PROGRAMMING FUNDAMENTALS

LESSON #4

PRESENTED BY

ANDREW BUNTINE

WHO AM I, ANYWAY?

- Technical Director at Hardhat
- 15 years experience
- Web developer
- Indie game developer
- bunts.io
- github.com/buntine

WHAT IS THE GOAL?

DEMYSTIFY THE LINGO

CREATE BASIC WEBSITES

SURVEY OF TECHNOLOGIES

WHY ARE YOU HERE?

1. INTRODUCTION

- Vocabulary
- Development process
- Basics of the Web
- Coding introduction

2. FRONT END

- HTML
- CSS
- Javascript

3. BACK END

- Ruby
- Rails
- Make a web app

4. THE REST!

- Tie up the loose ends!

GOALS FOR TODAY

- Recap of lesson #3
- Web Frameworks
- MVC
- Databases
- My choices
- Javascript
- JQuery
- Write some Javascript
- Image formats

RECAP

THINKING ALGORITHMICALLY

- Using creativity to solve problems.
- Breaking problems into sub-problems.
- Finding patterns.

THINKING LIKE A TURTLE

- We wrote some code in **logo**.
- We **refactored** that code to be succinct.

WE WROTE RUBY

- We applied some of our learnings in **ruby**.
- Variables.
- Functions.
- Looping.
- Conditionals.

WEB FRAMEWORKS

WHAT ARE THEY?

Softwares that takes away the pain of building dynamic web applications.

POPULAR OPTIONS

- Ruby on Rails
- Django
- AngularJS
- ASP.NET MVC
- + about 900 more...

WHAT DO THEY DO?

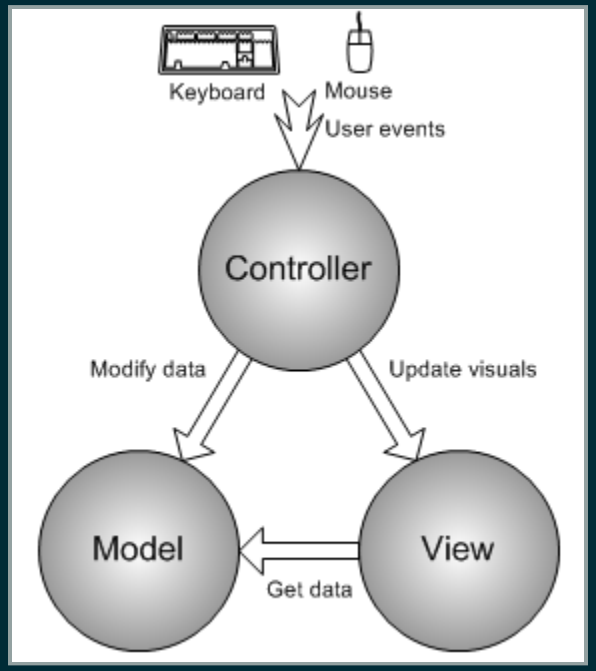
- Make life easier for Web Developers
- Support common tasks associated with web development:
  - Handling requests
  - Communicating with databases
- Provide standards
- Sessions / Cookies
- Security
- Caching

MVC

WHAT IS IT?

- Model - View - Controller
- It's a software pattern. A way of structuring our code.
- It's very common in Web Frameworks. Infact, nearly all major frameworks are based upon it.

HOW DOES IT WORK?



MODEL

- Represents the data that backs an application.
- Handles application rules and logic.
- Typically provides an abstraction over a record in a database.

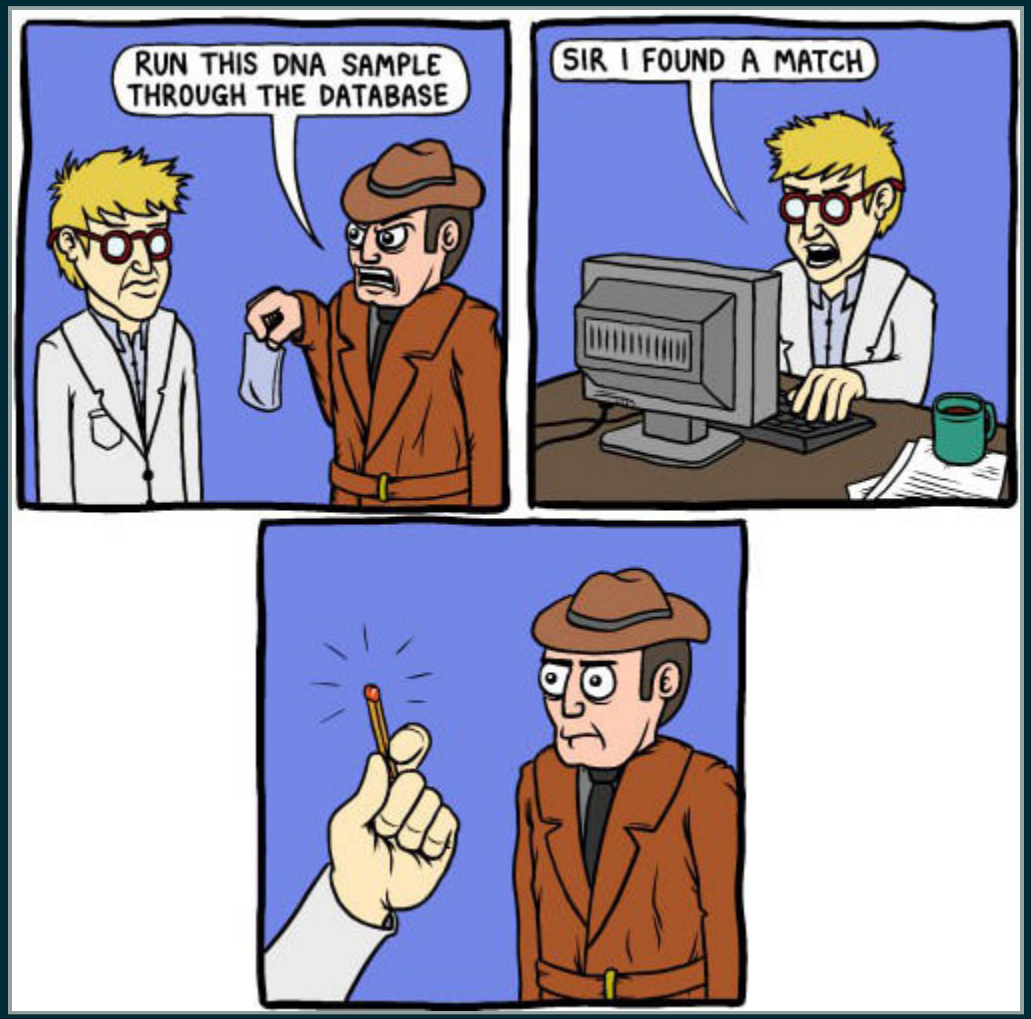
VIEW

- Any output of information is considered a view.
- Multiple views may exist - one for mobile and one for desktop, for example.

CONTROLLER

- Handles interaction with the user.
- Accepts input and provides output
- Delivers information from the model via the view.
- It's the glue!

DATABASES



WHAT ARE THEY?

- Software for storing data in a structured format
- A way of modelling relationships between data
- You may think of them as big spreadsheets
- We speak to them using SQL

HOW ARE THEY MADE?

- We start by drawing them as **Entity Relationship Diagrams** (ERDs)
- Then we clean them up via **normalisation**.
- Finally, we translate the design into **SQL**.

EXAMPLE #1

- I want to store products

EXAMPLE #2

- I want to store products
- I want to group them into categories
- Each product has **one** category

EXAMPLE #3

- I want to store products
- I want to group them into categories
- Each product might have **many** categories

MY CHOICES


Sinatra

SINATRA

- Written in Ruby.
- Simple, low-complexity, low-featureset.
- Picster was written in Sinatra.


RAILS

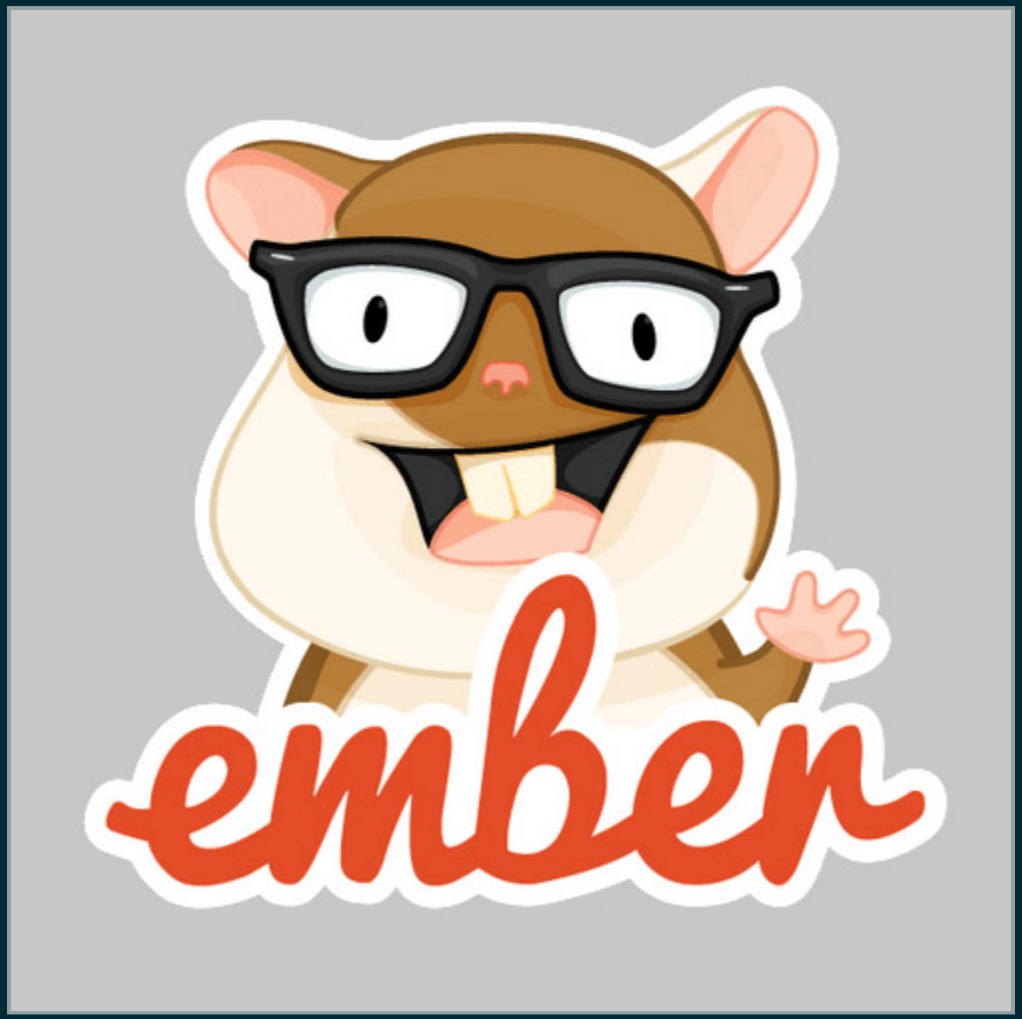
RUBY ON RAILS

- Written in Ruby.
- Big, complex.
- Suited to large applications.


Bottle

BOTTLE

- Written in Python.
- Small, easy to use. Python equivalent of Sinatra.


ember

EMBER.JS

- Written in Javascript.
- Stores lots of application logic on the client.
- Delegates the server to expose simple APIs.


laravel

LARAVEL

- Written in PHP.
- Big. Similar to Rails.
- Suited to large applications.

THERE IS NO PERFECT ANSWER!

- You need to weigh up the options.
- What do you already know?
- Which features make sense for your project?
- How well maintained is the language/framework?
- Will it be around in 12 months?

JAVASCRIPT


What is the world's most popular language?
A: French
B: English
C: Spanish
D: JavaScript

WHAT IS IT?

- General-purpose programming language
- Developed by Brenden Eich at Netscape in 1995
- Implemented in all major browsers, with slight differences
- Used to control behaviour and interact with the user
- Standardised under the name ECMAScript

WHAT DOES IT LOOK LIKE?

```
var hello = function(name) {
      return "Hello, " + name;
    };

var hello_world = hello("World!");

console.log(hello_world);
```

WHAT MAKES IT A PROGRAMMING LANGUAGE?

- It's **"Turing Complete"**!
- It provides constructs that we can use to create **algorithms**

POPULAR LIBRARIES

- JQuery (DOM wrapper, AJAX, etc)
- Three.js (Graphics)
- AngularJS (Frontend framework)
- Ember.js (Frontend framework)
- React.js (UI framework)

JQUERY

- Simplifies Javascript programming significantly
- Abstracts away much of the cross-device messiness
- Animations become very simple (no trigonometry requiired!)
- AJAX requests become a lot simpler to fire and handle

PLAIN JAVASCRIPT

```
var heading = document.getElementById("header");

heading.innerHTML = "Hello, world!";
heading.style.backgroundColor = "#eee456";
```

JAVASCRIPT + JQUERY

```
$("#heading")
  .html("Hello, world!")
  .css("background-color", "#eee456");
```

LEARN JAVASCRIPT FIRST!

- JQuery does a lot of **magic** under the hood
- Make sure you understand Javascript well before learning JQuery
- Including Jquery requires the browser to interpret an extra ~20,000 lines of code

IMAGE FORMATS

PNG

- Portable Network Graphics
- Supports transparency
- Lossless data compression, but high compression means more decoding
- Great for photos and images with lots of colour variations

JPG

- Joint Photography Experts Group
- No transparency
- Lossy compression
- Great for photos and images with lots of colour variations

GIF

- Graphics Interchange Format
- Supports transparency (no semi-transparency)
- Supports animation
- Very limited colour palette (256)
- Limited compression, so big filesize

SVG

- Scalable Vector Graphics format
- Supports transparency
- Best suited to flat colours and geometric shapes
- Can be scaled to any size without lost quality or increased file size

JPG AND SVG ARE USUALLY ENOUGH!

- JPG is great for photo-esque images
- SVG is great for simpler shapes, logos, etc

REVIEW

QUESTIONS?

THANK YOU!