Homework \_02

1.

**setTimeout** :is simply like calling the function after delay has finished. Whenever a function is called it is not executed immediately, but queued so that it is executed after all the executing and currently queued eventhandlers finish first. setTimeout(,0) essentially means execute after all current functions in the present queue get executed. No guarantees can be made about how long it could take.

**setImmediate:** is similar in this regard except that it doesn't use queue of functions. It checks queue of I/O eventhandlers. If all I/O events in the current snapshot are processed, it executes the callback. It queues them immedieately after the last I/O handler somewhat like process.nextTick. So it is faster.

Also (setTimeout,0) will be slow because it will check the timer at least once before executing. At times it can be twice as slow.

2. The major **difference between setImmediate** and **process**.**nextTick**is that **setImmediate** queues its callbacks on the event loop while**process**.**nextTick** doesn't. ... Since **process**.**nextTick** callbacks execute before going back to the event loop, they run before**setImmediate** callbacks.

3.