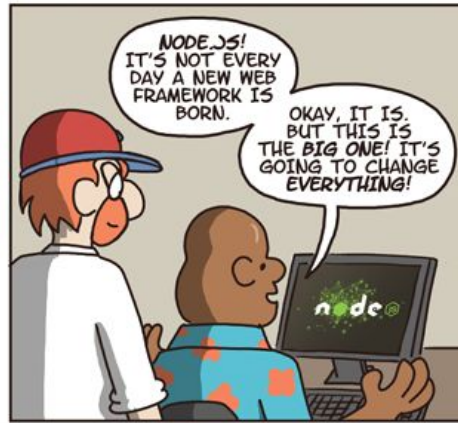




Not Invented Here™ © Bill Barnes & Paul Southworth



NotInventedHere.com

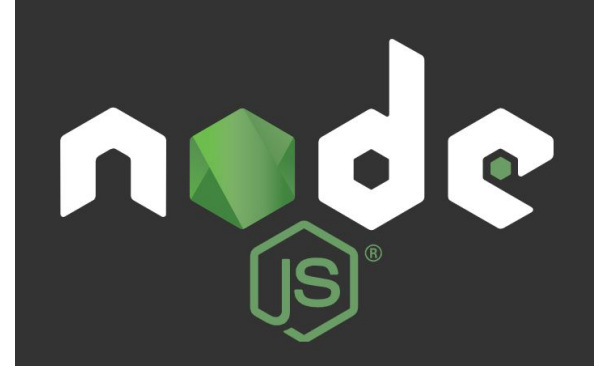
# CSc 337

## Nodejs

Benjamin Dicken

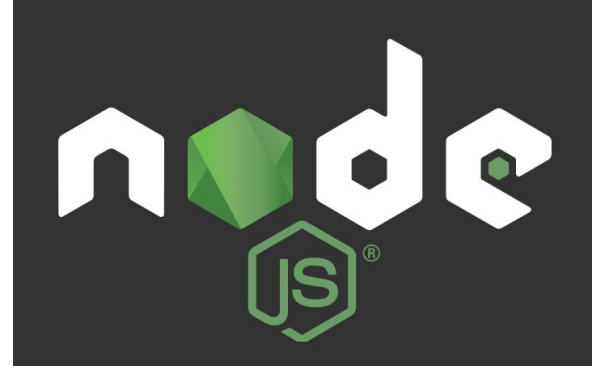
# What is Nodejs ?

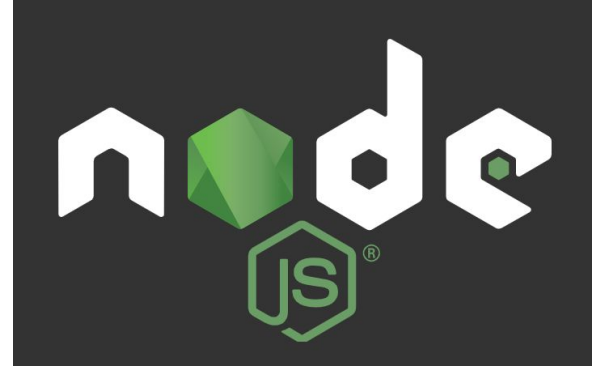
- Javascript Runtime
- Asynchronous
- Event Driven
- Can be used for
  - Local scripts/tools
  - Web servers!



# Why Node Js ?

- Same language for server and client
  - As opposed to using different language server-side, such as **php, python, ruby, java**
- Leverage callbacks, asynchronicity
- Works well as web server





Get to know this resource well:

<https://nodejs.dev/en/learn/introduction-to-nodejs/>

# Chrome V8

- The “engine” under the hood of node
- Runs javascript



```
thes = { 'fast'      : ['quick', 'agile', 'speedy'],
        'old'       : ['aged', 'antique'],
        'slow'      : ['sluggish', 'gradual'],
        'difficult' : ['hard', 'challenging', 'arduous'],
        'strong'    : ['durable', 'robust'] }
```

```
function addWord() {
  let base = prompt('Enter a base word: ');
  let newWord = prompt('Enter a new word: ');
  if (!(base in thes)) {
    thes[base] = [];
  }
  thes[base].push(newWord);
}

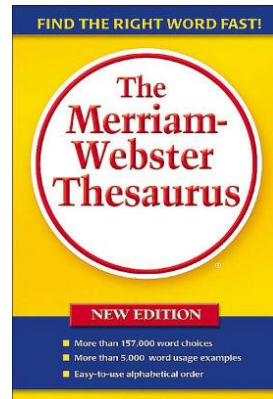
function lookUp(word) {
  if (word in thes) {
    let similar = thes[word];
    console.log('Similar words to ' + word + ':');
    for (i in similar) {
      console.log(' ' + similar[i]);
    }
  } else {
    console.log('Sorry, try another word!');
  }
}
```

# Implement with Node Instead

```
function main() {

  while (true) {
    let userInput = prompt('Enter a word or ADD: ');
    if (userInput == 'ADD') {
      addWord();
    } else if (userInput == 'EXIT') {
      console.log('stopping!');
      break;
    } else {
      lookUp(userInput);
    }
  }
}

main();
```



# Get a Line of input

```
const readline = require("readline");
const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

rl.question("Type Something: ", function(userInput) {
  console.log(userInput);
});
```

# What would this print?

```
const readline = require("readline");
const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});
console.log("W");
rl.question("X", function(userInput) {
  console.log("Y");
});
console.log("Z");
```



# Node-ified

```
const readline = require("readline");
const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

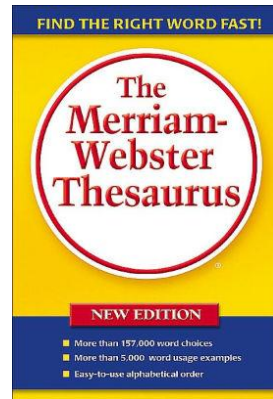
thes = { 'fast'      : ['quick', 'agile', 'speedy'],
        'old'       : ['aged', 'antique'],
        'slow'      : ['sluggish', 'gradual'],
        'difficult' : ['hard', 'challenging', 'arduous'],
        'strong'    : ['durable', 'robust'] }

function addWord() {
  rl.question("Enter a base word: ", function(base) {
    rl.question("Enter a new word: ", function(newWord) {
      if (!(base in thes)) {
        thes[base] = [];
      }
      thes[base].push(newWord);
      main();
    });
  });
}

function lookUp(word) {
  if (word in thes) {
    let similar = thes[word];
    console.log('Similar words to ' + word + ':');
    for (i in similar) {
      console.log(' ' + similar[i]);
    }
  } else {
    console.log('Sorry, try another word!');
  }
  main();
}
```

```
function main() {
  rl.question("Enter a word or ADD: ", function(userInput) {
    console.log(userInput);
    if (userInput == 'ADD') {
      addWord();
    } else if (userInput == 'EXIT') {
      console.log('stopping!');
      process.exit(0);
    } else {
      lookUp(userInput);
    }
  });
}

main();
```



# Getting a web server up-and-running

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

```
const hostname = '127.0.0.1';
```

```
const port = 3000;
```

```
const server = http.createServer((req, res) => {
```

```
  res.statusCode = 200;
```

```
  res.setHeader('Content-Type', 'text/plain');
```

```
  res.end('Hello World');
```

```
});
```

```
server.listen(port, hostname, () => {
```

```
  console.log(`Server running at http://${hostname}:${port}/`);
```

```
});
```

# How do we access the URL?

What if you want to reply differently?

```
http://localhost:3000/nfl
```

```
http://localhost:3000/nba
```

# Read from file (simple)

```
const fs = require('fs');
fs.readFile('./hi.txt', 'utf8', function (err,data) {
  if (err) { return console.log(err); }
  console.log(data);
});
```

# Write to file (simple)

```
const fs = require('fs');
const content = '<html>....</html>';
fs.writeFile('./hi.html', content, err => {
  if (err) {
    console.error(err);
  }
  // file written successfully
});
```

# Read from file (line by line)

```
async function readLines(fileName) {  
  const fileStream = fs.createReadStream(fileName);  
  const rl = readline.createInterface({  
    input: fileStream,  
    crlfDelay: Infinity });  
  for await (const line of rl) {  
    console.log(line);  
  }  
}
```

# Using line-reader module

```
const lineReader = require('line-reader');

lineReader.eachLine('/path/to/file', function(line, last) {
  console.log(line);
  if (last) { /* wrap-up code */ }
});
```

<https://stackabuse.com/reading-a-file-line-by-line-in-node-js/>

```

const readline = require("readline");
const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

thes = { 'fast'      : ['quick', 'agile', 'speedy'],
        'old'       : ['aged', 'antique'],
        'slow'      : ['sluggish', 'gradual'],
        'difficult' : ['hard', 'challenging', 'arduous'],
        'strong'    : ['durable', 'robust'] }

function addWord() {
  rl.question("Enter a base word: ", function(base) {
    rl.question("Enter a new word: ", function(newWord) {
      if (!(base in thes)) {
        thes[base] = [];
      }
      thes[base].push(newWord);
      main();
    });
  });
}

function lookUp(word) {
  if (word in thes) {
    let similar = thes[word];
    console.log('Similar words to ' + word + ':');
    for (i in similar) {
      console.log(' ' + similar[i]);
    }
  } else {
    console.log('Sorry, try another word!');
  }
  main();
}

```

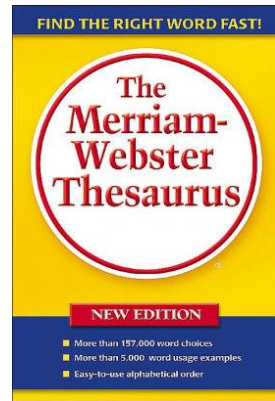
# Change to respond to requests

```

function main() {
  rl.question("Enter a word or ADD: ", function(userInput) {
    console.log(userInput);
    if (userInput == 'ADD') {
      addWord();
    } else if (userInput == 'EXIT') {
      console.log('stopping!');
    } else {
      lookUp(userInput);
    }
  });
}

main();

```





```

const readline = require("readline");
const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

thes = { 'fast'      : ['quick', 'agile', 'speedy'],
        'old'       : ['aged', 'antique'],
        'slow'      : ['sluggish', 'gradual'],
        'difficult' : ['hard', 'challenging', 'arduous'],
        'strong'    : ['durable', 'robust'] }

function addWord() {
  rl.question("Enter a base word: ", function(base) {
    rl.question("Enter a new word: ", function(newWord) {
      if (!(base in thes)) {
        thes[base] = [];
      }
      thes[base].push(newWord);
      main();
    });
  });
}

function lookUp(word) {
  if (word in thes) {
    let similar = thes[word];
    console.log('Similar words to ' + word + ':');
    for (i in similar) {
      console.log('  ' + similar[i]);
    }
  } else {
    console.log('Sorry, try another word!');
  }
  main();
}

```

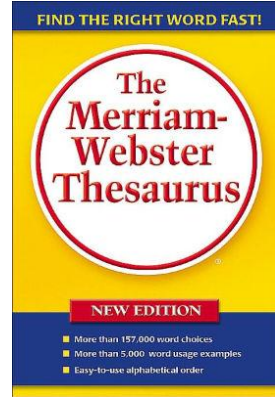
# Change to save and load to/from file

```

function main() {
  rl.question("Enter a word or ADD: ", function(userInput) {
    console.log(userInput);
    if (userInput == 'ADD') {
      addWord();
    } else if (userInput == 'EXIT') {
      console.log('stopping!');
    } else {
      lookUp(userInput);
    }
  });
}

main();

```



# Digital Ocean

- Github student developer pack
  - <https://education.github.com/pack>
- Setting up a droplet
- Connect to droplet
- Install node
- Start a live server on the droplet
- IPs and domain names
- Reply with HTML

