

Please install

+ Sign up for [GitHub.com](https://github.com)

Git

NodeJS

Sublime (or some text editor)

Building the Web with JavaScript

IEEE Student Branch @ Monash

Overview of Today (Part 1)

Outcome of Today

How does the web work?

Building the Web (tooling)

Where does it all fit in?

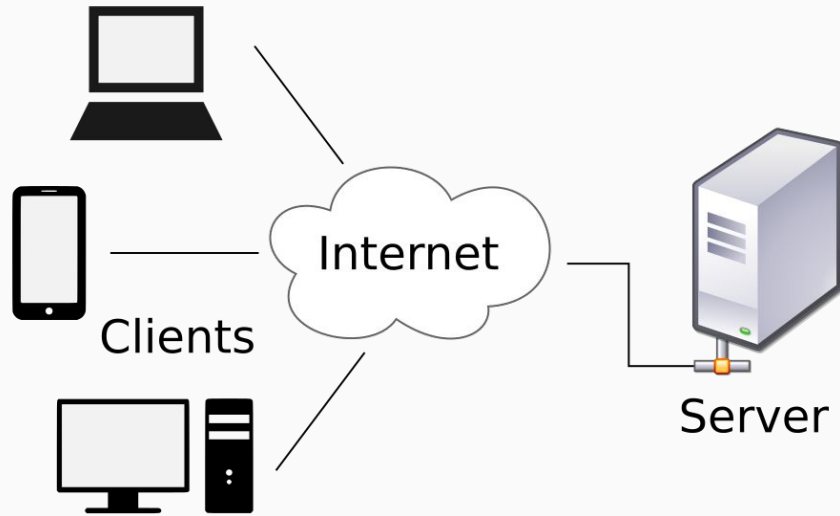
Overview of Today (Part 2)

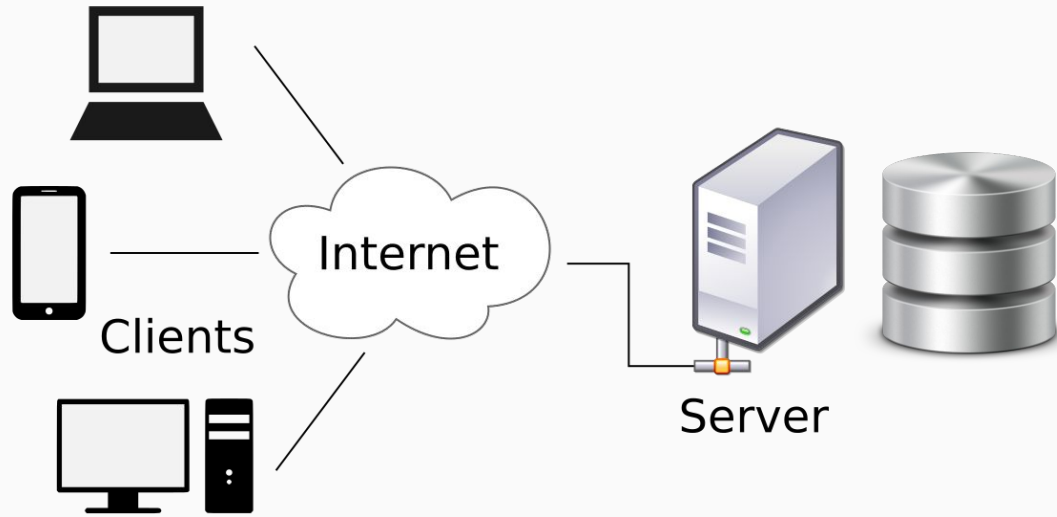
Why JavaScript?

Using Git

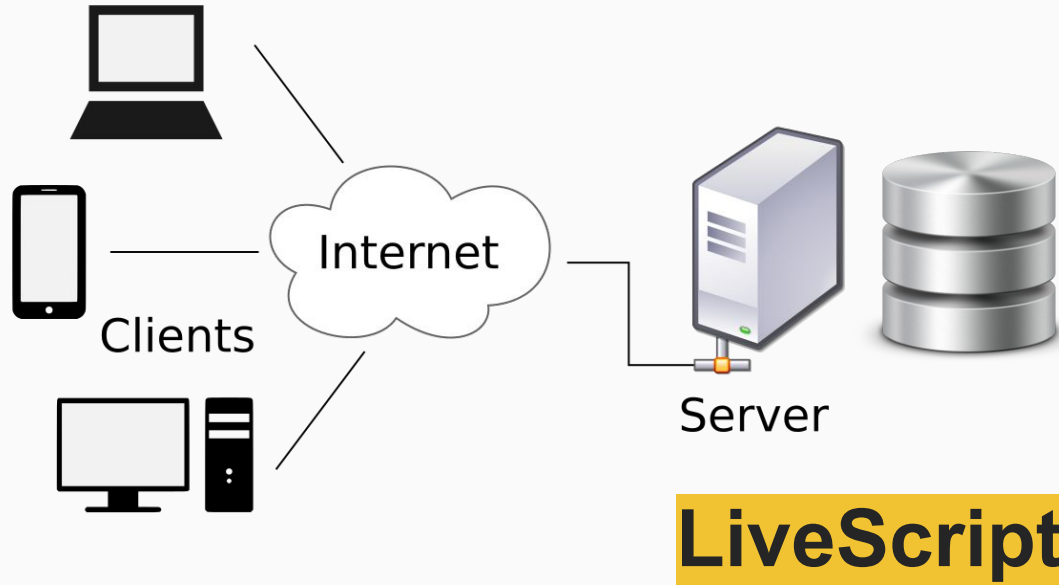
The MEAN Stack

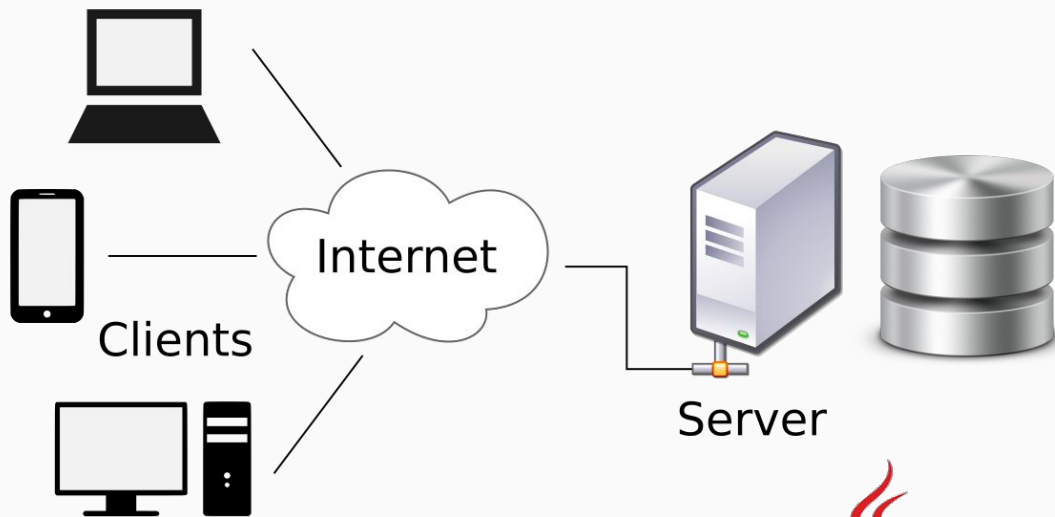
Using Cordova



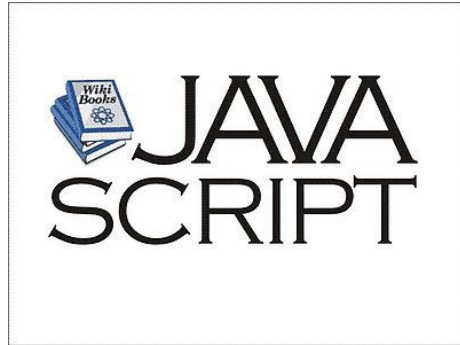
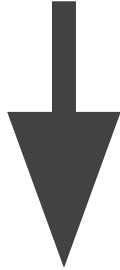


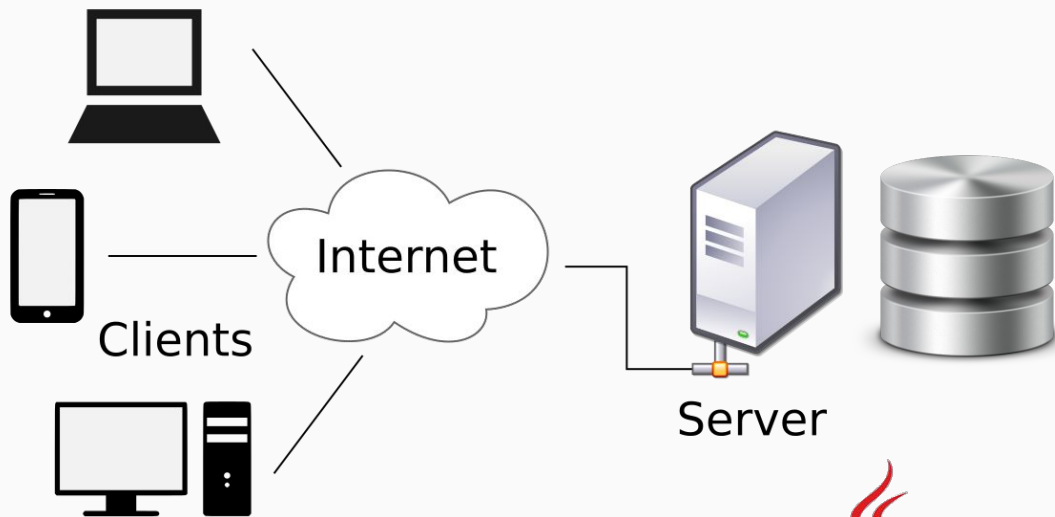
LiveScript

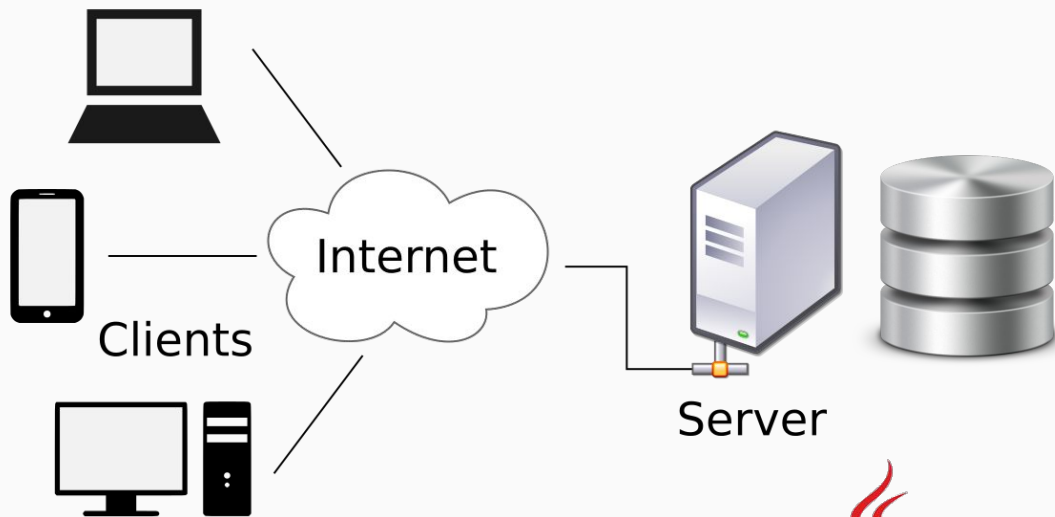




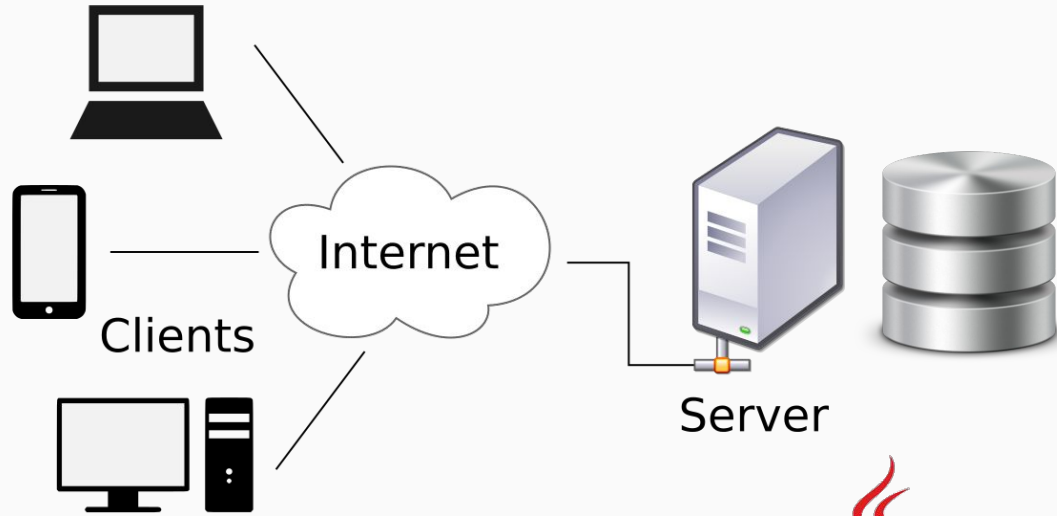
LiveScript

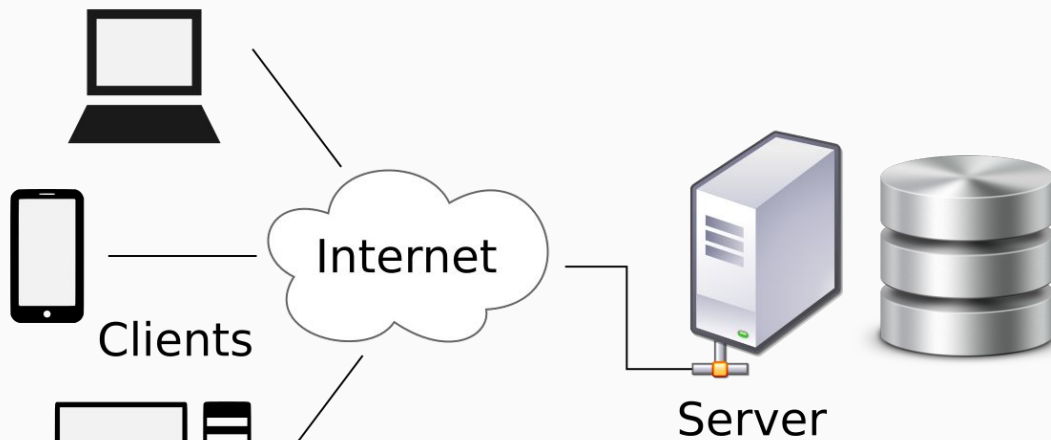


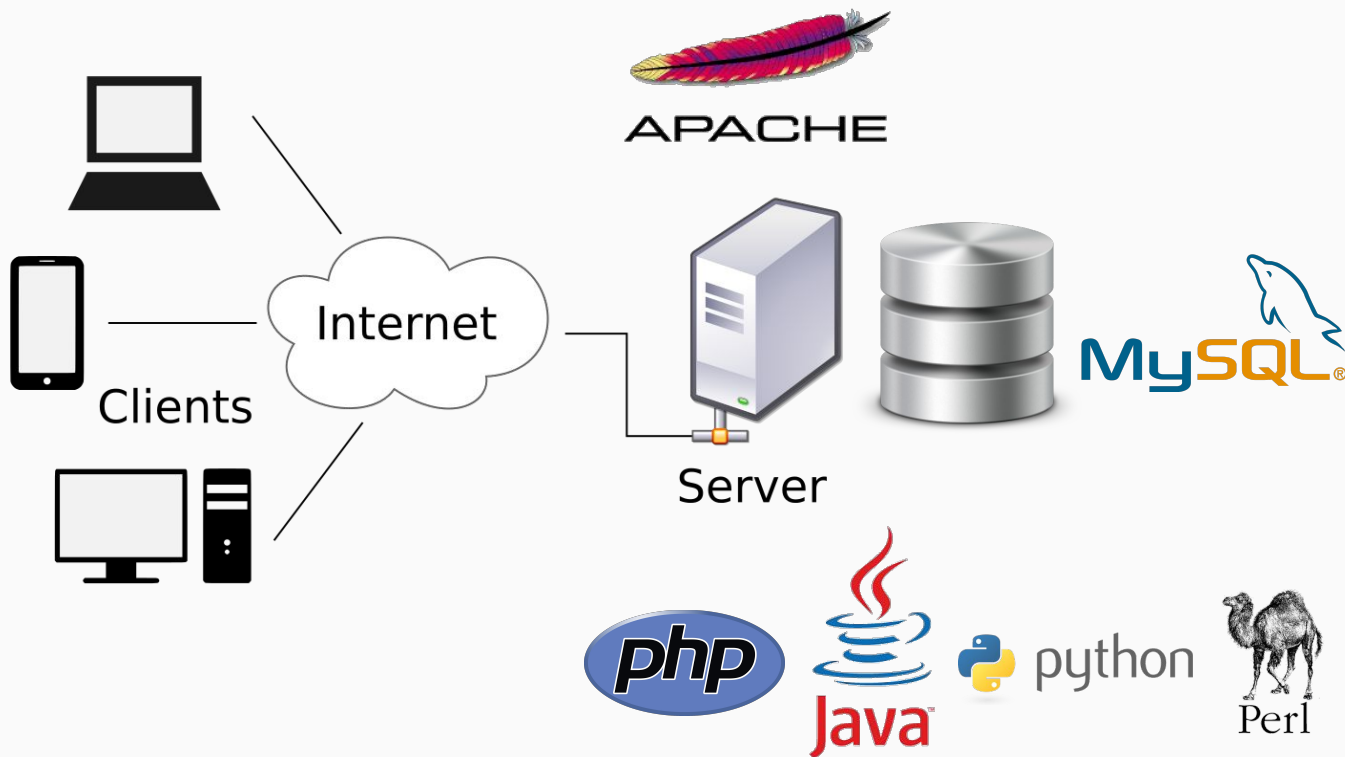


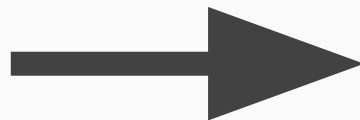
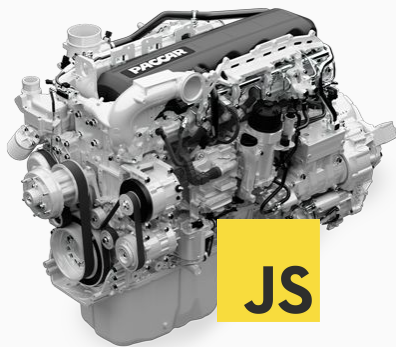




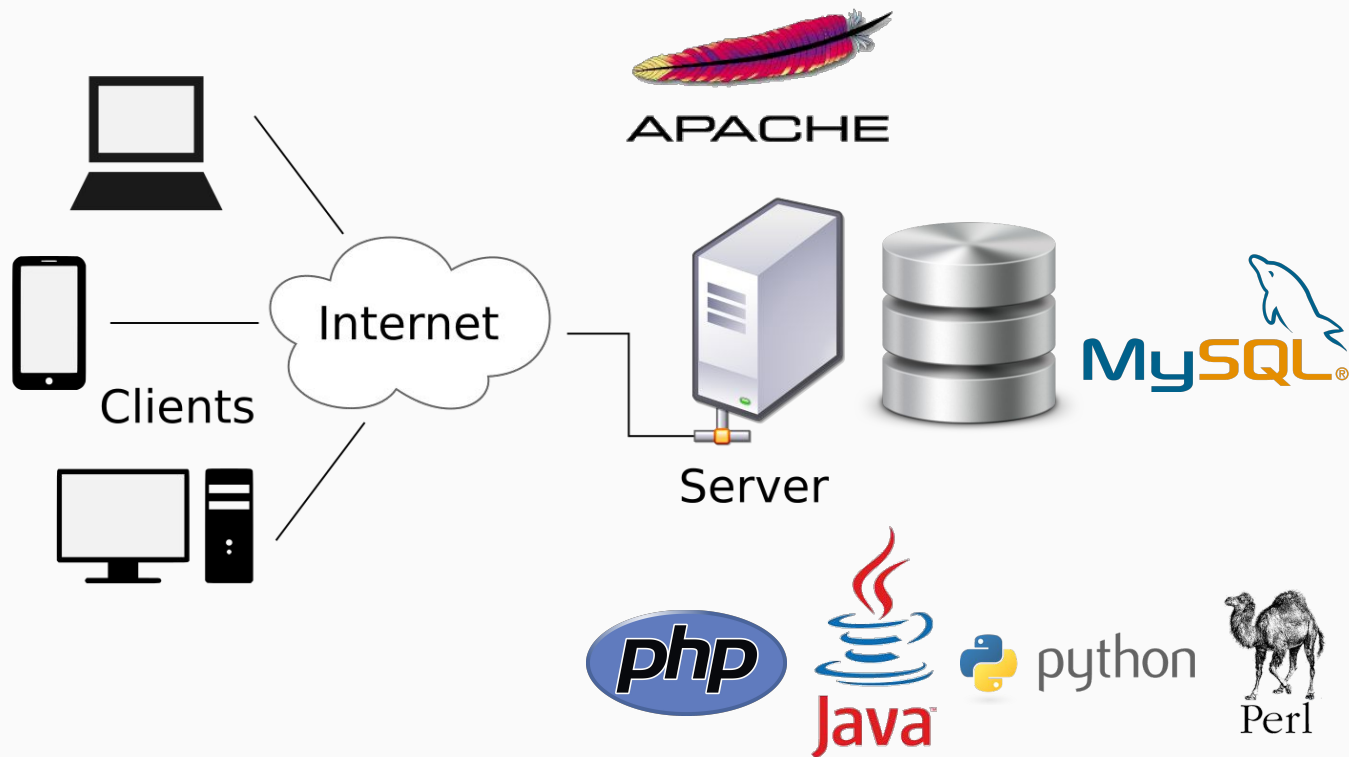


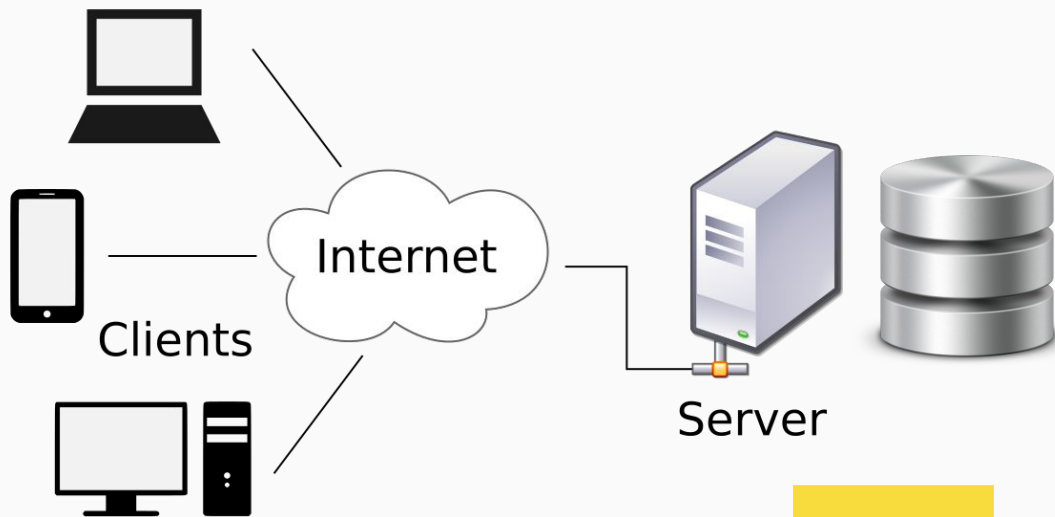












Building the Web



express



Node

Chrome's JavaScript engine without the browser

Run JavaScript on the server

Good for real time applications (like instant messaging)



Express

A library (set of tools) to help you
make server applications in NodeJS

express

MongoDB

A JavaScript Database

Very easy to add more database storage when you go from 10,000 to 10,000,000 users



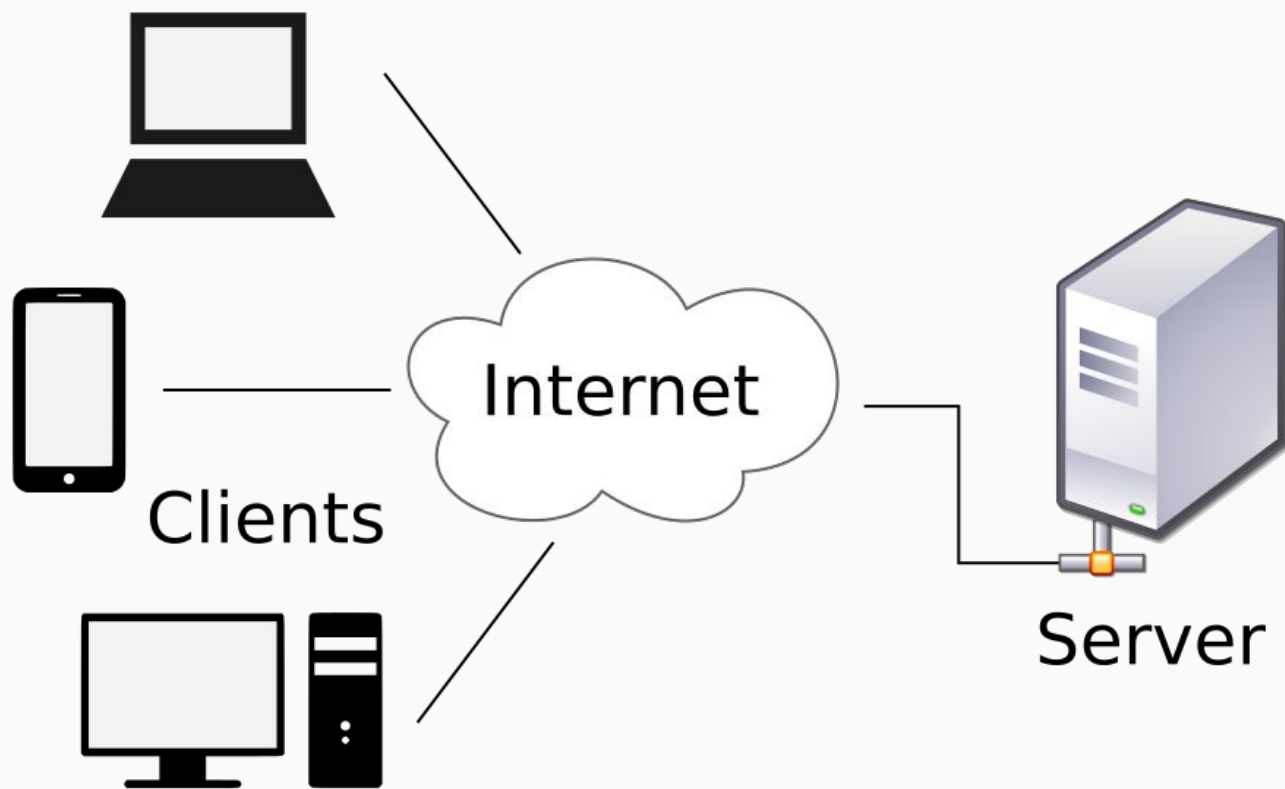
Angular

A a set of tools and a philosophy

Helps keep our code organised

Helps new programmers understand the code





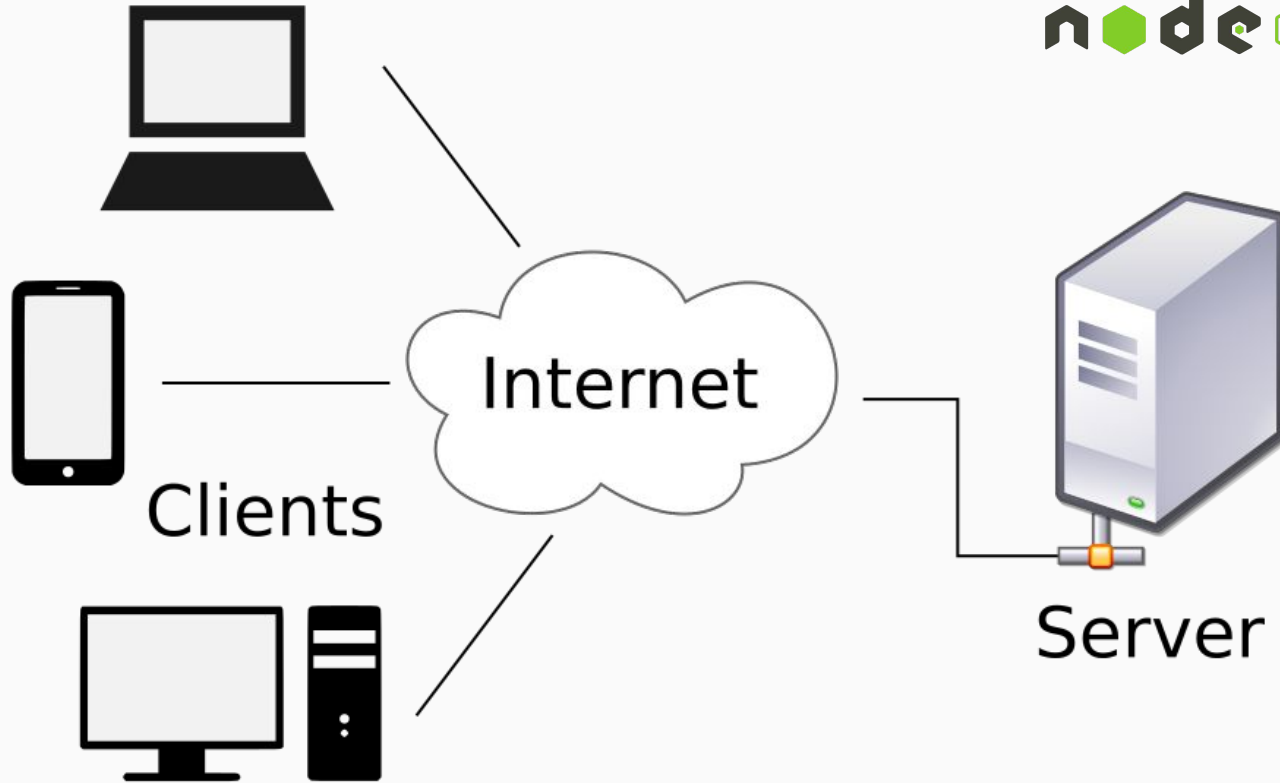
<http://mars.mu>

7QNZV6



express





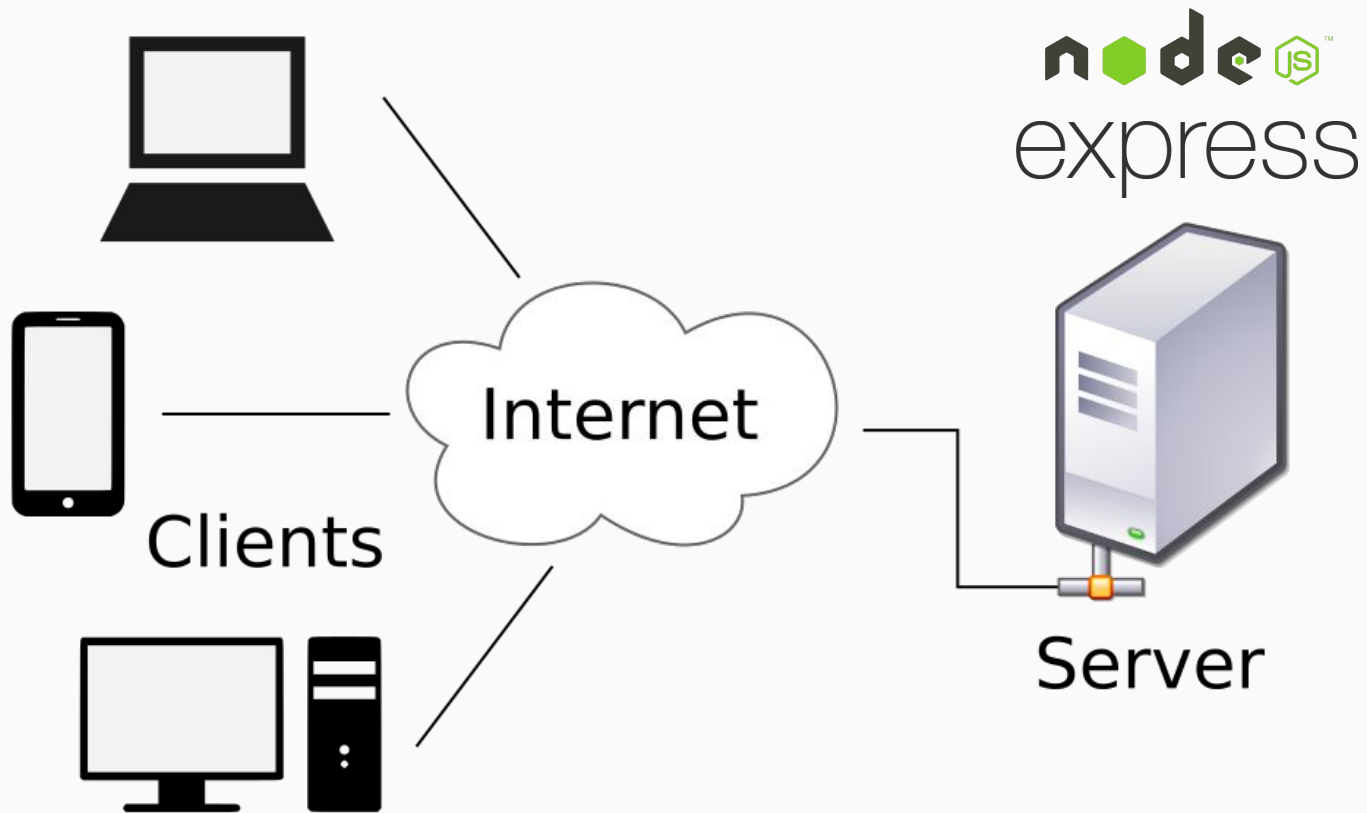
<http://mars.mu>

7QNZV6



express





<http://mars.mu>

7QNZV6



express





Clients



node
express



Server

<http://mars.mu>

7QNZV6



express





Clients



node
express



Server



mongoDB.

<http://mars.mu>

7QNZV6

Why build apps with JavaScript? (multiple correct answers)

- A) Web apps run on Android, iOS and Windows Phone
- B) Web apps run on Windows, OSX and Linux
- C) You can build the server and client with one language
- D) You don't have to worry about making users update their apps

JavaScript - The Language

```
int a = 0;
```

```
float b = 2;
```

```
double c = 5;
```

```
char d = 'a';
```

```
var a = 0
```

```
var b = 2
```

```
var c = 5
```

```
var d = 'a'
```

[codecademy.com](https://www.codecademy.com)

JavaScript - The Language

C Code

```
while ( true ) {  
    if (button_pressed == true) {  
        // fire the missiles  
    }  
}
```

```
// other stuff
```

JavaScript Code

```
button.onclick( fire )
```

```
function fire () {  
    // fire the missiles  
}
```

```
// other stuff
```

JavaScript - The Language

Original Example

```
button.onclick( fire )  
  
function fire () {  
    // fire the missiles  
}  
  
// other stuff
```

In-line Example

```
button.onclick( function () {  
    // fire the missiles  
} )  
  
// other stuff
```

goo.gl/misoRr

Get the slides



Git

Do I need this?

Group Report

Working in a team?

Working for a company?

Working for a client?

Key Commands

`git pull`

`git add -A`

`git commit`

`git push`

Side note `git clone <repository name>`

Does it work?

Type `git --version` into your terminal / command prompt

Raise your hand if you don't get something like `git version 1.9.1`

Cloning Practise

Go to: `goo.gl/09UN9Y`

Make sure you use https (unless you've set up git for ssh)

Cloning Practise

In your terminal `cd` to your desktop and type:

```
git clone <right click paste url>
```

Cloning Practise

Sign in to **github.com**

Click green **New repository** button (you may need to verify your account)

Cloning Practise

Enter **NewWords** as the **Repository Name**

Click **Initialize this repository with a README**

Add gitignore **Node**

Choose a license

Cloning Practise

Now see if you can **clone** your repository to the **desktop**.

Hint: `git clone <paste url here>`

NodeJS

What does Node do?

Program your computer:

- Access file system
- Access databases
- Automate tasks
- Build apps (like servers)

Does it work? (Windows)

Run `node -v` from the command line. **If it doesn't work:**

1. Click **Start > Control Panel**
2. In the **Search** box on the top right, type **"system path"** (no quotes)
3. Click **"Edit the system environment variables"**
4. Click the **Environment Variables** button on the bottom right.
5. In the lower box titled **"System Variables"** scroll down to **Path** and click the **Edit** button.
6. Add the path to NodeJS's bin file to the System Path

Does it work? (OSX/Linux)

Run `node -v` from the terminal. **If it doesn't work:**

Mac OSX

1. Type `touch .profile` to create a profile if none exists
2. Type `open profile` to edit your `.profile`
3. Add line `export PATH=$PATH:<path to node>`

Linux

1. Add `PATH=$PATH:<path to node>` to `.bashrc` in your home folder

Note: An example `<path to node>` may look like `"/home/nathan/Downloads/node-v4.4.0-linux-x64/bin/node"`

Hello, World!

Inside your **NewWords** repository make a text file named **index.js**

Write the following inside this file:

```
var name = "Barry"
```

```
console.log("Hello there, " + name)
```

Hello, World!

In your terminal, `cd` to the **NewWords** repository and type:

`node index.js` to run the JavaScript file.

Hello, World!

Now delete this file. It's time to get serious.

Stage 0

The MEAN Stack

What are we building?

An app to learn new words

Folder Structure

Two applications in your **NewWords** repository:

1. server
2. client

Client Files

Create three files in your client folder

1. index.html *
2. index.js
3. style.css

Server Files

Create one file in your server folder

1. `server.js`

npm init

Initialise an NPM project in each folder using `npm init`

Notice the new **package.json** file - this is like a cheatsheet for your app

npm install

Install express in your server folder by using `npm install express --save`

Building a Server

In server/server.js type the following:

```
var express = require('express')  
var app = express()  
  
app.use(express.static('../client'))  
  
app.listen(8080)
```

Let's test it out

In your terminal / command line run `node server.js` to run your server app

Let's test it out

Now, open a browser and go to `http://localhost:8080`

Let's test it out

You've just built your first webserver!

Good time to do a “git push”

In your command prompt/terminal type `git add -A`

[Bonus: type `git status` to check what git is ready to commit]

Good time to do a “git push”

Type `git commit` and type a message like “Create folder and file structure”

Good time to do a “git push”

Type `git push`

*** Please tell me who you are.

Run

Copy and paste these lines, one at a time, into the terminal using your email and name

```
git config --global user.email "you@example.com"  
git config --global user.name "Your Name"
```

to set your account's default identity.

Omit `--global` to set the identity only in this repository.

```
fatal: unable to auto-detect email address (got  
'nathan@somecomputer.(none)')
```

warning: push.default is unset; its implicit value has changed in Git 2.0 from 'matching' to 'simple'. To squelch this message and maintain the traditional behavior, use:

```
git config --global push.default matching
```

To squelch this message and adopt the new behavior now, use:

```
git config --global push.default simple
```

...

Copy and paste this line into
the terminal

Check GitHub

Go to your NewWords GitHub repository. You should be able to see the code you just pushed.

[GitHub may need a refresh before the latest code is shown]

Stage 1

The MEAN Stack

Linking CSS and JavaScript

Let's link the CSS and JS files:

```
<link rel="stylesheet" type="text/css" href="style.css">
```

```
<script src="index.js"></script>
```

Linking CSS and JavaScript

Check that they work by typing in some CSS to **style.css**:

```
body {  
    font-family: Helvetica;  
}
```

And JavaScript to **index.js**:

```
document.write("hello!")
```

Linking CSS and JavaScript

Refresh your browser on <http://localhost:8080>

Did it work?

Mobile Friendly

Make it mobile friendly:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

What is Angular?

<https://github.com/nathansherburn/MeLTS>

vs.

<https://github.com/SLC3/MARS>

Let's get some Angular in there

In your client folder, type `npm install angular --save`

Let's get some Angular in there

In client/index.html:

```
<html ng-app="myApp">
```

```
<body ng-controller="MyController">
```

```
<script src="node_modules/angular/angular.js"></script>
```

Let's get some Angular in there

In client/index.html:

```
<h1>Learn New Words</h1>
```

```
<div>
```

```
  <input ng-model="newWord">
```

```
</div>
```


Let's get some Angular in there

In client/index.js:

```
var myApp = angular.module('myApp',[])
```

```
myApp.controller('MyController', function($scope) {  
  $scope.newWord = 'cat'  
})
```

Let's test it out!

Back in your browser, go to `http://localhost:8080`

Git time again

See if you can **add**, **commit** and then **push** your code to your NewWords GitHub repository.

Hint: Run the following commands from the 'root directory' (NewWords folder):

```
git add -A  
git commit  
git push
```

Stage 2

The MEAN Stack

JSON - JavaScript Object Notation

```
{  
  "firstName": "Nathan",  
  "currentAge": 25,  
  "isStudent": true,  
  "university": "Monash University"  
}
```

Angular Services

Add the following to server/server.js file:

```
app.post('/saveCurrent', function (req, res) {  
  console.log("got a request to save something!")  
})
```

Angular Services

Add the following somewhere in the **body** of your client/client.html file:

```
<button ng-click="saveThisWord()">Save</button>
```

Angular Services

Add the following to client/index.js:

```
myApp.service('HistoryService', function($http) {  
  
  })
```


Angular Services

Add the following code inside the **HistoryService**:

```
var baseUrl = "http://localhost:8080/"

this.saveWord = function (newWord) {
  var url = baseUrl + "saveCurrent"
  return $http.post(url, {"word": newWord})
}
```

Angular Services

Add the following code inside **MyController**:

```
$scope.saveThisWord = function () {  
    HistoryService.saveWord( $scope.newWord )  
    .then(saveSuccess, error)  
}
```

(this guy will hook up to our ng-click in index.html)

Angular Services

Add the following code inside **MyController**:

```
function saveSuccess (json) {  
    console.log(json)  
}
```

```
function error (err) {  
    console.log(err)  
}
```

Angular Services

Let's test it on `http://localhost:8080`

(Spoiler: it probably won't work yet)

Angular Services

In your server folder, type `npm install nedb --save`
and `npm install body-parser --save`

MongoDB-ish

Back in your server/server.js file, add:

```
var bodyParser = require('body-parser')  
app.use(bodyParser.json())
```

```
var Nedb = require('nedb')  
var database = new Nedb({ filename: './data/data.db', autoload:  
true })
```

MongoDB-ish

Remove:

```
console.log("got a request to save something!")
```

and...

MongoDB-ish

And write:

```
var data = { word: req.body.word, date: Date.now() }

var done = function () {
  console.log('I just wrote to the database')
  res.end("done")
}
```


MongoDB-ish

Finally, add:

```
database.insert(data, done)
```

Let's test it!

Restart node and run the latest server with `ctrl+c` and then `node server.js`

Back in your browser, refresh `http://localhost:8080` and mess with the buttons

Check the `server/data` folder to see if your words are being saved

[Note: this folder was automatically created for you]

Why do we have MongoDB?

People want apps that will scale (like Google's)

It's free and open source

Good for prototyping

It's JavaScript

Git time again

Add, commit and then **push** your code to your NewWords GitHub repository.

Hint: Run the following commands from the 'root directory' (NewWords folder):

```
git add -A  
git commit  
git push
```

Stage 3

The MEAN Stack

Retrieving from the Database

Add the following lines to your client/index.html file:

```
<div ng-repeat="w in words"> {{w.word}} </div>
```

```
<button ng-click="getSavedWords()">Update List</button>
```

Retrieving from the Database

Add this function to your HistoryService (found in client/index.js):

```
this.getSaved = function () {  
  var url = baseUrl + "getSaved"  
  return $http.get(url)  
}
```

Retrieving from the Database

Add this code to MyController (in client/index.js):

```
$scope.words = []  
  
$scope.getSavedWords = function() {  
  HistoryService.getSaved()  
    .then(loadSuccess, error)  
}
```


Retrieving from the Database

Add this function to MyController (found in client/index.js):

```
function loadSuccess (json) {  
  $scope.words = json.data  
}
```

Retrieving from the Database

Add this route to server/server.js

```
app.get('/getSaved', function (req, res) {  
  // setup query and done function here  
  
  database.find(query, done)  
})
```

Retrieving from the Database

Replace `// setup query and done function here` with:

```
var query = {}

var done = function (err, data) {
  console.log('I just read stuff from the database')
  res.send(data)
}
```

Does it work?

Restart node and run the latest server with `ctrl+c` and then `node server.js`

Refresh localhost:8080 in your browser again to see if it worked.

Can you save and load from the database?

Git time again

Add, commit and then **push** your code to your NewWords GitHub repository.

Hint: Run the following commands from the 'root directory' (NewWords folder):

```
git add -A  
git commit  
git push
```

Stage 4

The MEAN Stack

Gif it up!

Add the following line to your client/index.html:

```

```

Gif it up!

Add the following Angular service to your client/index.js:

```
myApp.service('GifService', function($http) {  
  
    var baseUrl = "https://api.giphy.com/v1/gifs/"  
    var apiKey = "dc6zaTOxFJmzC"  
  
    // We'll add some functions to get gifs here  
  
})
```


Gif it up!

Add the following functions to your client/index.js:

```
this.getGifs = function (query) {  
  var url = baseUrl + "search?q=" + query + "&api_key=" + apiKey  
  return $http.get(url)  
}
```

Gif it up!

Add the following function to MyController:

```
$scope.showGifs = function($event) {  
    GifService.getGifs( $event.currentTarget.innerHTML )  
    .then(gifSuccess, error)  
}
```

Gif it up!

Add the following function to MyController:

```
function gifSuccess (json) {  
  if (json.data.data[0]) {  
    $scope.gifUrl = json.data.data[0].images.fixed_height.url  
  } else {  
    $scope.gifUrl = "http://goo.gl/tioFyj"  
  }  
}
```

Gif it up!

Add the following line at the top of MyController to hold the image url:

```
$scope.gifUrl = ''
```

Finished!

Cordova

Building Native Apps

Normally you have to make separate apps for

- iOS (Swift or Objective C)
- Android (Java)
- Windows Phone (C#)



Can we do better?

What if we had a:

- Stripped down browser
- Can only run one website
- Has access to all of the phone's features

Cordova

Simple browser with access to all of the phone's features.

One code base for

- iOS
- Android
- Windows Phone



Cordova

Simple browser with access to all of the phone's features.

One code base for

- iOS
- Android
- Windows Phone



Using Cordova

Install it globally (-g): `npm install -g cordova`

Create an app: `cordova create MyCordovaApp` (open this file)

Add a platform (Android, iOS etc): `cordova platform add android`

Test it out: `cordova run android`

[Note: you'll need the Android SDK kit from android.com]

Thanks for coming :)

Anonymous Survey

goo.gl/forms/Wsni3WSQaz

Resources

This Workshop

- github.com/nathansherburn/web-workshop (goo.gl/09UN9Y)
- And the [slides](https://goo.gl/misoRr) (goo.gl/misoRr)

Good Libraries to Checkout:

- [Socket.io](https://socket.io/) - send data around in real time
- [Mongoose](https://mongoosejs.com/) - help structuring your data
- Native [MongoDB driver](https://mongodb.github.io/mongo-node-driver/) - same as NeDB but uses a MongoDB database

Resources

Recommended courses from [codecademy](#):

- JavaScript
- HTML & CSS
- Angular
- Git

Publishing your app on the internet:

- Heroku
- Nodejitsu
- Google Cloud Platform (not free)

Some good tutorials to get started:

- [scotch.io](#)
- [LearnYouNode](#)

Use Google! :)

- eg “angular services tutorial”
- eg “mongodb tutorial”

Resources

Getting [started](#) with MongoDB on Node

Codeschool's [Shaping up with Angular](#)

How to [query](#) MongoDB

Setting up and using [Express](#)

Exclusive Monash JavaScript [Resources](#) from Dr Michael Wybrow

Building [Cordova](#) apps