

NODE

LEON NOEL

Pokemon Master.

AGENDA

- Morning Challenge
- Git + Github
- Nonblocking I/O + Event Loop
- ES6+
- Core Modules
- Lab Time

REVIEW

What do you need help with?

Repo Setup?

NONBLOCKING I/O

A program can make a request for a network resource while doing something else, and then, when the network operation is finished, a callback with run that handles the result

Disk and Network access are slow, you do not want to block business logic while you read files or send messages.

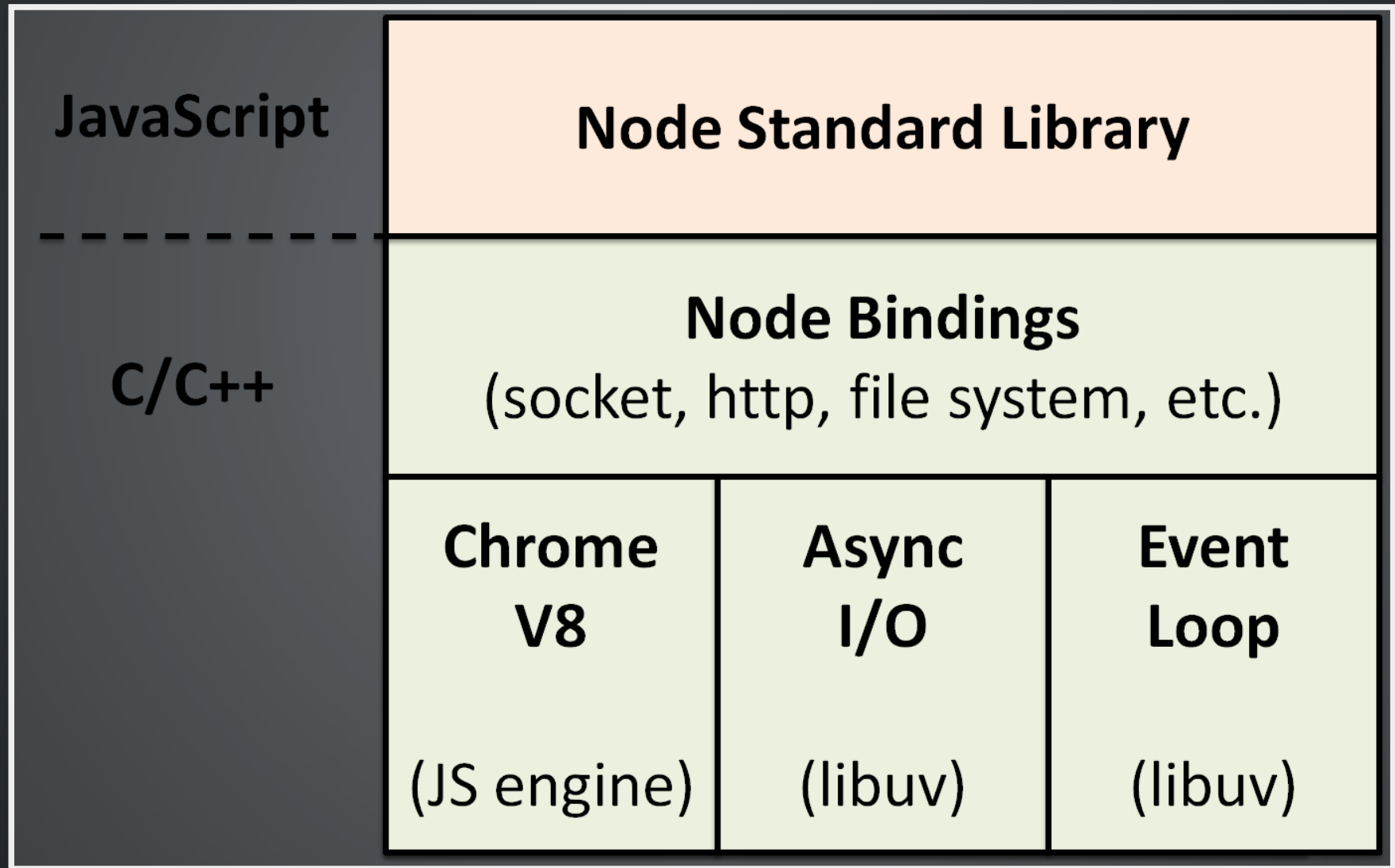
THE EVENT LOOP



EVENT LOOP

1. Browser responds to request due to Nodes HTTP server library
 2. Triggers a callback to run (added from express)
 3. Callback causes DB query to run
 4. Eventually app responds with JSON via HTTP
- 3 Nonblocking Network Calls

NODE & V8



RELEASE SCHEDULE

LTS, Current, Nightly

NPM && CORE MODULES

NPM

Package Manager / command-line tool

FILESYSTEM

```
const fs = require('fs');

fs.appendFile('mynewfile1.txt', 'Hello content!', function (err) {
  if (err) throw err;
  console.log('Saved!');
});
```

Read files Create files Update files Delete files Rename files

NETWORKING

```
const http = require('http');
const fs = require('fs');
http.createServer(function (req, res) {
  fs.readFile('demofile.html', function(err, data) {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write(data);
    res.end();
  });
}).listen(8000);
```

DEBUGGER

```
node debug hello.js
```

ES6

LET && CONST

Fix scoping issues! No longer just function or Global!

You can now have a block scope for things like if statements, for loops, and any other block!

CLASSES

Prototypal inheritance with syntactical sugar

```
function User(){  
}
```

```
class User(){  
  constructor(){  
  
  }  
  method(){  
  
  }  
}
```

TEMPLATE STRINGS

```
const greeting = "Hello"  
  
console.log("I want to say "+greeting+" to everyone!")  
  
console.log("I want to say ${greeting} to everyone!")
```

PRACTICE

Create a simple web application that uses the fs and http modules. Use http to create the server and fs to read your html file. Include vanilla ES6 js in a script tag at the bottom of your html file. Try creating a coin flip guessing game!