NODE.JS

EXPRESS.JS + MONGODB

Karan Goel

12.17.2014

HI, I'M KARAN.

- CS, UW
- Developer
- goel.io
- @KaranGoel
- Java -> Python -> JS -> Dart

- Node.js
- Express.js
- MongoDB
- Live coding a web app

SO, NODE.JS?

- It's just Javascript.
- Server side JS on Google's V8
- Single threaded
- Everything is asynchronous
- All I/O is non-blocking

USES

- REST API's
- Filesystem IO
- Data streaming
- Heavy computations
- General scripting

WHY NODE.JS

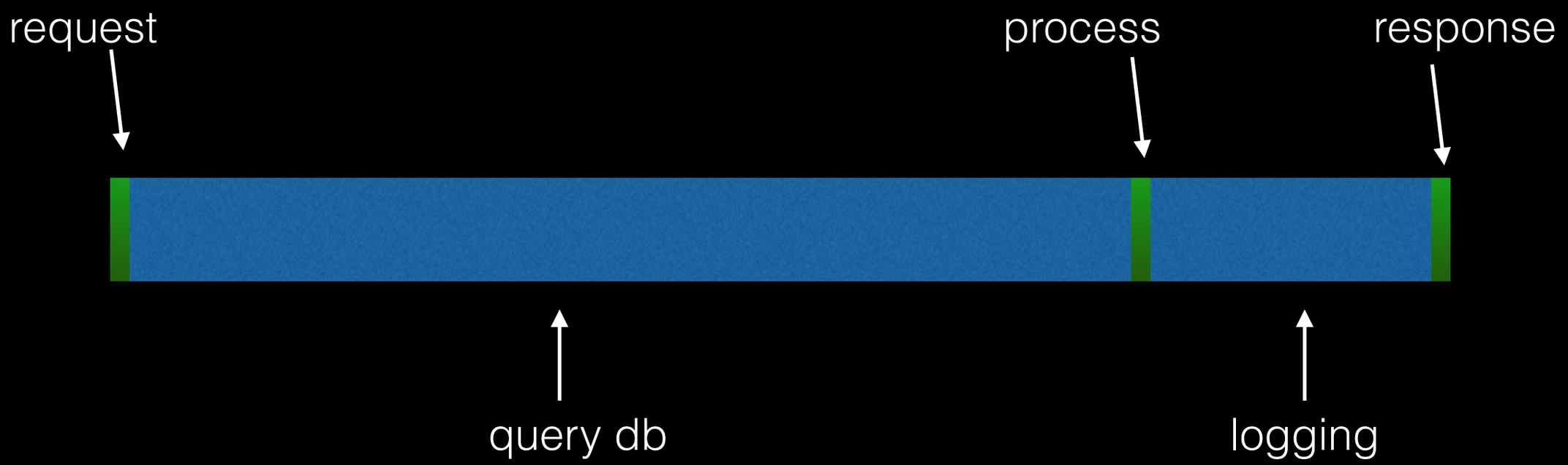
L1	3 cycles
L2	14 cycles
RAM	250 cycles
Disk	41,000,000 cycles
Network	240,000,000 cycles

Source: Ryan Dahl's, 2008

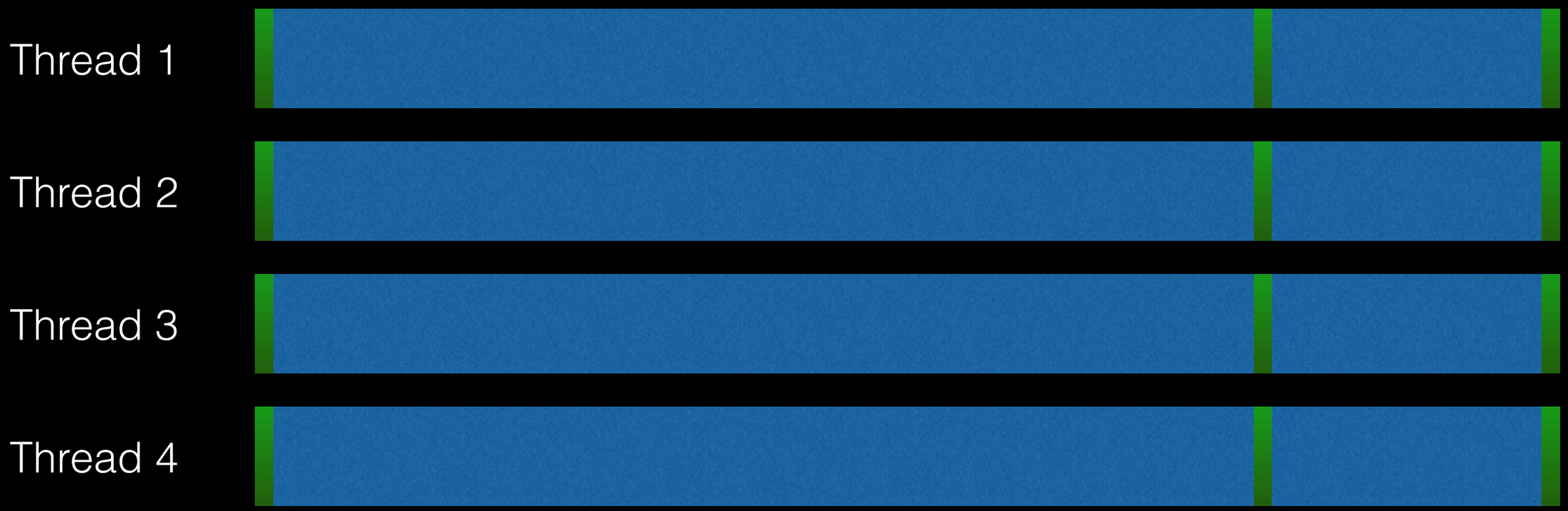
tldr

IO on disk takes much longer than processor

Waiting…

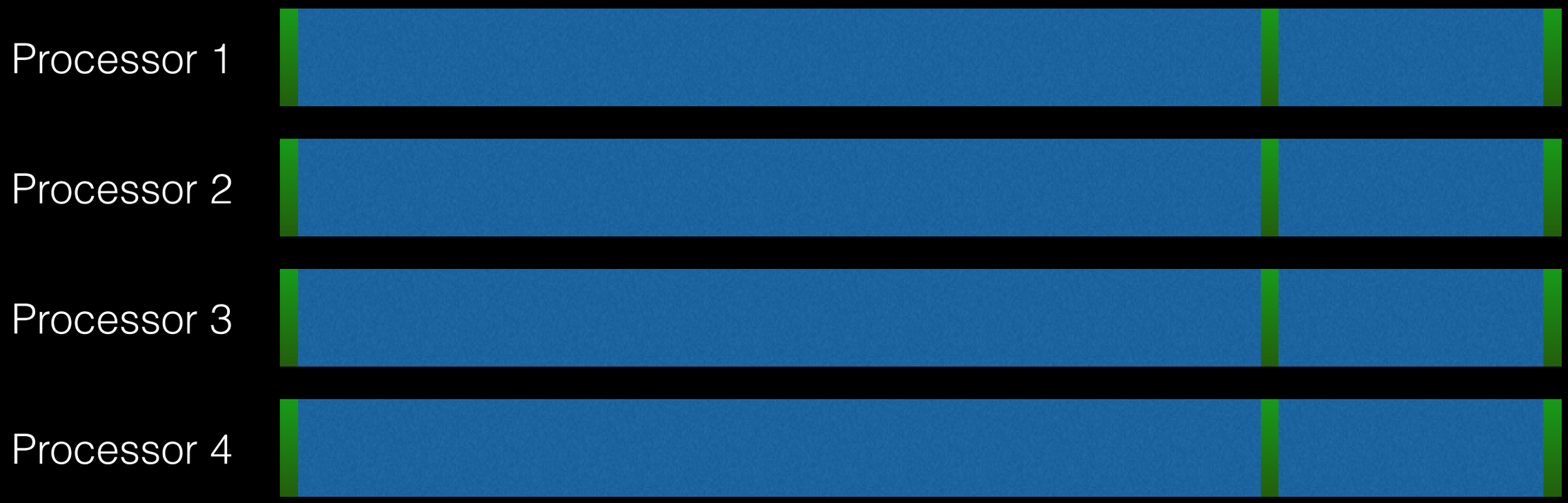


Solution 1: Threads



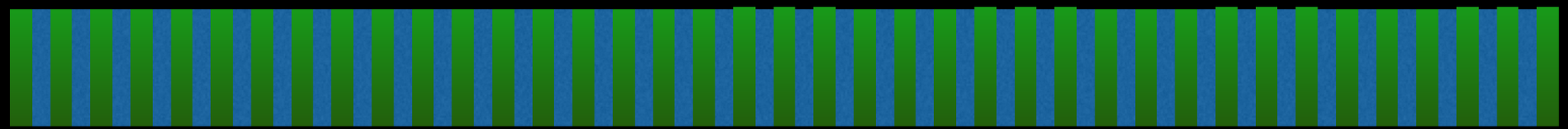
Context-switching overheads
Lots of memory
Complicated logic

Solution 2: Processors



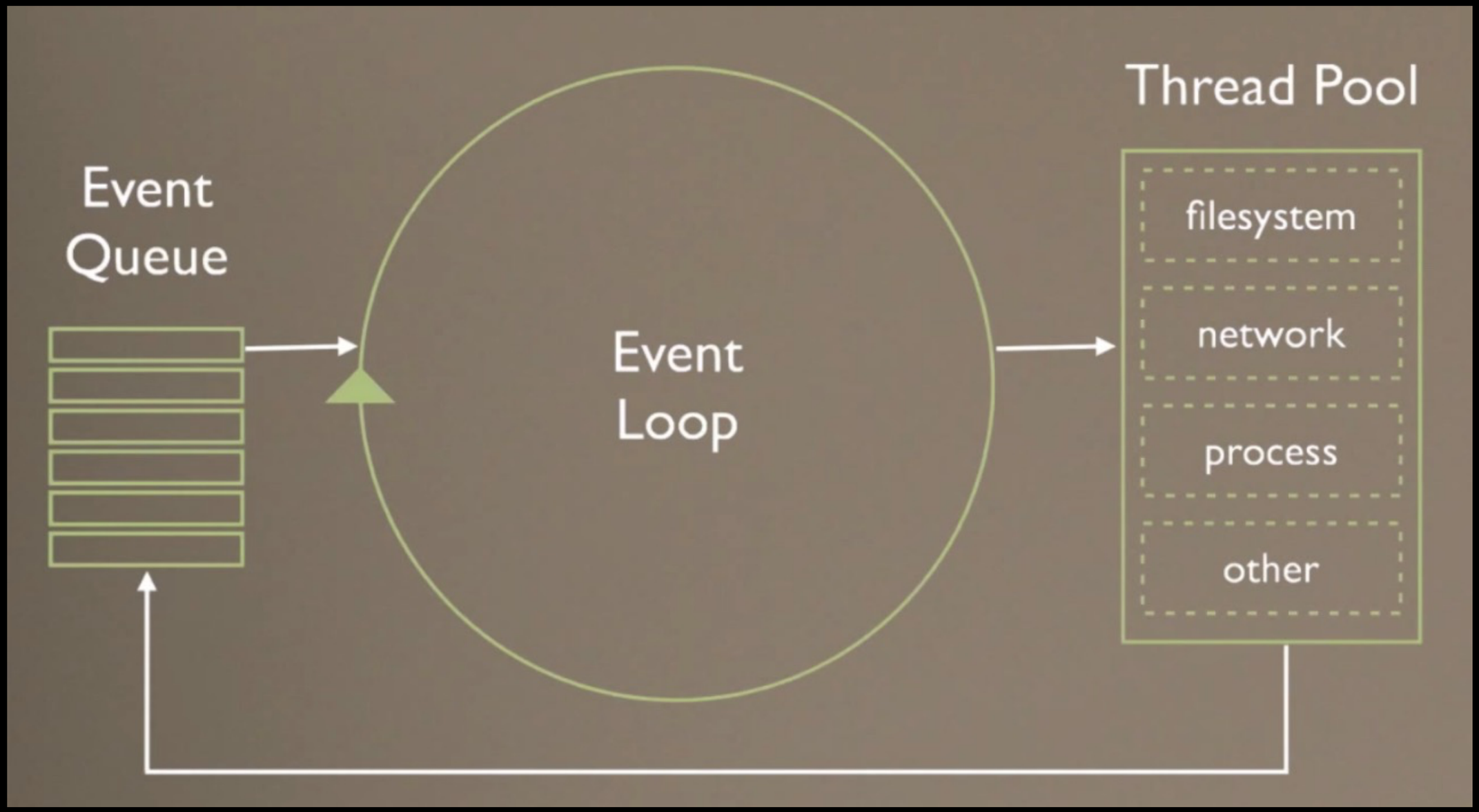
Scheduling overheads
Memory (even more)

Solution 3: Event Loop



Handle other requests and delegate IO

EVENT LOOP



Source: Jeff Kunkle

SYNCHRONOUS CODE

```
var data = readFromDatabase();
print(data);
doSomethingUnrelated();
```

Everything is blocked
CPU cycles wasted

ASYNCHRONOUS CODE

```
readFromDatabase(function(data) {
    print(data);
)};
doSomethingUnrelated();
```

doSomethingUnrelated called immediately

JS IN TERMINAL

```
karan:~$ node
> 1 + 1
2
> "1" == 1
true
> "1" === 1
false
```

HTTP Server

```
cd source/1/
node server.js
```

EXPRESSJS

- Web application framework
- Inspired by Sinatra (ruby)
- High-level API access to Node
- *npm install -g express*

HELLO WORLD IN EXPRESSJS

```
var express = require('express')
var app = express()

app.get('/', function (req, res) {
  res.send('Hello World!')
})

var server = app.listen(3000, function () {
  var host = server.address().address
  var port = server.address().port
  console.log('Example app listening at http://%s:%s', host, port)
});
```

MIDDLEWARE

- Middleware is basically a function which accepts request and response objects and a next function.

MONGODB

- NoSQL database
- Databases = Databases,
  Collections = Tables,
  Documents = Rows

A DOCUMENT

```
{
    _id: 1234,
    author: { name: "Bob Davis", email : "bob@bob.com" },
    date: { $date: "2010-07-12 13:23UTC" },
    location: [ -121.2322, 42.1223222 ],
    comments: [
        { user: "jgs32@hotmail.com",
          upVotes: 22,
          downVotes: 14,
          text: "Great point! I agree" },
        { user: "holly.davidson@gmail.com",
          upVotes: 421,
          downVotes: 22,
          text: "You are a moron" }
    ]
}
```

MONGOOSE

- Mongoose extends native drivers to Node.js
- Define your own schema (models)
- *npm install mongoose*

Let’s Make Something

A Blog

List all posts
View individual posts
Add a post
Commenting?

QUIRKS

- Asynchronous IO
  - Lots of nesting
  - Lots of code = Lots of places for bugs
    - Solution - github.com/caolan/async
- Debugging is a pain
  - No intuition for stack trace
  - Solution - node-inspector

BENEFITS

- Multi-platform - Windows, Mac, Source
- npm is kinda neat
- Lightweight - Single thread
- Monoglot (JS everything (Benefit?))
- Community

WHAT NEXT?

- MEAN stack - mean.io
- io.js - iojs.org
- Node is hard, but very fun
- Practice, a lot!

HOW TO PRACTICE

- Write small programs
- Write small packages
- Write small web apps
- Re-implement packages
- Read the source (?)

THAT IS ALL.

Questions?

@KaranGoel

https://github.com/karan/node-slides