

1. What is LibUV?

- a multi-platform support library with a focus on asynchronous I/O.
- Asynchronous file and file system operations
- Thread pool

2. Explain the difference between `setImmediate(f)` and `setTimeout(f, Time)`?

- **`setTimeout(f, Time)`**: schedules a callback `f` to run after a specific `Time`, the functions are registered in the **timers phase** of the event loop.
- **`setImmediate(f)`**: schedules a callback `f` to run at **check phase** of the event loop after IO events' callbacks.
- When running `setTimeout` with a delay 0 ms and an `setImmediate` async method, the order of execution can never be guaranteed.

```
setTimeout(() => console.log("this is setTimeout..."), 0);  
setImmediate(() => { console.log('immediate'); });
```

3. Explain the difference between `process.nextTick(f)` and `setImmediate(f)`?

- **`setImmediate(f)`**: schedules a callback `f` to run at **check phase** of the event loop after IO events' callbacks.
- `process.nextTick()` is not part of the event loop, it adds the callback into the nextTick queue. Node processes **all the callbacks** in the nextTick queue after the current operation completes and before the event loop continues

```
setImmediate(() => { console.log('immediate'); });
```

```
process.nextTick(() => console.log('nexttick'));
```

the result will be:

nexttick

immediate