1. **Explain why do we want sometimes to use setImmediate instead of using setTimeout?**

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.

SetTimeout has 4ms minimum delay while on a recent browser a chunk take ~0.2ms of processing, removing these 4ms make a huge difference.

2. **Explain the difference between process.nextTick and setImmediate?**

process.nextTick()

The callback of a process.nextTick() is placed at the head of the event queue and is completely processed before I/O or timer callbacks but still after execution of the current execution context. It is used when we need to postpone emitting an event until after the caller has had the chance to register an event listener for this event.

setImmediate()

setImmediate is similar to setInterval/setTimeOut in that it has a cancelImmediate() in the same way as cancelInterval/cancelTimeout, but it lacks a time as a second argument.

Actually in a sense it is closer to process.nextTick. The difference with nextTick is that setImmediate's callback is queued after I/O callbacks, while nextTick's callback is queued to execute before I/O callbacks.

3. **Name 10 global modules available in Node environment?**
   - Require
   - Module

- Global
- setTimeout
- setImmediate
- setInterval
- Process
- Buffer
- v8
- http