks are executed when the corresponding I/O operation completes.

   4. Check Queue:

      The check queue holds callbacks scheduled to run using setImmediate(). These callbacks are executed after the current iteration of the event loop completes, but before the next iteration begins.

5. Close Queue:

     The close queue holds callbacks associated with the close event emitted by streams or other objects. These callbacks are executed when the corresponding object is closed.

6. Poll Phase Queue:

     The poll phase queue holds callbacks specifically related to the poll phase of the event loop, which handles low-level I/O operations. These callbacks are executed during the poll phase.

2. Output

start
end
nextTick 1
nextTick 2
Promise....1
Promise....2
timeout 1
timeout 2
nextTick 3
timeout 3

3. Output

Timeout
readFile…….
Immediate

4. Output when running Node.js: undefined

Output in browser: Hello world

This keyword is global inside browser which is window but Var is not exposed because we have a wrapper function so when we call this. Message, we do not have message in the global object and that is why it is undefined. In the browser we don't have a wrapper function that is why everything is executed in the window

5. **Output**

Code will throw an error because the variable getName is not a function. The object is being imported and stored in the getName variable, hence cannot be called a function because it ()

6. Code throws an error, because module pattern2 cannot be found.

7. **Output:** Josh Edward

Reason: In this line of code, const {getFullname} = require('./pattern2), there is destructuring assignment of an object getFullname hence an output of Josh Edward

8. {firstname: 'John', lastname: 'Smith'}