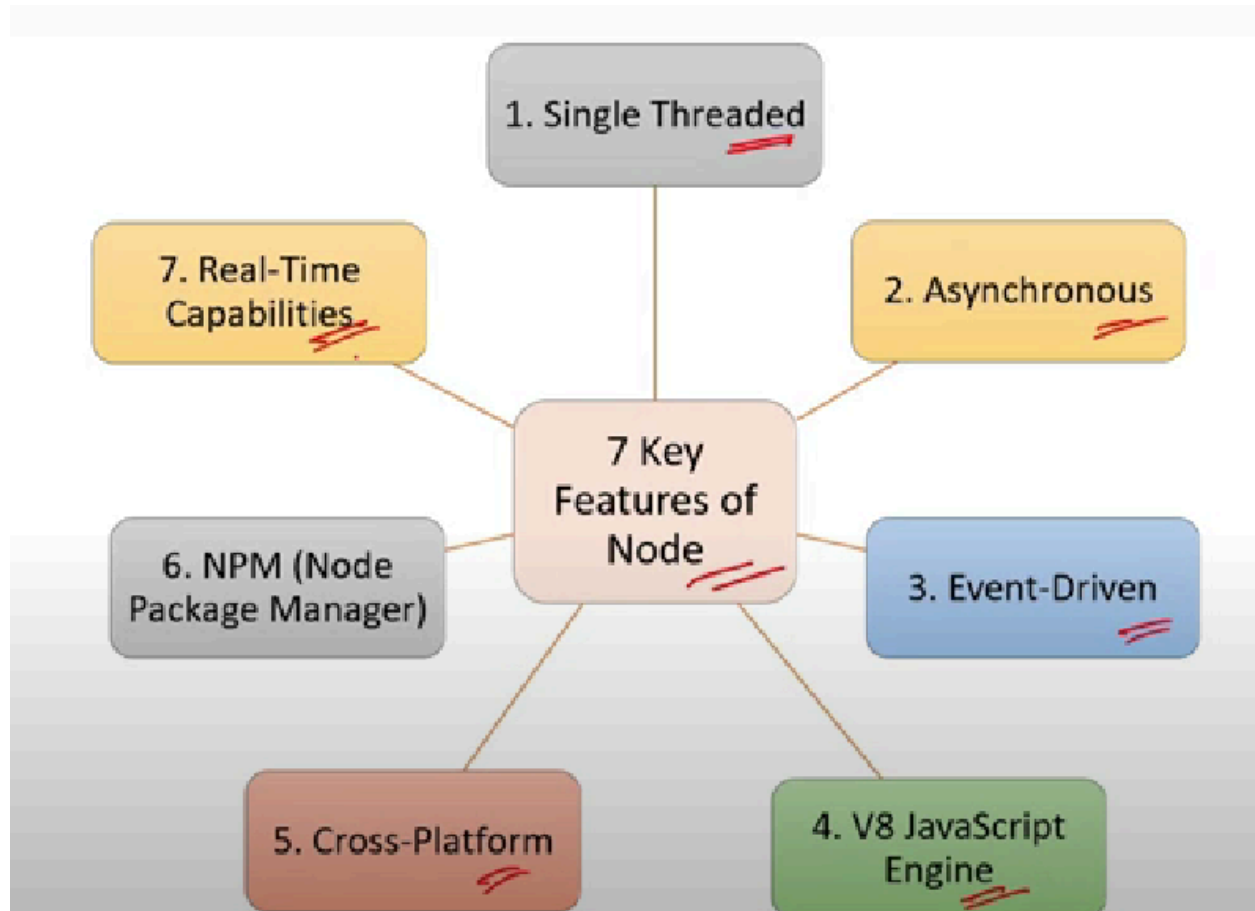


Question : Why node js is best choice to for web technologies already we have technologies like python or java ,, etc or what are the 7 main important features of node js ?

Ans:

Why it is best is because of its features only.



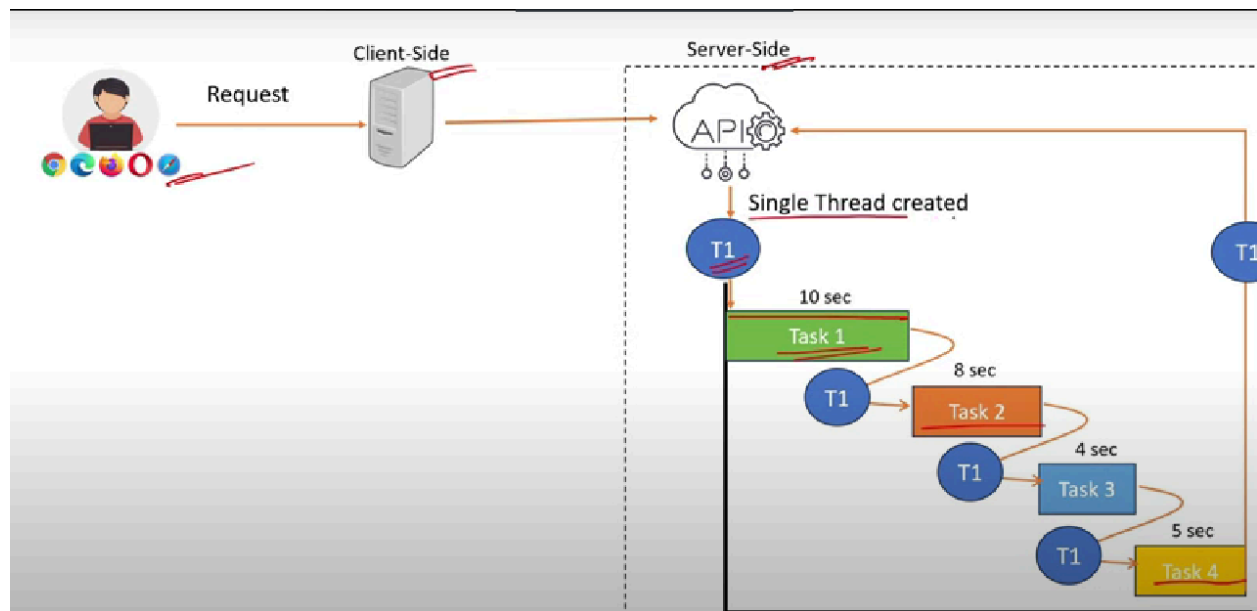
Question: what is single threaded programming ?

Ans:

The below image represents the flow execution of a program in a single thread. In the image the client sends a request to the server. the request came to the server and performed the task. Suppose for example it has to solve 5 independent sub tasks inside it (means → request). Then what happened here is in a single threaded programming a single thread will be created like in the image .

Then the thread will complete task(T1) and then T2 and T3 ,, like that

Note: it is not executes the T2 until the completion of T1 that means it performs the tasks synchronously



Single thread: the process of performing (or executing the code synchronously) the tasks synchronously is called single threaded programming or synchronous programming.

In the above diagram we can clearly observe that one task is depending on the other. (after completion T1 only performs T2. similarly completion of T2 only performs T3.)

Note: after completion of all the tasks the thread will inform the (means sends the response back to the API)

Note: the above picture is an example for single threaded synchronous programming

★ Node.js is a single threaded programming language. Even it is a single threaded programming language we can achieve the Asynchronous programming (asynchronous means executing the program asynchronously)

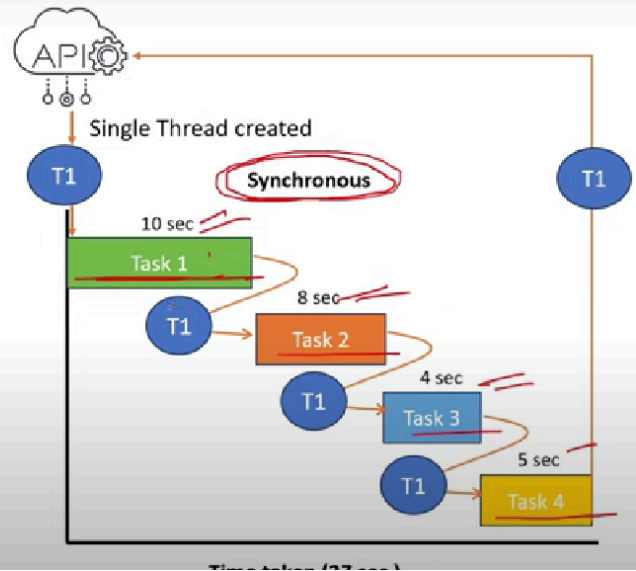
Question : what is synchronous programming ?

Ans: In which each task should be performed one after the other

The **disadvantages** of this programming is performance because it is performing the task one after the other right. So, the performance is slow.

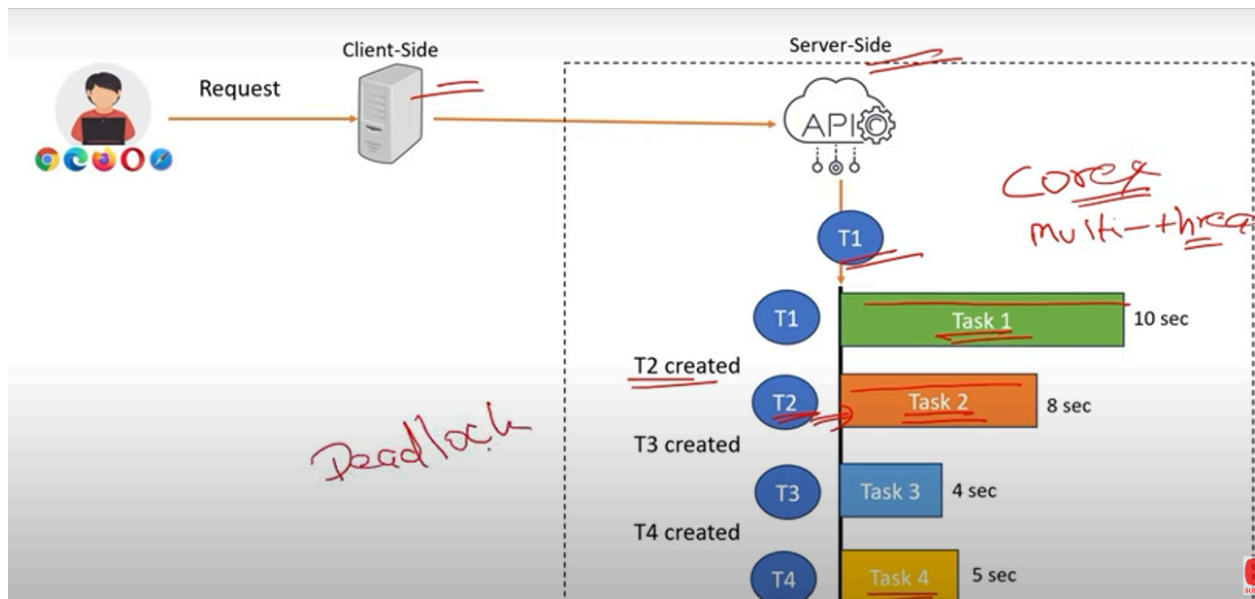
This approach takes more time and is a blocking approach (means → we can absorb in the diagram until task1 completed the task2 has to wait → task1 is blocker for task2)

- ❖ In a synchronous program, **each task is performed one after the other**, and the program waits for each operation to complete before moving on to the next one.



Question : what is multithreaded programming ?

Ans:



In the diagram when the client sends a request to the server. Suppose the request has to perform 4 tasks (observe in the diagram). To perform the task1 the server creates the thread then this thread executes the task1. In multithreading when one thread is performing the task then it sees the other task then immediately thread T2 will be created to perform the task 2.(see in the picture). This T2 starts the task2

While T1 thread waits to complete the tasks1 then T2 thread will be created and starts the tasks2.

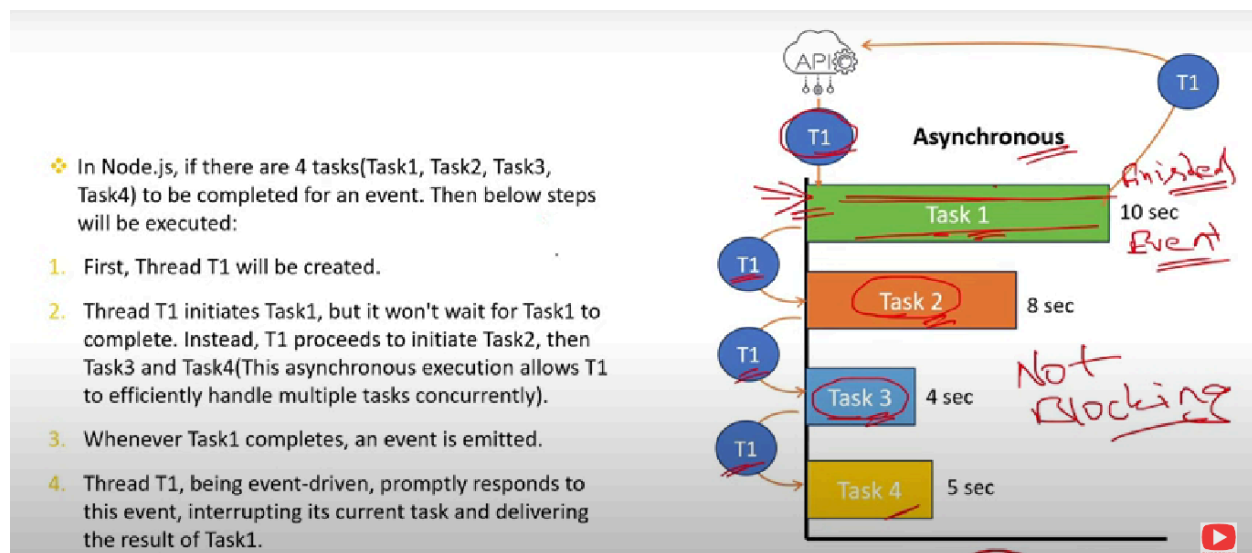
Similarly T3 thread will be created and starts the task3 and T4 also.

We can clearly observe in the picture different threads performing the different tasks .

This multithreading approach is very good for CPU intensive tasks or heavy desktop tasks .

But internet api applications multithreading thread could be a problem because multiple threads can create deadlock problems if not properly handled.

Question: what is asynchronous programming ?



- The above diagram shows the flow of the asynchronous programming
- Suppose client send request it has to perform 4 task .like above
- In asynchronous flow the single thread T1 will create and it will initiate the task1. Unlike synchronous flow the thread T1 will not wait for the completion of the task1.
- Not wait for the task1 completion the thread T1 immediately jumps to initiate the task 2 or starts the task2
- Similarly the T1 will not wait for the task2 completion. It immediately jumps to initiate the task3 or starts the task3 (we can observe it in the diagram) .and similarly task4 .

- ★ It is non blocking .way (the asynchronous programming)
- ★ Node js is a **asynchronous** and also **event driven**

When task1 executes independently then the thread T1 is busy initiating the other tasks .so, how will the system know when the task1 is completed or finished ?





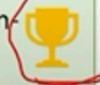


Ans: here comes the role of events in node

Whenever any task is completed or finished it will raise an event then that event is heard by the thread. At that movement the thread will stop whatever it is doing and immediately report the task1 completion to source. After reporting the thread will resume its works

In the picture when task1 is completed then it emits an event. Then this event is heard by the thread T1 and will stop whatever work it is doing.now the T1 reports the task1 completion information to the source .after reporting then T1 restarts its work. **Because this node js is also called event driven**

Question: what is the difference between the synchronous and asynchronous programming

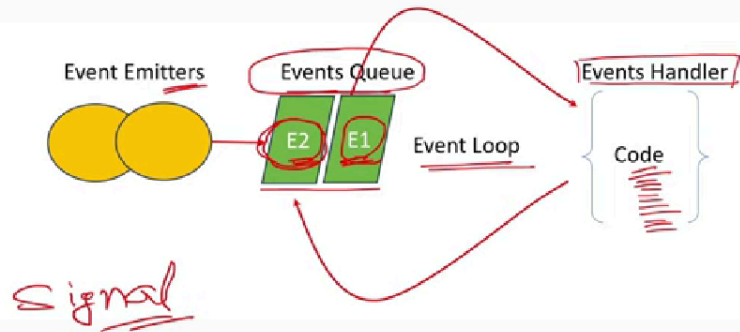
?

Synchronous programming	Asynchronous programming
1. In synchronous programming, tasks are executed one after another in a sequential manner . 	In synchronous programming, tasks can start, run, and complete in parallel
2. Each task must complete before the program moves on to the next task. 	Tasks can be executed independently of each other. 
3. Execution of code is blocked until a task is finished. 	Asynchronous operations are typically non-blocking. 
4. Synchronous operations can lead to blocking and unresponsiveness. 	It enables better concurrency and responsiveness. 

Question: What are Events ,Event Emitter, Event Queue, Event Loop and Event Driven

?

- ❖ **Event:** Signals that something has happened in a program.
- ❖ **Event Emitter:** Create or emit events.
- ❖ **Event Queue:** Events emitted queued(stored) in event queue.
- ❖ **Event Handler(Event Listener):** Function that responds to specific events
- ❖ **Event Loop:** The event loop picks up event from the event queue and executes them in the order they were added.
- ❖ **Event Driven Architecture:** It means operations in Node are drive or based by events.



Question: what are the main features and advantages of a Nodejs ?

Features	Advantages
1. Asynchronous	Enables handling <u>multiple concurrent requests</u> & <u>non blocking execution</u> of thread.
2. V8 JS Engine	Built on the <u>V8 JS engine</u> from Google Chrome, Node.js <u>executes code fast</u> .
3. Event-Driven Architecture	<u>Efficient handling events</u> . Great for real time applications like chat applications, gaming applications(using web sockets) where <u>bidirectional communication</u> is required.
4. Cross-Platform	Supports deployment on various operating systems, <u>enhancing flexibility</u> .
5. JavaScript	Coding in JS language therefore <u>no need to learn a new language</u> .
	Suitable for building <u>scalable applications</u> that can handle increased loads.

Question: what are the disadvantages of node js ? When to use ? When to not use ?

When to Use Node.js?

- ❖ Ideal for **real-time applications** like chat applications, online gaming, and collaborative tools due to its event-driven architecture.
- ❖ Excellent for building **lightweight and scalable RESTful APIs** that handle a large number of concurrent connections.
- ❖ Well-suited for building **microservices-based architectures**, enabling modular and scalable systems.

When not to use Node.js(disadvantages):

- ❖ **CPU-Intensive Tasks:** Avoid for applications that involve heavy CPU processing(Image/Video Processing, Data Encryption/Decryption,) as Node.js may not provide optimal performance in such scenarios because it single threaded and for heavy computation multi-threaded is better.