NodeJS

- Open-Source
- Cross-platform (mobile, pc – windows/Linux)
- **Runtime Environment**
- Used in server-side development
- Based on JS framework but not a framework
- I/O intensive web applications

Architecture

- Event-driven architecture and a non-blocking I/O API

Special Features

- Asynchronous and event driven
- Very Fast
- Handling of concurrent requests
- No Buffering – (outputs data in chunks)

Advantages

- Great for prototyping and Agile
- Superfast
- High Scalability
- Cleaner consistent codebase
- Large open-source library

It's not

- A programming language
- A framework

It's a runtime environment for executing JS code to develop server side applications

Usage

To Use

- I/O applications – chat apps
- Data Streaming apps
- Data Intensive Real-time Applications (DIRT)
- Ad Servers

Not To Use

- CPU intensive – since its single threaded

Regular vs NodeJS

Other client server architecture applications are stateless

Blocking Code

Read file from Filesystem, set equal to "contents"

Print contents

Do Something else

```
// Blocking
const fs = require('fs');
const data = fs.readFileSync('/file.md'); // blocks here until file is read
console.log(data);
moreWork(); // will run after console.log
```

Non-Blocking Code

Read file from Filesystem

Whenever you're complete, print the contents

Do Something else

[This is a Callback]

```
// Non-blocking
const fs = require('fs');
fs.readFile('/file.md', (err, data) => {
  if (err) throw err;
  console.log(data);
});
moreWork(); // will run before console.log
```

Apache vs NGIX

- Apache uses one thread per connection
- NGINX uses event loop instead of using threads

Why Event Looping is not Adapted Often

Missing Infrastructure

- Some libraries don't support it
- Anonymous functions in C makes call back difficult
- Database lib doesn't support asynchronous queries

Too much infrastructure

Even though very good event loop platforms exist like Event-Machine, Twisted, AnyEvent

- Users are confused how to combine with other libraries
- Need of expert knowledge on event loops and non-blocking I/O