MEAN

GOALS

Have Fun

Make Stuff

Learn

Feel Good

What the heck is the MEAN stack?

A combination of JavaScript -based technologies /frameworks that are used to create web applications.

Previously the LAMP stack was very popular (Linux, Apache, MySQL, PHP/Python/Perl).

The MEAN stack is all written in a single language (JS) and is used for building single-paged applications.

Let’s build a house!

HOUSE	WEB PAGE
Foundation, Walls, Windows, Roof	HTML Elements: <body>, <h1>, , <p>
All walls are painted red, Eastern-facing windows have purple trim	CSS rules: body { color: red } p { color: purple } .eastern { text-align: center }
Light switch Thermostat	Javascript: When user flicks light switch, turn on lights When thermostat timer ends, turn off heat

Folder structure

- public <!-- holds our front-end files -->

----- main.js <!-- Angular code -->

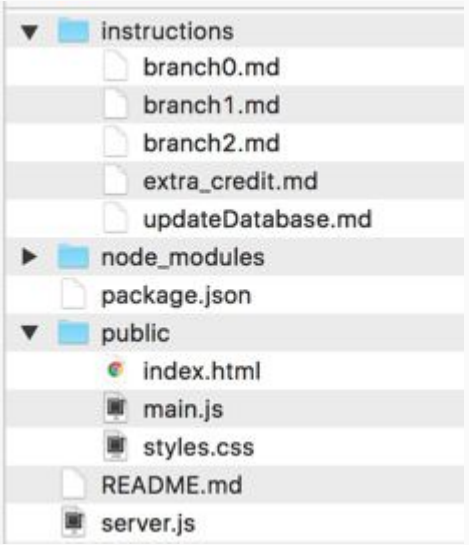
----- index.html <!-- main HTML view -->

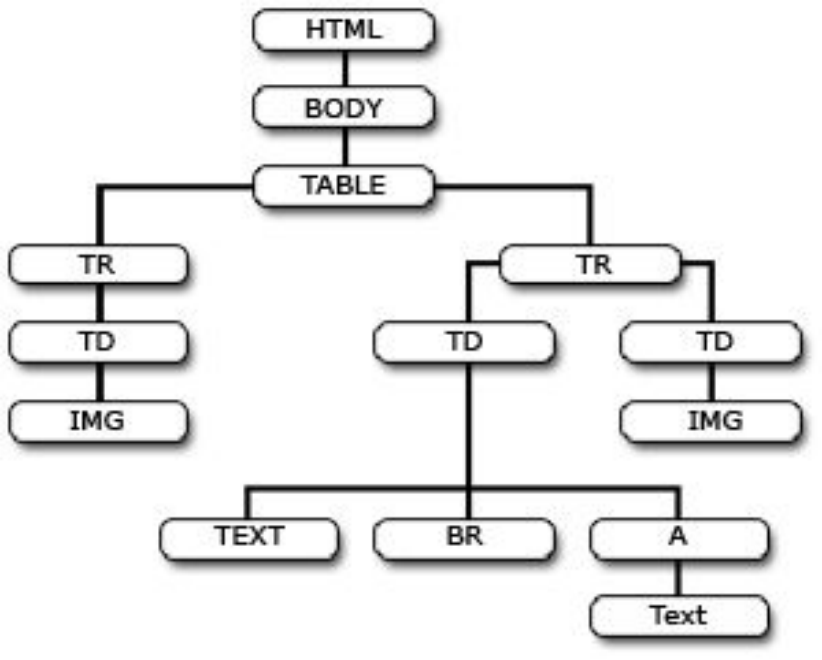
----- styles.css <!-- CSS stylesheet -->

- package.json <!-- npm configuration file -->

- server.js <!-- Backend Node file - contains database config -->

- instructions <!-- Step by Step guide to building this application-->





- Animals
  - Birds
    - Parrot
    - Sparrow
  - Mammals
    - Humans
    - Cows
  - Fish
    - Salmon
  - Reptiles
    - Snake
    - Lizard
- Plants
  - Trees
    - Mango
    - Banyan
    - Redwood
  - Shrubs
    - Cactus
    - Congress

JSFiddle

The HTML, CSS, and JS file all reference each other to create the web page.

```
<body>
  <h1> This is a header</h1>
  <p>
   This is a paragraph.
  </p>
  <input type="text" id="input1"
/>
  <button id="button1">
  Pop-up
  </button>
</body>
```

```
body {
  color: blue
}

h1 {
  color: red
}

p {
  color: green
}
```

```
document.getElementById("button1").onclick = function() {
  alert(  document.getElementById("input1").value  );
};
```

Javascript is run in the browser

Javascript is mainly run in the browser - directly in Chrome, Safari, Firefox, and even Internet Explorer. To see this for yourself - open up your browser's console and paste in the following code:

```
function bark(name){
  return name + " says woof";
}

function moo(name){
  return name + " says moo"
}
```

You can invoke a function by calling it with parentheses like so:

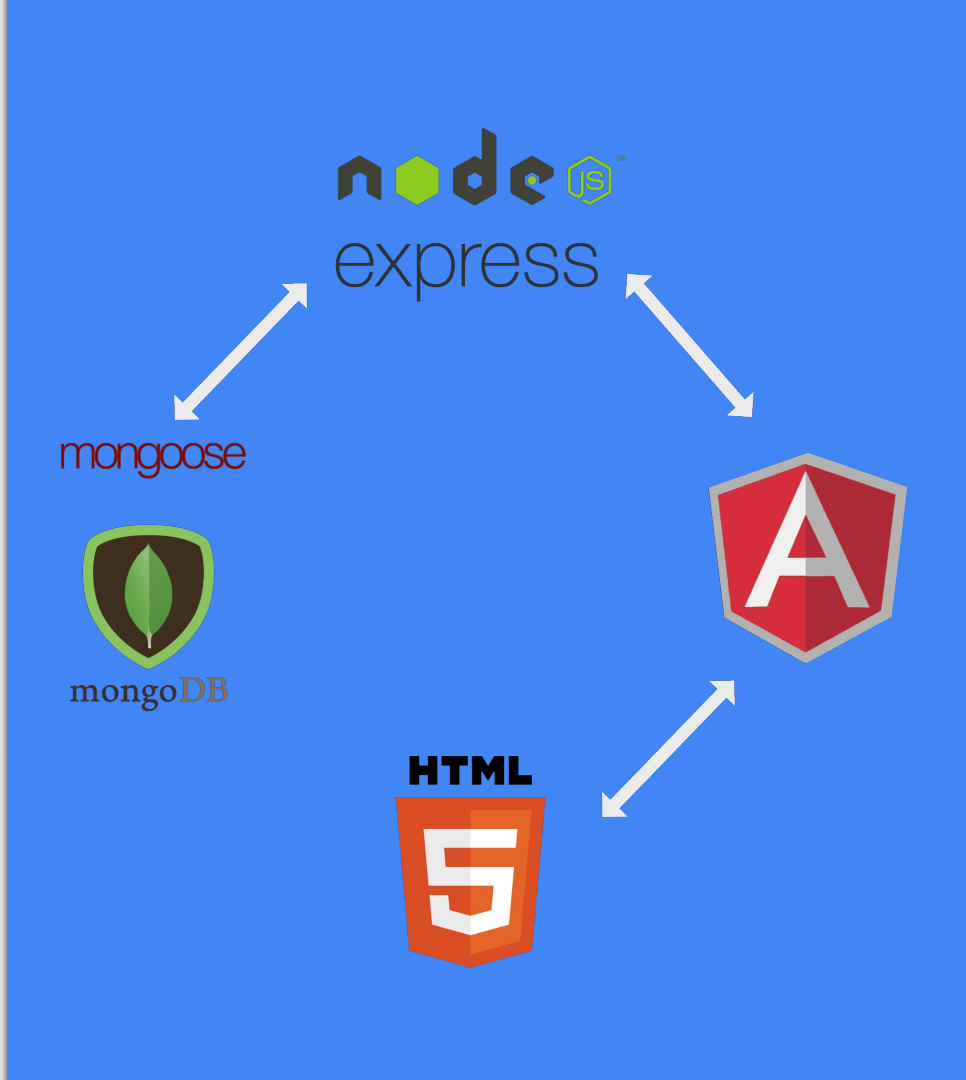
```
bark("sparky"); // returns "sparky says woof"

moo("bessie"); // returns "bessie says woof"
```

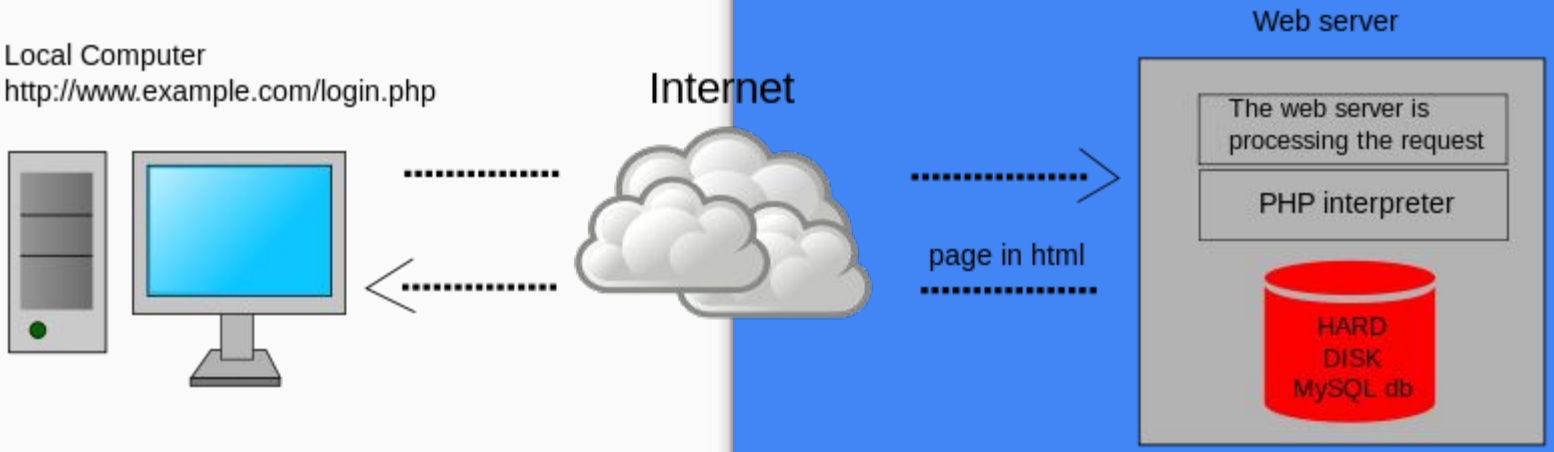
Javascript can also be used for servers

- Listens for requests and handles them
- Serves up static files like HTML, CSS
- Connects to databases
- Can be used to gather data from other web pages

MEAN

MongoDB
Express
Angular
Node




Local Computer
http://www.example.com/login.php
Internet
Web server
The web server is processing the request
PHP interpreter
page in html
HARD DISK MySQL db

Javascript Objects

Objects are collections of properties.

You can define an object by using the “object literal notation” like this:

```
var myObject = {name: “Mary” };
```

Properties are set on objects using bracket or dot notation like this:

```
myObject.name = “Tom”; //over-writes the name property on myObject

myObject[‘age’] = 56; // sets the age property on myObject
```

Object Methods

You can store functions on objects like this:

```
myObject.printName = function() {
    console.log(this.name);
}
```

The **this** keyword refers to the object that is calling the function. In this case, it refers to myObject.

```
myObject.printName( ); // prints TOM
```

Loops

You can iterate through a collection's values and act on them:

```
var myLuckyNumbers = [4, 8, 15, 16, 23, 42];

var double = function(number) { return number * 2; };

for (var i=0; i < myLuckyNumbers.length; i++) {

    console.log(double(myLuckyNumbers[i]));

}
```

Diving into Node’s event loop

Many traditional web-serving technologies handle requests by creating a new thread using up RAM.

Node is single-threaded, and uses an event loop (a queue of tasks) run in order. A node server can handle thousands of requests simultaneously, as it processes them in a non-blocking i/o manner.

JSON

JSON is an acronym for “JavaScript Object Notation” which has become a widely-used format for storing and transmitting data.

This is a JSON string:

```
var myDog = ‘ { “name” : “Marley”, “breed” : “Bichon-Poodle”, “age” : 77 } ‘ ;
```

We take this JSON string and turn it into a Javascript Object by invoking the JSON.parse function:

```
JSON.parse(myDog);
```

We can convert objects into JSON by invoking the JSON.stringify() function:

```
JSON.stringify(otherDog);
```

Non-relational Databases

Relational Vs Non-Relational (SQL vs no-SQL)

Non-relational databases are sometimes schema-less, may be structured like a graph, store data as JSON objects.

MongoDB stores objects in the database in a key: value JSON format

http://www.jsoneditoronline.org/

Autodesk Contacts:
John Choi - John.choi@autodesk.com
Yu Pu - yu.pu@autodesk.com
Joy Zhao - qinqin.zhao@autodesk.com

Instructors
Chris Bassano - chris.bassano@reactorcore.com - github.com/christo4b
Jake Pace - jake.pace@hackreactor.com - github.com/jakeyrob

Course Materials:
github.com/christo4b/mean_adsk