welcome to...

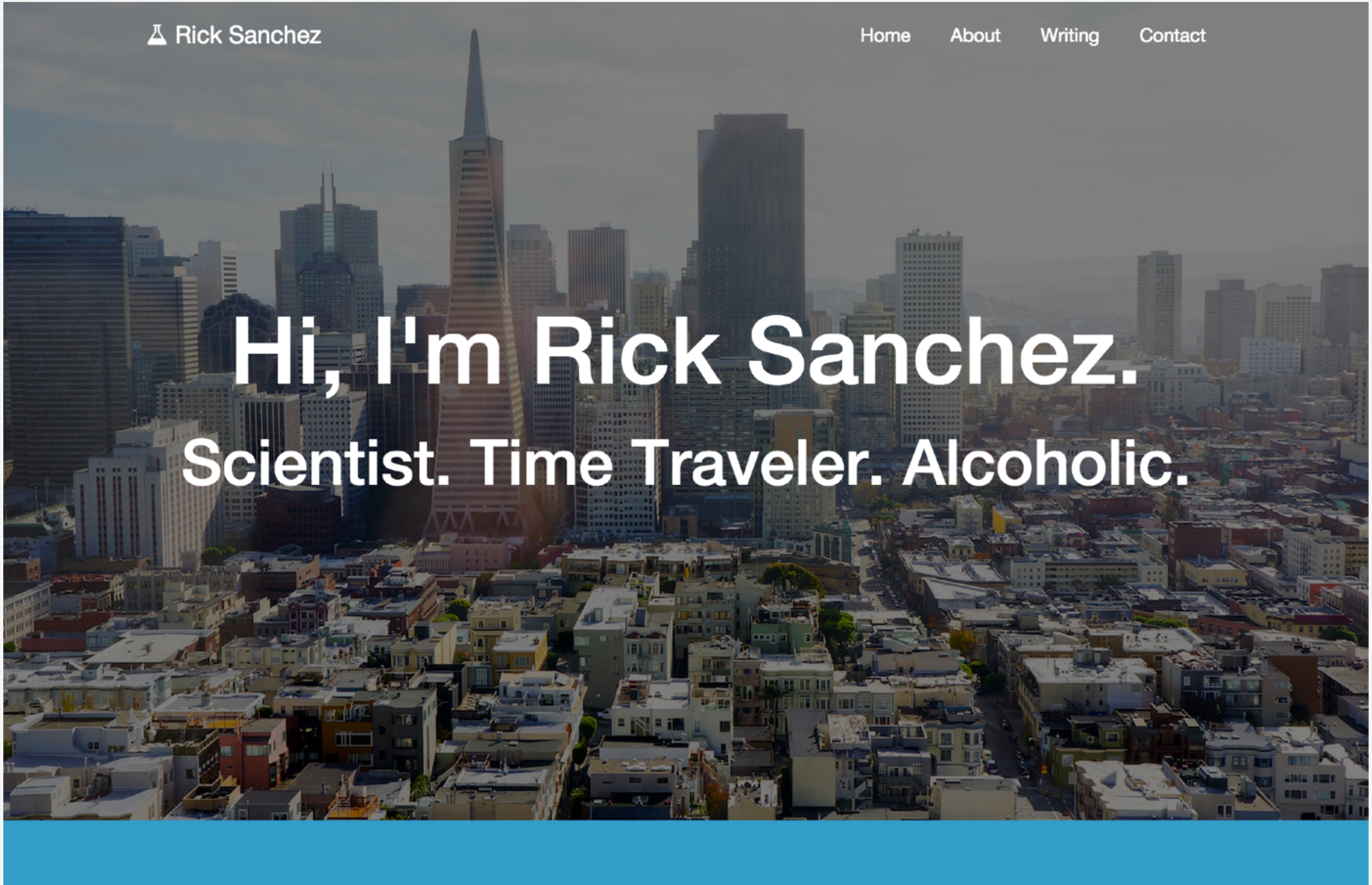
BOILERCAMP

HOW TO WEB DEV

Introduction / Project Overview

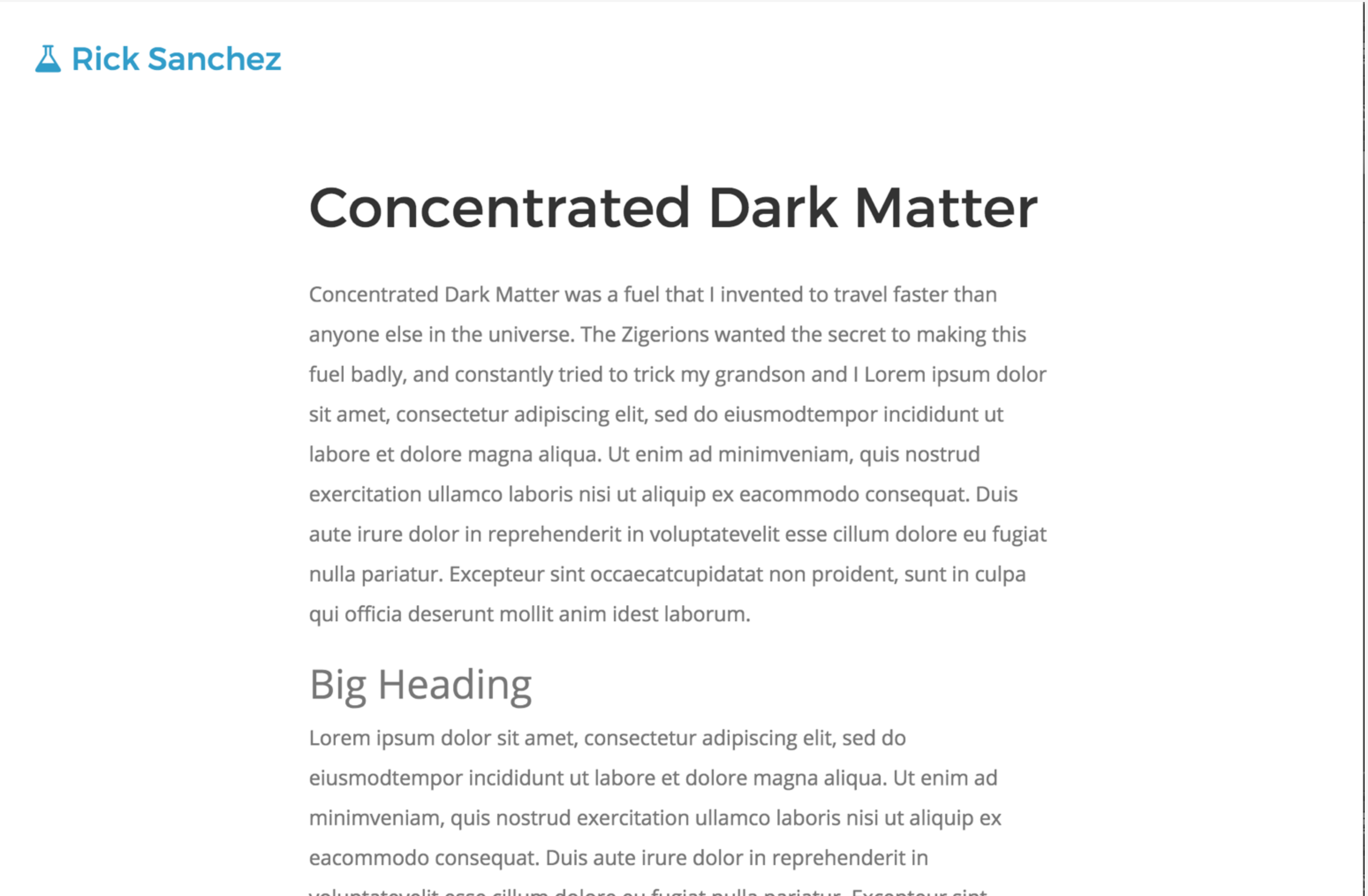
The Plan

Personal Website/Blog



Hi, I'm Rick Sanchez.

Scientist. Time Traveler. Alcoholic.



Rick Sanchez

Concentrated Dark Matter

Concentrated Dark Matter was a fuel that I invented to travel faster than anyone else in the universe. The Zigerions wanted the secret to making this fuel badly, and constantly tried to trick my grandson and I Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmodtempor incididunt ut labore et dolore magna aliqua. Ut enim ad minimveniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex eacommodo consequat. Duis aute irure dolor in reprehenderit in voluptatevelit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecatcupidatat non proident, sunt in culpa qui officia deserunt mollit anim idest laborum.

Big Heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmodtempor incididunt ut labore et dolore magna aliqua. Ut enim ad minimveniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex eacommodo consequat. Duis aute irure dolor in reprehenderit in voluptatevelit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint

Schedule

- Introduction / Project Overview
- HTML / CSS
- Client-side JavaScript
- Lunch
- Node.js / Express.js
- MongoDB / MVC

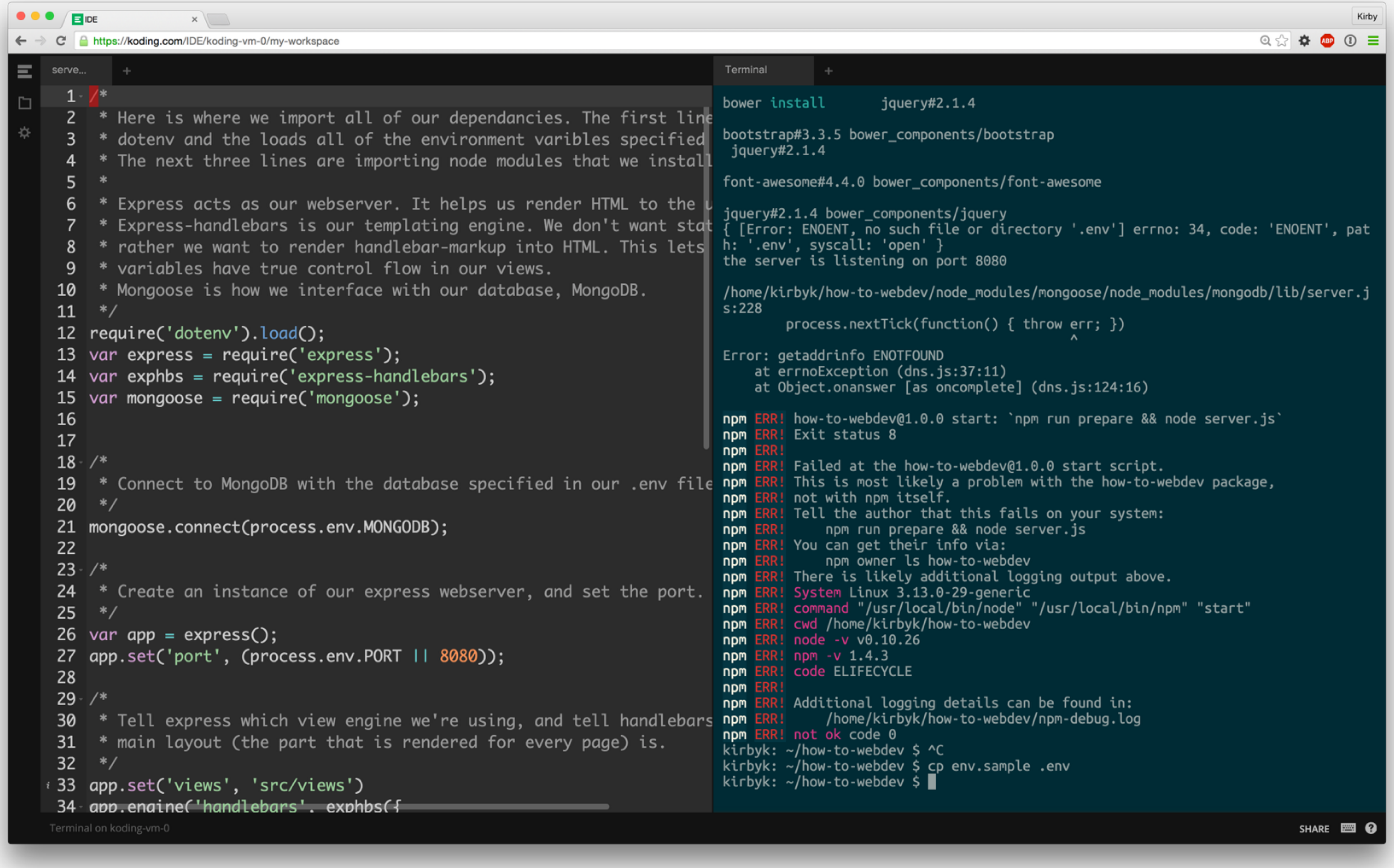
It’s okay if you don’t understand everything.

It’s okay if you don’t finish a section.

Please ask for help.

BoilerCamp.slack.com

Our Environment

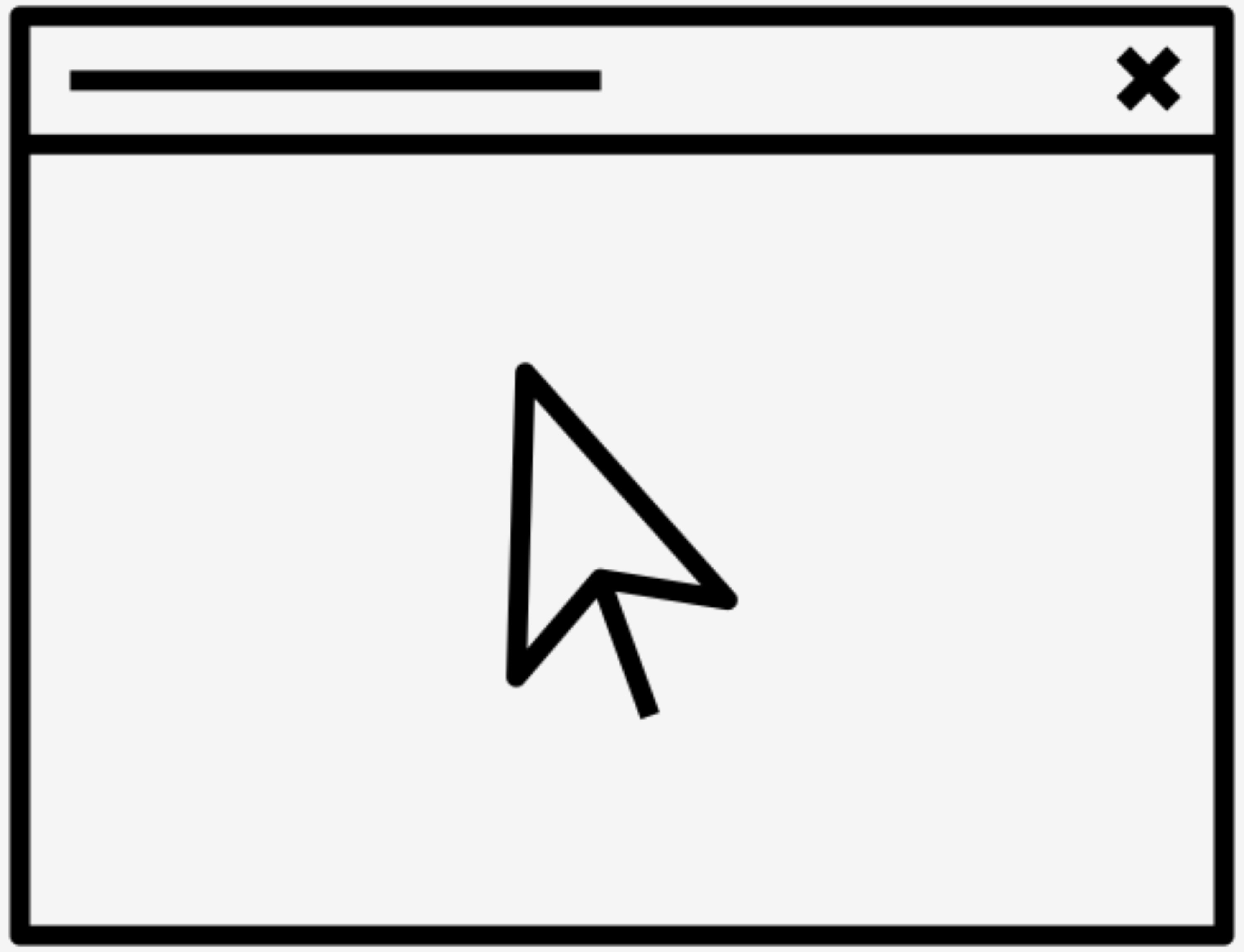


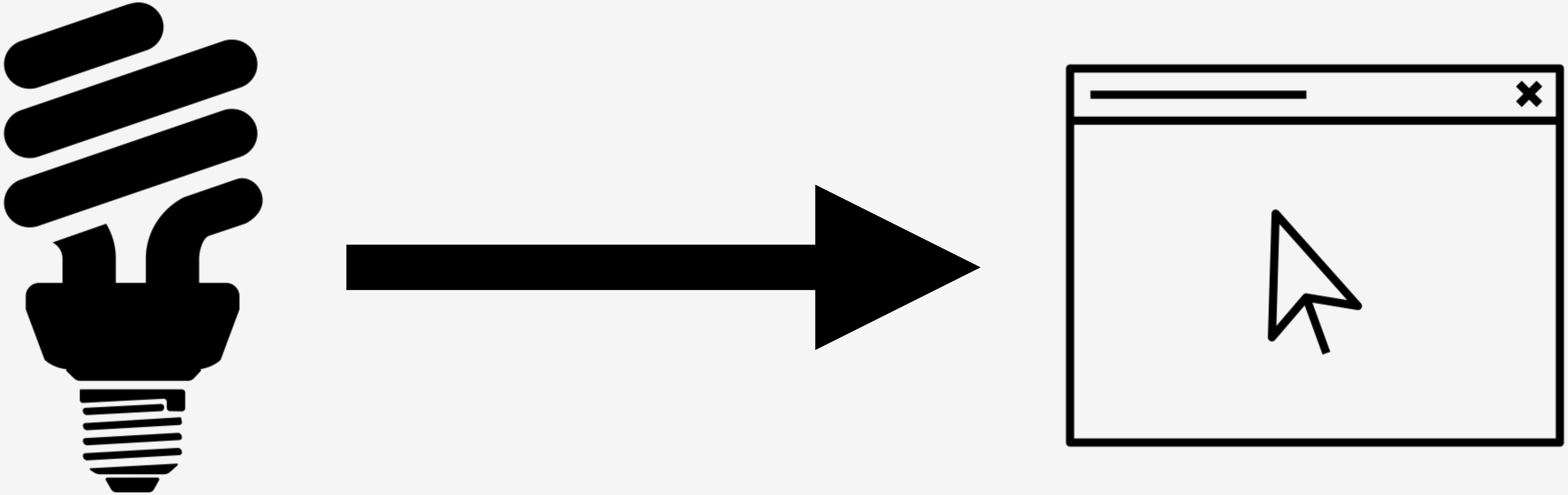
```
/*
 * Here is where we import all of our dependancies. The first line
 * dotenv and the loads all of the environment varibles specified
 * The next three lines are importing node modules that we install
 *
 * Express acts as our webserver. It helps us render HTML to the u
 * Express-handlebars is our templating engine. We don't want stat
 * rather we want to render handlebar-markup into HTML. This lets
 * variables have true control flow in our views.
 * Mongoose is how we interface with our database, MongoDB.
 */
require('dotenv').load();
var express = require('express');
var exphbs = require('express-handlebars');
var mongoose = require('mongoose');


/*
 * Connect to MongoDB with the database specified in our .env file
 */
mongoose.connect(process.env.MONGODB);

/*
 * Create an instance of our express webserver, and set the port.
 */
var app = express();
app.set('port', (process.env.PORT || 8080));

/*
 * Tell express which view engine we're using, and tell handlebars
 * main layout (the part that is rendered for every page) is.
 */
app.set('views', 'src/views')
app.engine('handlebars', exphbs({
```

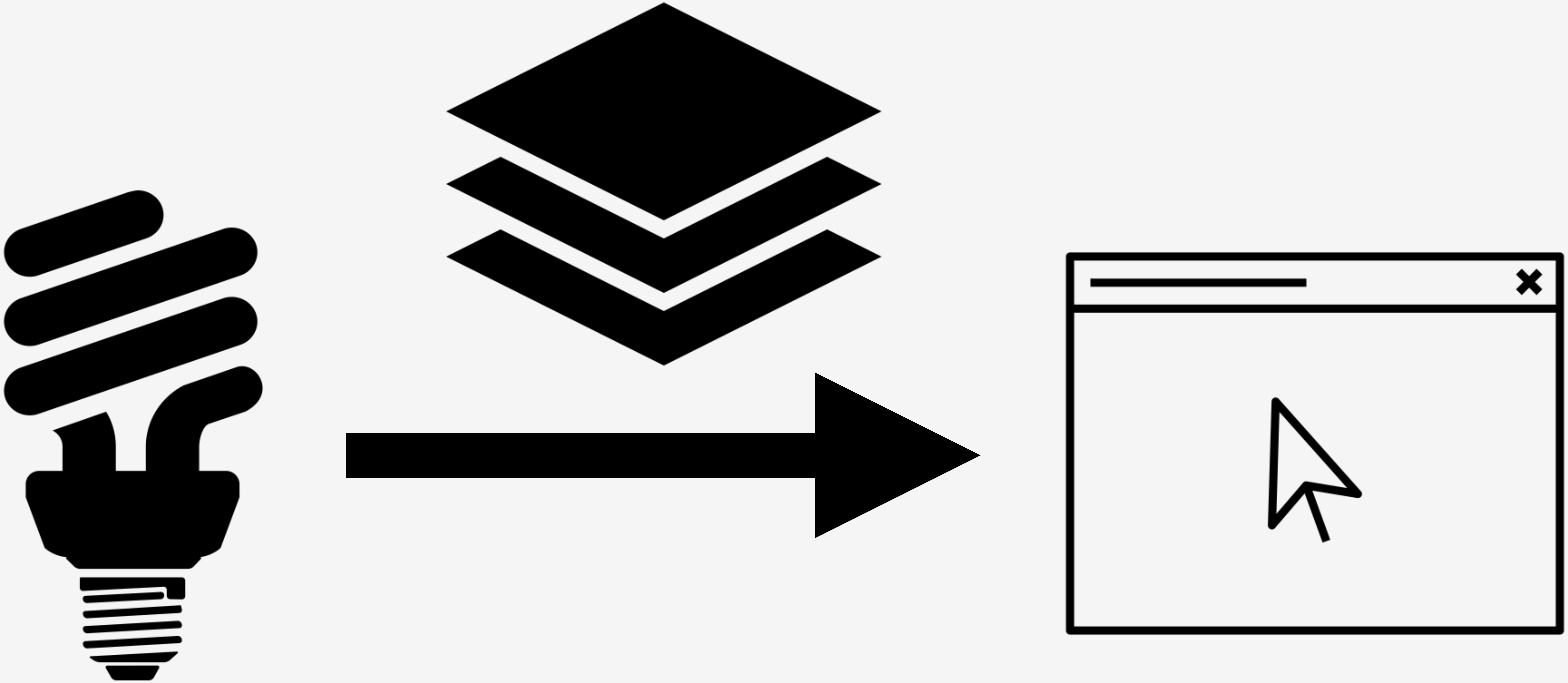
```
bower install       jquery#2.1.4

bootstrap#3.3.5 bower_components/bootstrap
 jquery#2.1.4

font-awesome#4.4.0 bower_components/font-awesome

jquery#2.1.4 bower_components/jquery
{ [Error: ENOENT, no such file or directory '.env'] errno: 34, code: 'ENOENT', pat
h: '.env', syscall: 'open' }
the server is listening on port 8080

/home/kirbyk/how-to-webdev/node_modules/mongoose/node_modules/mongodb/lib/server.j
s:228
        process.nextTick(function() { throw err; })
                                            ^
Error: getaddrinfo ENOTFOUND
    at errnoException (dns.js:37:11)
    at Object.onanswer [as oncomplete] (dns.js:124:16)

npm ERR! how-to-webdev@1.0.0 start: `npm run prepare && node server.js`
npm ERR! Exit status 8
npm ERR!
npm ERR! Failed at the how-to-webdev@1.0.0 start script.
npm ERR! This is most likely a problem with the how-to-webdev package,
npm ERR! not with npm itself.
npm ERR! Tell the author that this fails on your system:
npm ERR!     npm run prepare && node server.js
npm ERR! You can get their info via:
npm ERR!     npm owner ls how-to-webdev
npm ERR! There is likely additional logging output above.
npm ERR! System Linux 3.13.0-29-generic
npm ERR! command "/usr/local/bin/node" "/usr/local/bin/npm" "start"
npm ERR! cwd /home/kirbyk/how-to-webdev
npm ERR! node -v v0.10.26
npm ERR! npm -v 1.4.3
npm ERR! code ELIFECYCLE
npm ERR!
npm ERR! Additional logging details can be found in:
npm ERR!     /home/kirbyk/how-to-webdev/npm-debug.log
npm ERR! not ok code 0
kirbyk: ~/how-to-webdev $ ^C
kirbyk: ~/how-to-webdev $ cp env.sample .env
kirbyk: ~/how-to-webdev $ 
```

Getting from an Idea to a Product

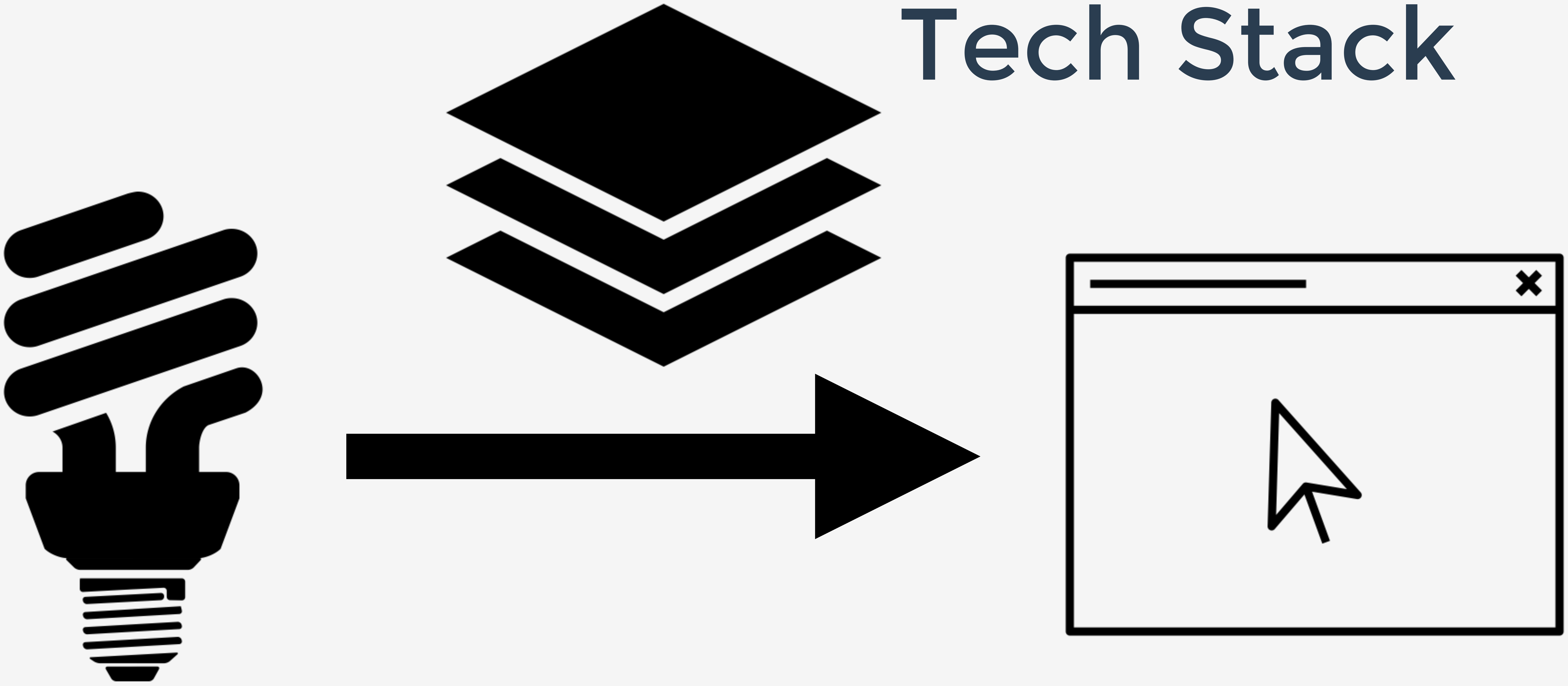
Ideas are easy.
Implementation is hard.







Tech Stack



What’s a tech stack?

A tech (technology) stack is all of the technologies that make up our application.

- What devices will the application run on?
- Will my application need a server?
- Will I need to store data somewhere?

What devices will the application run on?

Desktop and mobile.

What devices will the application run on?

Desktop and mobile.

(HTML / CSS)

Will my application need a server?

Yup.

Will my application need a server?

Yup.

(Node.js / Express.js)

Will I need to store data somewhere?

Yup.

Will I need to store data somewhere?

Yup.

(MongoDB)

Frontend versus Backend

Frontend

Frontend

Anything that the user **can** see and interact with.

Frontend

- HTML
- CSS
- JavaScript

Backend

Backend

Anything that the user **can’t** see and interact with.

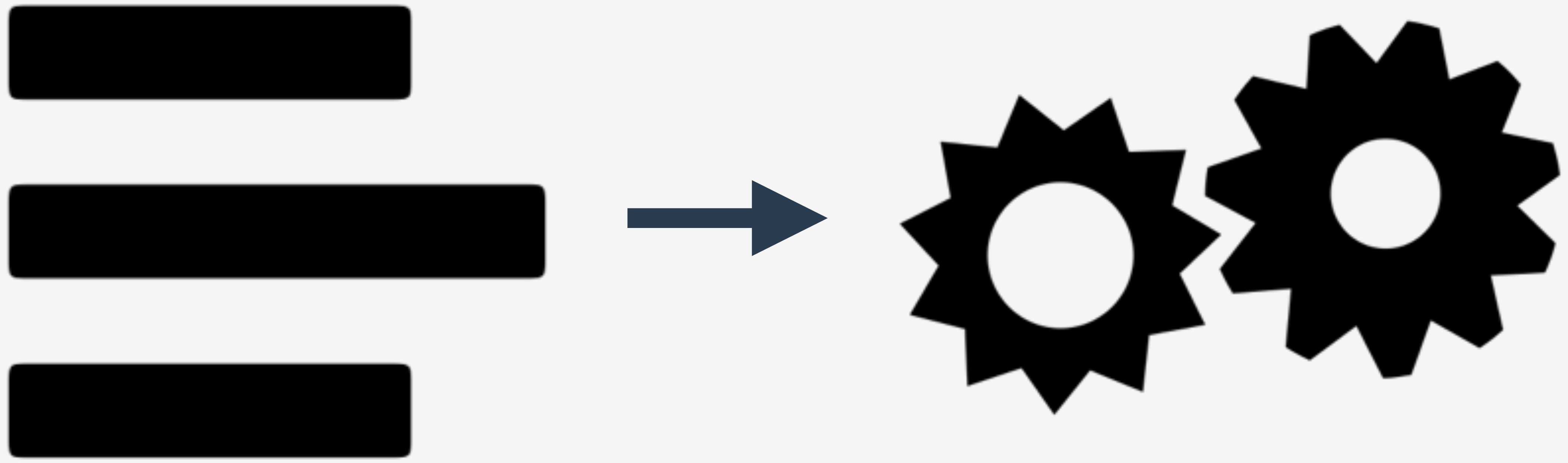
Backend

- Ruby on Rails
- PHP
- Node.js
- Java
- Django

How Internet Browsers Work

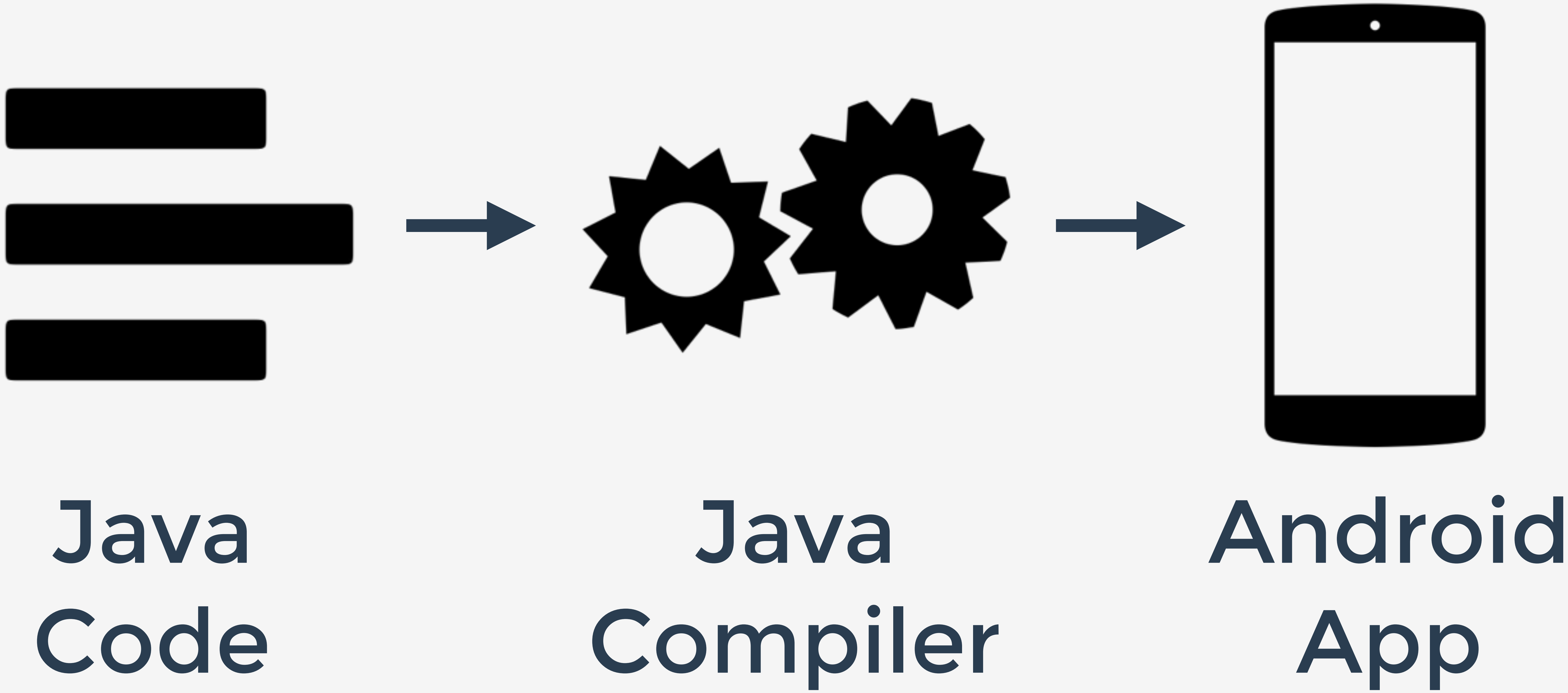
Text → Magic → Product

Java
Code



Java Code

Java Compiler

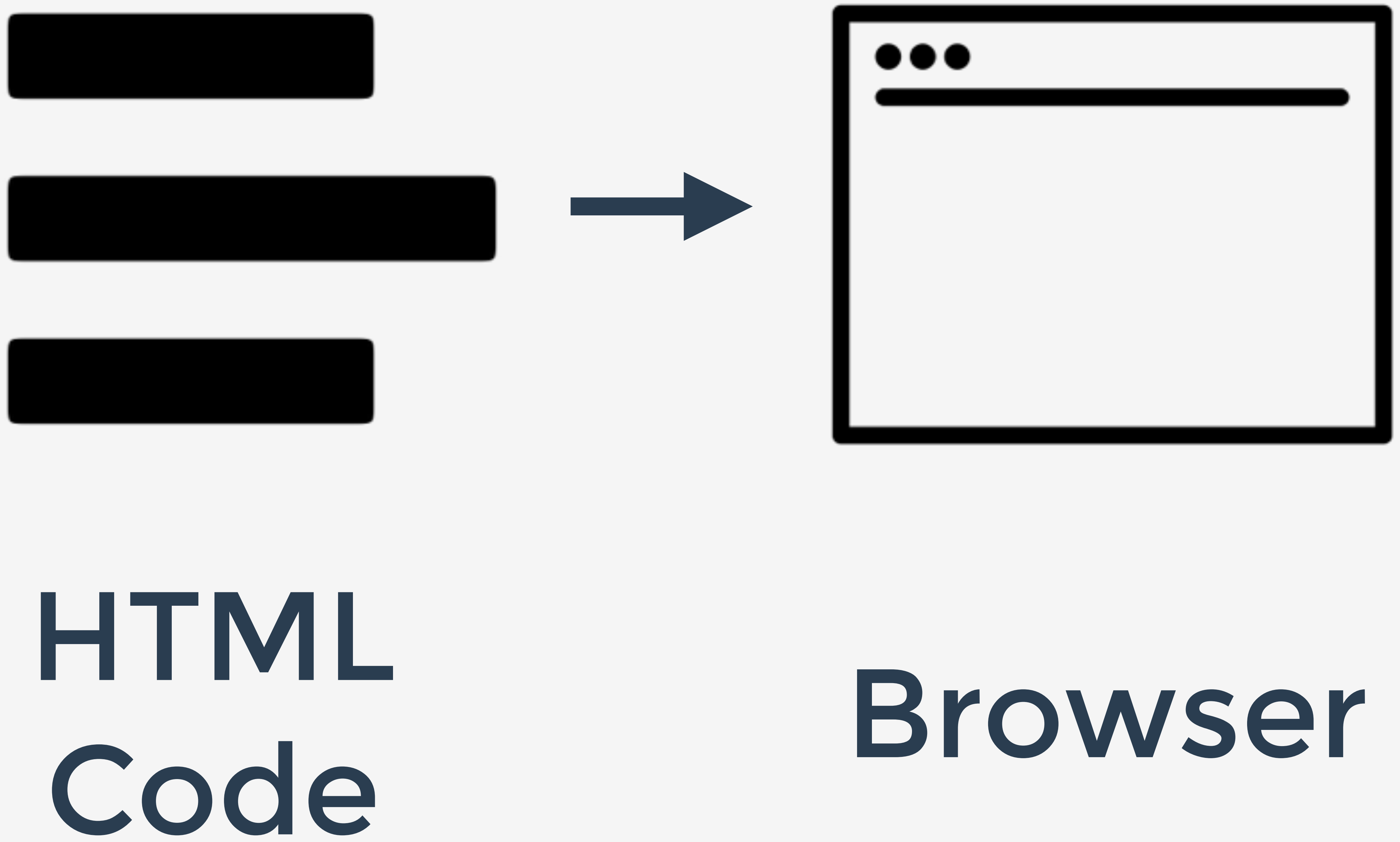


Java Code

Java Compiler

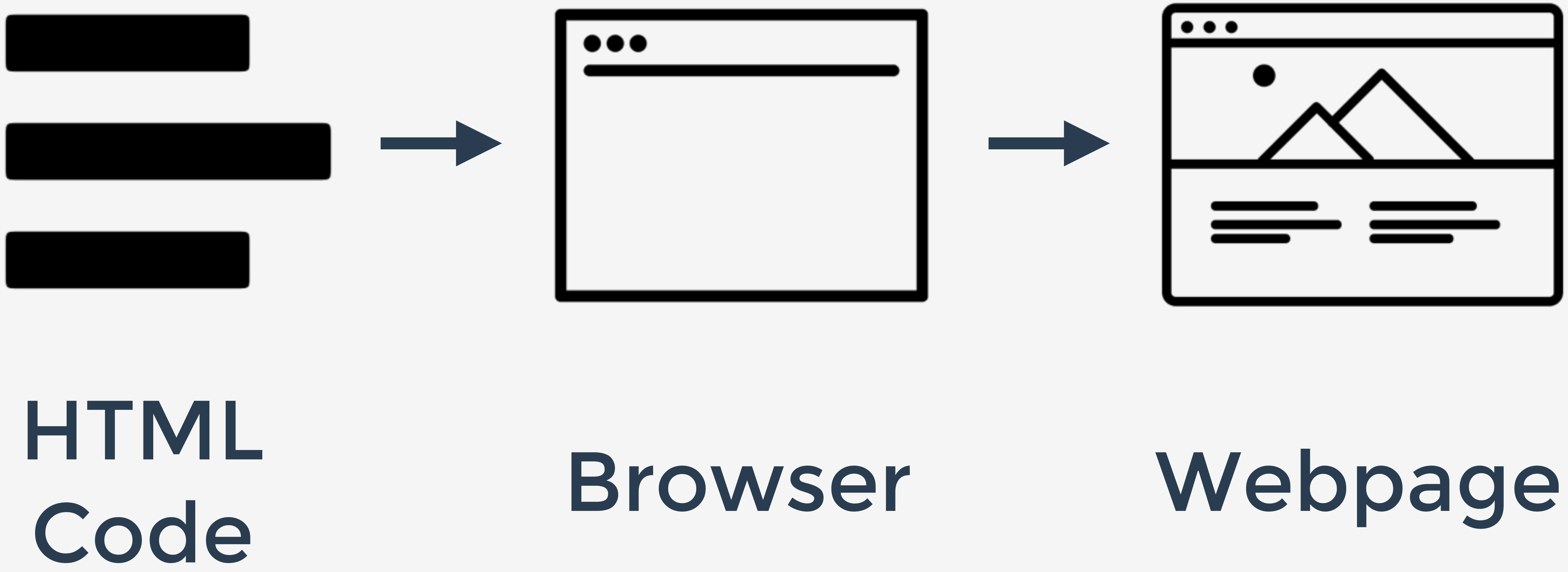
Android App

HTML
Code



HTML Code

Browser



HTML Code

Browser

Webpage

Webpage, Website, Webapp

Webpage

A single web document.

Website

A collection of webpages.

Webapp

A website with a backend.

Recap

- The Plan
- Our Environment
- Getting from an Idea to a Product
- How Internet Browsers Work

Questions?

HTML / CSS

What are HTML & CSS?

What is HTML?

HyperText Markup Language.

HyperText Markup Language.

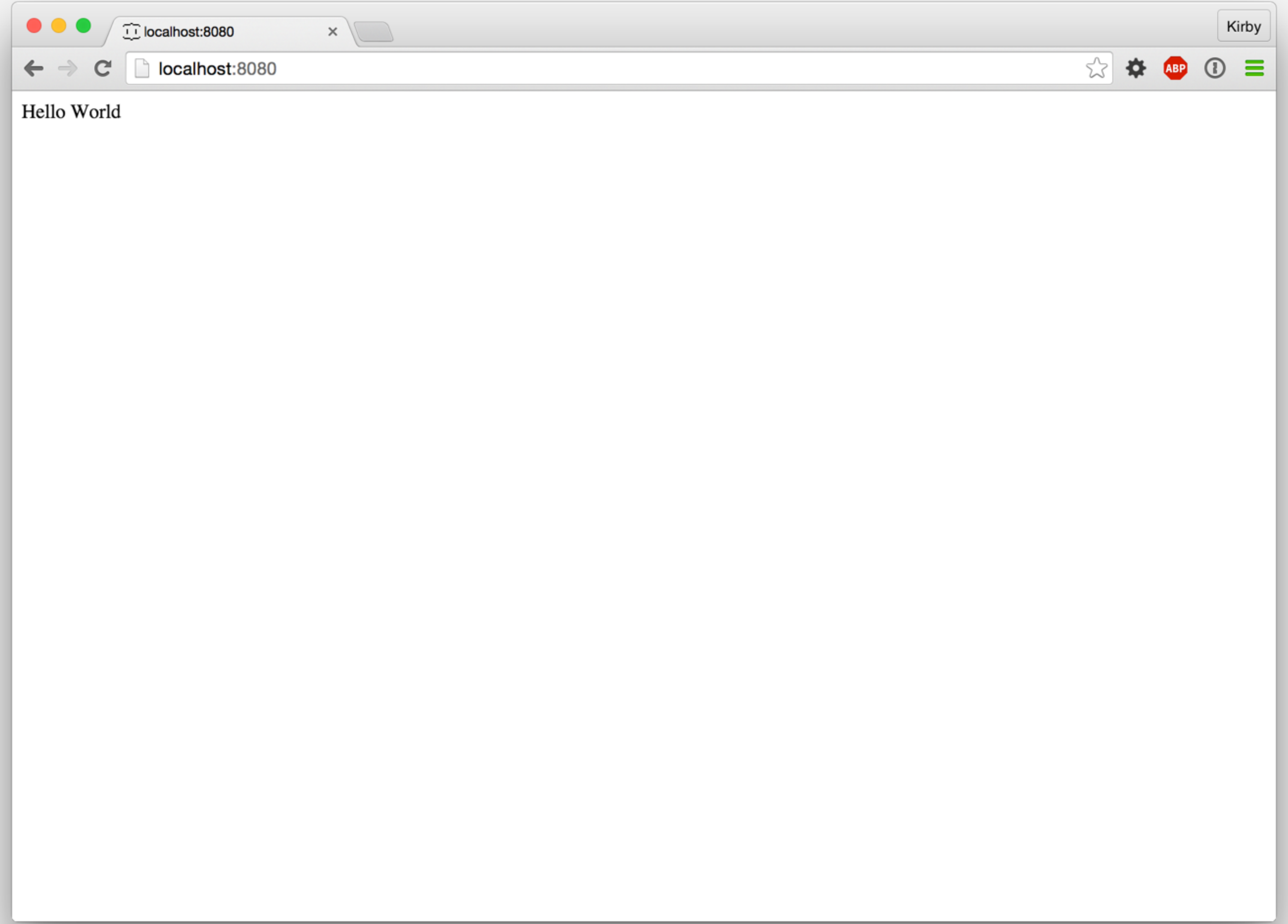
Markup?

HTML makes up all of the content of a webpage.

HTML is not a programming language.

Hello World in HTML

```
<!DOCTYPE html>
<html lang='en'>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <p>Hello World</p>
  </body>
</html>
```



Hello World

Breaking Hello World Down

Tags

```html
<!DOCTYPE html>
<html lang='en'>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <p>Hello World</p>
  </body>
</html>
```

```html
<!DOCTYPE html>
<html lang='en'>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <p>Hello World</p>
  </body>
</html>
```

Attributes

```html
<!DOCTYPE html>
<html lang='en'>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <p>Hello World</p>
  </body>
</html>
```

Common Tags

- html
- head
- body
- p, h1, h2, h3, h4
- div

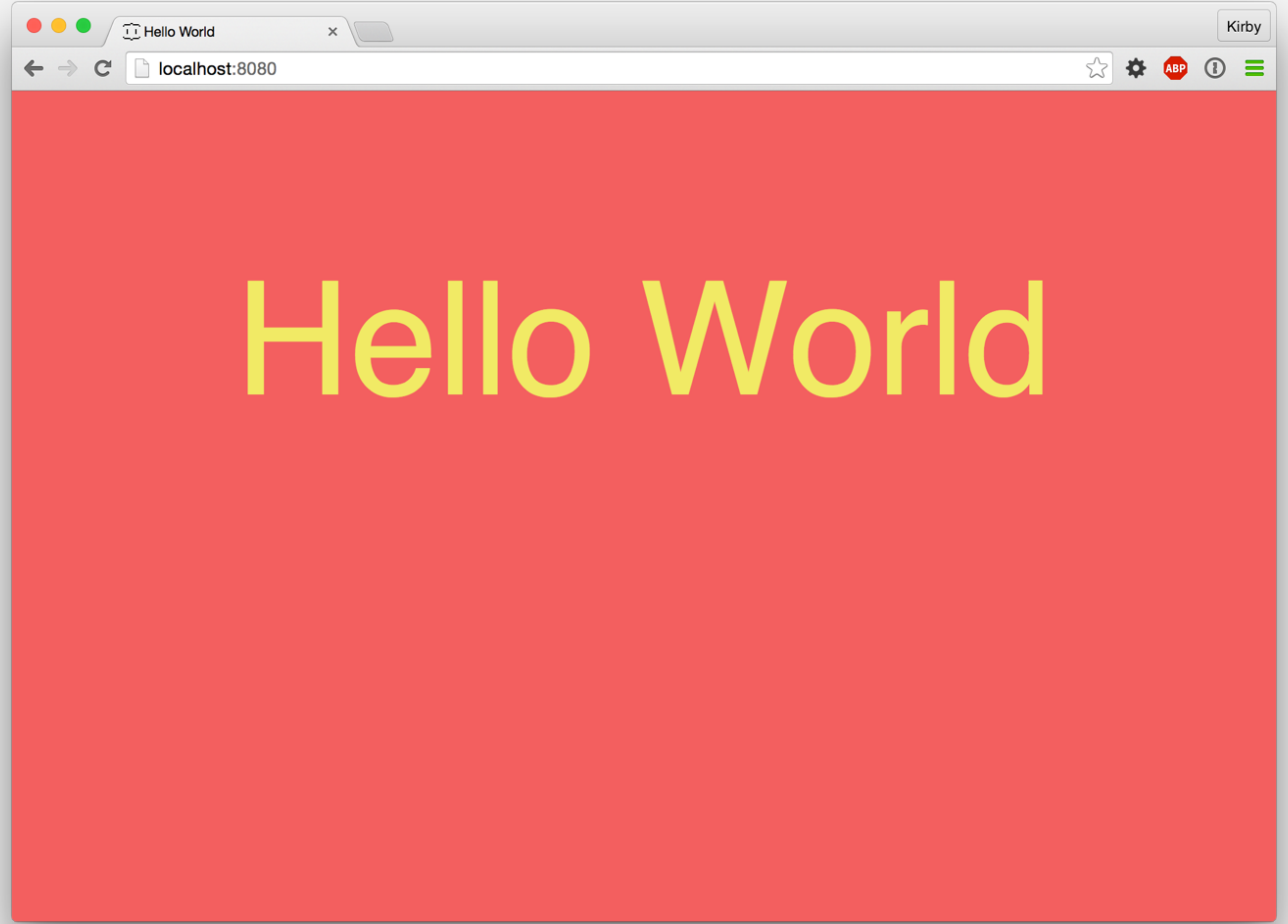
What is CSS?

Cascading Style Sheets.

CSS is the language used to style the content of a webpage.

Beautifying Hello World with CSS

```
<!DOCTYPE html>
<html lang='en'>
  <head>
    <title>Hello World</title>
    <style>
      body {
        background-color: #F35F5F;
        text-align: center;
      }

      .hello-world {
        color: #F1EA65;
        font-size: 8em;
        font-family: 'Helvetica'
      }
    </style>
  </head>
  <body>
    <p class='hello-world'>Hello World</p>
  </body>
</html>
```



Hello World

Breaking Beautified Hello World Down

Selectors

```
<!DOCTYPE html>
<html lang='en'>
  <head>
    <title>Hello World</title>
    <style>
      body {
        background-color: #F35F5F;
        text-align: center;
      }

      .hello-world {
        color: #F1EA65;
        font-size: 8em;
        font-family: 'Helvetica'
      }
    </style>
  </head>
  <body>
    <p class='hello-world'>Hello World</p>
  </body>
</html>
```

Properties

```
<!DOCTYPE html>
<html lang='en'>
  <head>
    <title>Hello World</title>
    <style>
      body {
        background-color: #F35F5F;
        text-align: center;
      }

      .hello-world {
        color: #F1EA65;
        font-size: 8em;
        font-family: 'Helvetica'
      }
    </style>
  </head>
  <body>
    <p class='hello-world'>Hello World</p>
  </body>
</html>
```

Classes

```html
<!DOCTYPE html>
<html lang='en'>
  <head>
    <title>Hello World</title>
    <style>
      body {
        background-color: #F35F5F;
        text-align: center;
      }

      .hello-world {
        color: #F1EA65;
        font-size: 8em;
        font-family: 'Helvetica'
      }
    </style>
  </head>
  <body>
    <p class='hello-world'>Hello World</p>
  </body>
</html>
```

Separating CSS from HTML

```html
<!DOCTYPE html>
<html lang='en'>
  <head>
    <title>Hello World</title>
    <link rel='stylesheet' href='style.css'>
  </head>
  <body>
    <p class='hello-world'>Hello World</p>
  </body>
</html>
```

index.html

```css
body {
  background-color: #F35F5F;
  text-align: center;
}

.hello-world {
  color: #F1EA65;
  font-size: 8em;
  font-family: 'Helvetica'
}
```

style.css

Recap

- HTML is content
- CSS is style

Questions?

Client-Side JavaScript

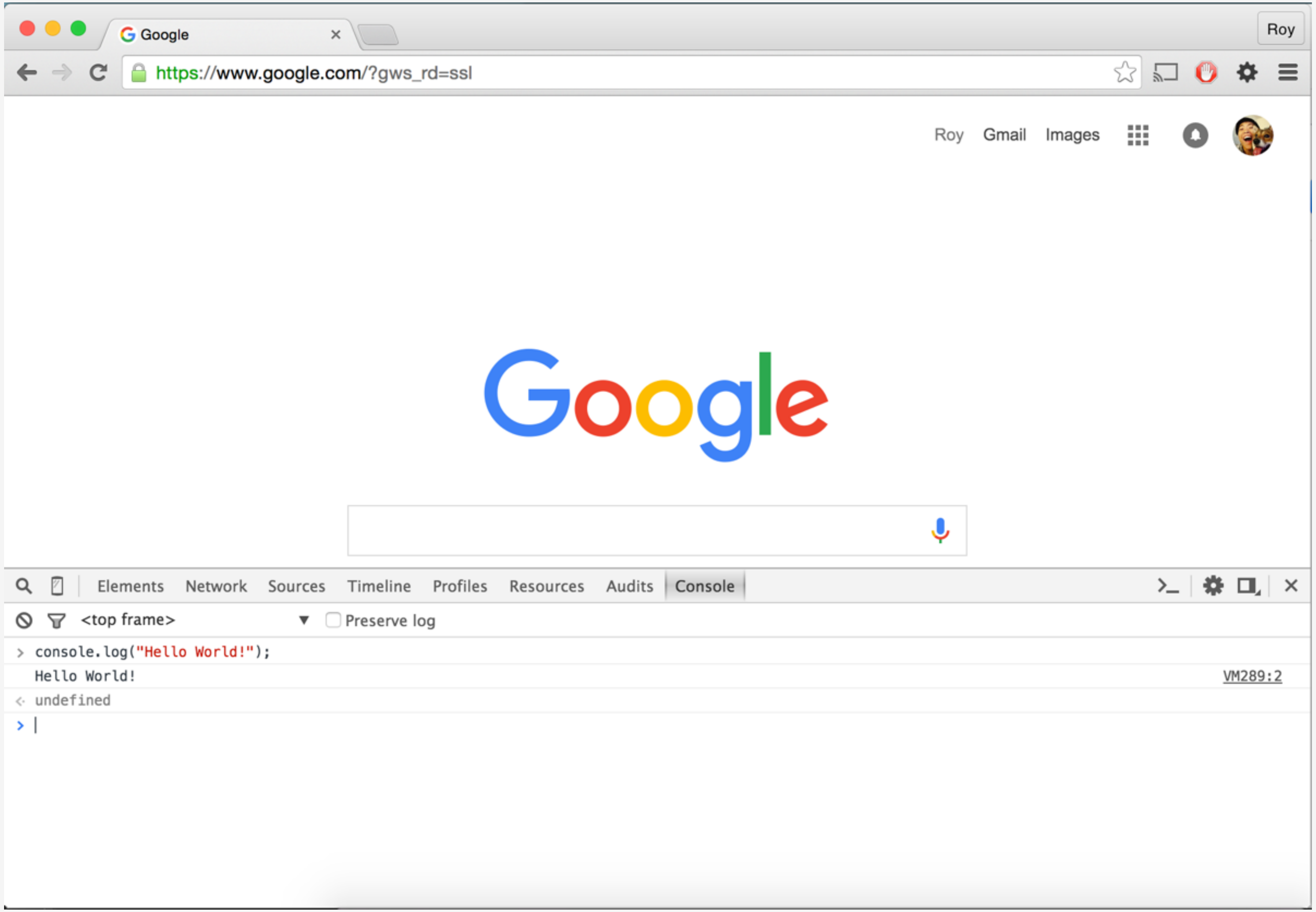
What is JavaScript?

JavaScript is a programming language

JavaScript is a programming language that runs in the browser

Hello World in JavaScript

```
console.log("Hello World!");
```



```
> console.log("Hello World!");
  Hello World!                                          VM289:2
<· undefined
> |
```

Breaking Hello World Down

Using a function called console.log

Similar to System.out or printf

Variables

Variables store information

JavaScript variables are special

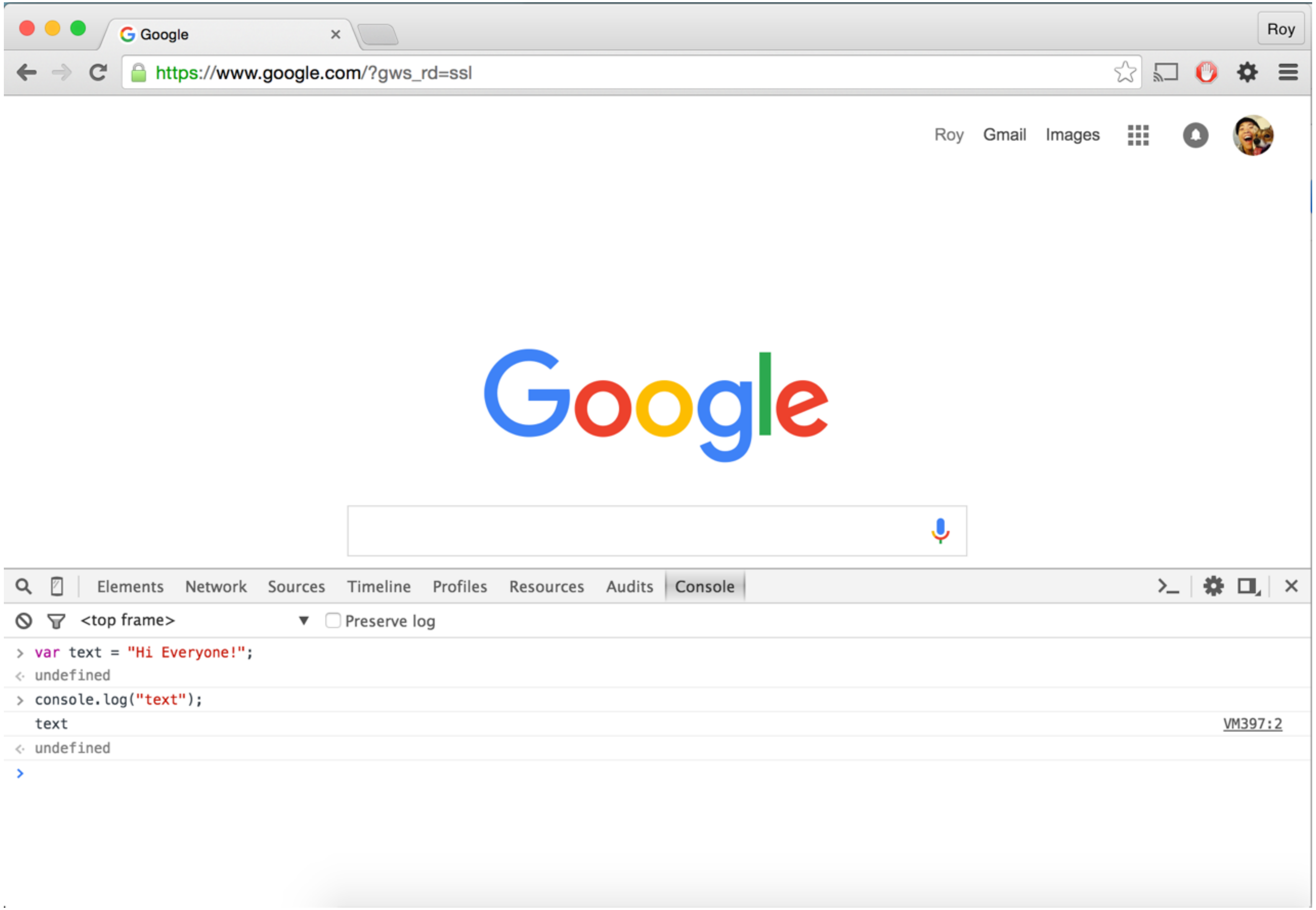
Instead of declaring ints, floats, strings, etc.
JavaScript infers everything at runtime

Everything is declared with var

Back to Hello World

```
var text = "Hi Everyone!";
console.log(text);
```

```
var text = "Hi Everyone!";
console.log(text);
```



```
> var text = "Hi Everyone!";
< undefined
> console.log("text");
  text                                                     VM397:2
< undefined
>
```

Functions

Functions encapsulate the logic

Functions have 3 parts:
Arguments, Body, Return

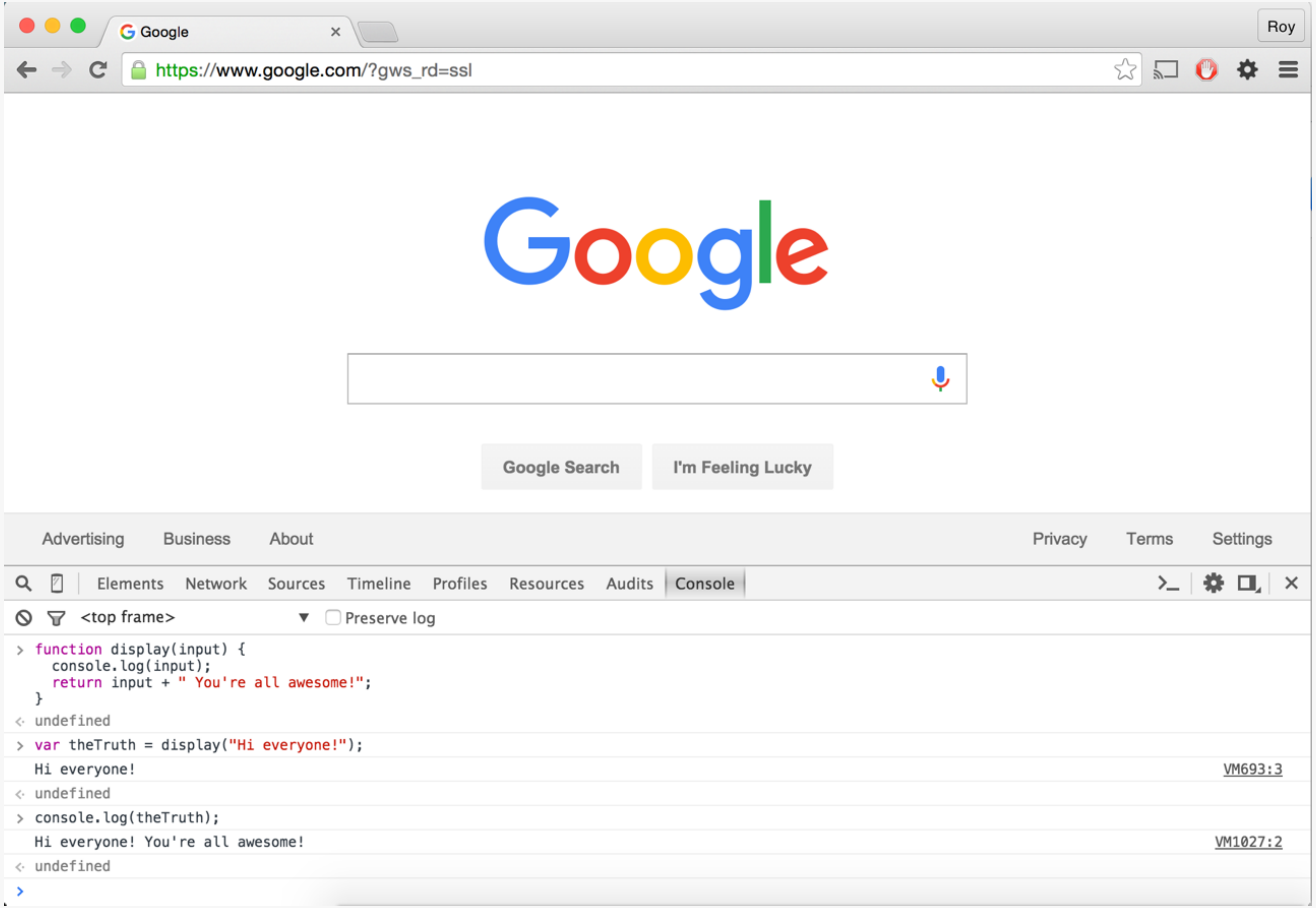
Arguments are the inputs

Body is the logic

Return is the output

Back to Back to Hello World

```
function display(input) {
  console.log(input);
  return input + " You're all awesome!";
}
var theTruth = display("Hi everyone!");
console.log(theTruth);
```



```
> function display(input) {
    console.log(input);
    return input + " You're all awesome!";
  }
< undefined
> var theTruth = display("Hi everyone!");
  Hi everyone!                                        VM693:3
< undefined
> console.log(theTruth);
  Hi everyone! You're all awesome!                   VM1027:2
< undefined
>
```

```
function display(input) {
  console.log(input);
  return input + " You're all awesome!";
}
var theTruth = display("Hi everyone!");
console.log(theTruth);
```

```
function display(input) {
  console.log(input);
  return input + " You're all awesome!";
}
var theTruth = display("Hi everyone!");
console.log(theTruth);
```

```
function display(input) {
  console.log(input);
  return input + " You're all awesome!";
}
var theTruth = display("Hi everyone!");
console.log(theTruth);
```

But how can we use this with our website?

What is jQuery?

Let’s JavaScript manipulate the DOM

What’s the DOM?

DOM is a model of all the HTML on the page

jQuery selects things in the DOM
(similar to CSS)

jQuery adds logic to those things

jQuery can add effects and it can add events

Back to Hello World

```html
<!DOCTYPE html>
<html lang='en'>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <p class='hello-world'>Hello World</p>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
    <script type="text/javascript">
      $('.hello-world').fadeOut(1000);
    </script>
  </body>
</html>
```

```
<!DOCTYPE html>
<html lang='en'>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <p class='hello-world'>Hello World</p>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
    <script type="text/javascript">
      $('.hello-world').fadeOut(1000);
    </script>
  </body>
</html>
```

```
<!DOCTYPE html>
<html lang='en'>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <p class='hello-world'>Hello World</p>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
    <script type="text/javascript">
      $('.hello-world').fadeOut(1000);
    </script>
  </body>
</html>
```

```
<!DOCTYPE html>
<html lang='en'>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <p class='hello-world'>Hello World</p>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
    <script type="text/javascript">
      $('.hello-world').fadeOut(1000);
    </script>
  </body>
</html>
```

Recap

- JavaScript is a programming language typically run on browsers
- jQuery is a library that lets us change things on the web page

Questions?

Node.js / Express.js

What are Node.js and Express.js?

What is Node.js?

Node.js is the only real dev language

Node.js is a wrapper around JavaScript

Instead of only running the browser,
JavaScript can now run outside the browser

Hello World in Node.js

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

http.createServer(function (request, response) {
    response.writeHead(200);
    response.end('Hello World\n');
}).listen(8080);
```

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

http.createServer(function (request, response) {
    response.writeHead(200);
    response.end('Hello World\n');
}).listen(8080);
```

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

http.createServer(function (request, response) {
    response.writeHead(200);
    response.end('Hello World\n');
}).listen(8080);
```

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

http.createServer(function (request, response) {
    response.writeHead(200);
    response.end('Hello World\n');
}).listen(8080);
```

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

http.createServer(function (request, response) {
    response.writeHead(200);
    response.end('Hello World\n');
}).listen(8080);
```

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

http.createServer(function (request, response) {
    response.writeHead(200);
    response.end('Hello World\n');
}).listen(8080);
```

Node.js also uses JSON

So what’s JSON?

JavaScript Object Notation

It’s a serialized version of a JavaScript object

We have keys and values

And the values of a key
can be another object

What is Express.js?

Express.js Framework for building web apps

Express.js takes care of the server set up,
so you can just focus on the logic

```
var express = require('express');
```

Hello World in Express.js

```
app.get('/', function (req, res) {
  res.send('Hello World!');
});
```

Routes

Routes are like roads for the internet

They handle the different URLs of your site

Routing example in Express.js

```
app.get('/', function(req, res) {
    res.sendfile('./views/index.html');
});
```

```
app.get('/', function(req, res) {
    res.sendfile('./views/index.html');
});
```

```
app.get('/', function(req, res) {
    res.sendfile('./views/index.html');
});
```

But what if we want to handle server logic in our HTML?

What is Handlebars?

Templating engine

Handles server logic in your HTML

Handlebars Example

```
<h1>All my cats</h1>
{{#each cats}}
  <p>{Cat}</p>
{{/each}}
```

```
<h1>All my cats</h1>
{{#each cats}}
  <p>{Cat}</p>
{{/each}}
```

```
<h1>All my cats</h1>
{{#each cats}}
  <p>{Cat}</p>
{{/each}}
```

Recap

- Node.js lets you run JavaScript outside of the browser
- Express.js is a Node.js framework for building web apps
- Handlebars lets us include server logic in the frontend

Questions?

MongoDB / MVC

What are MongoDB & MVC?

What is MongoDB?

MongoDB is a database

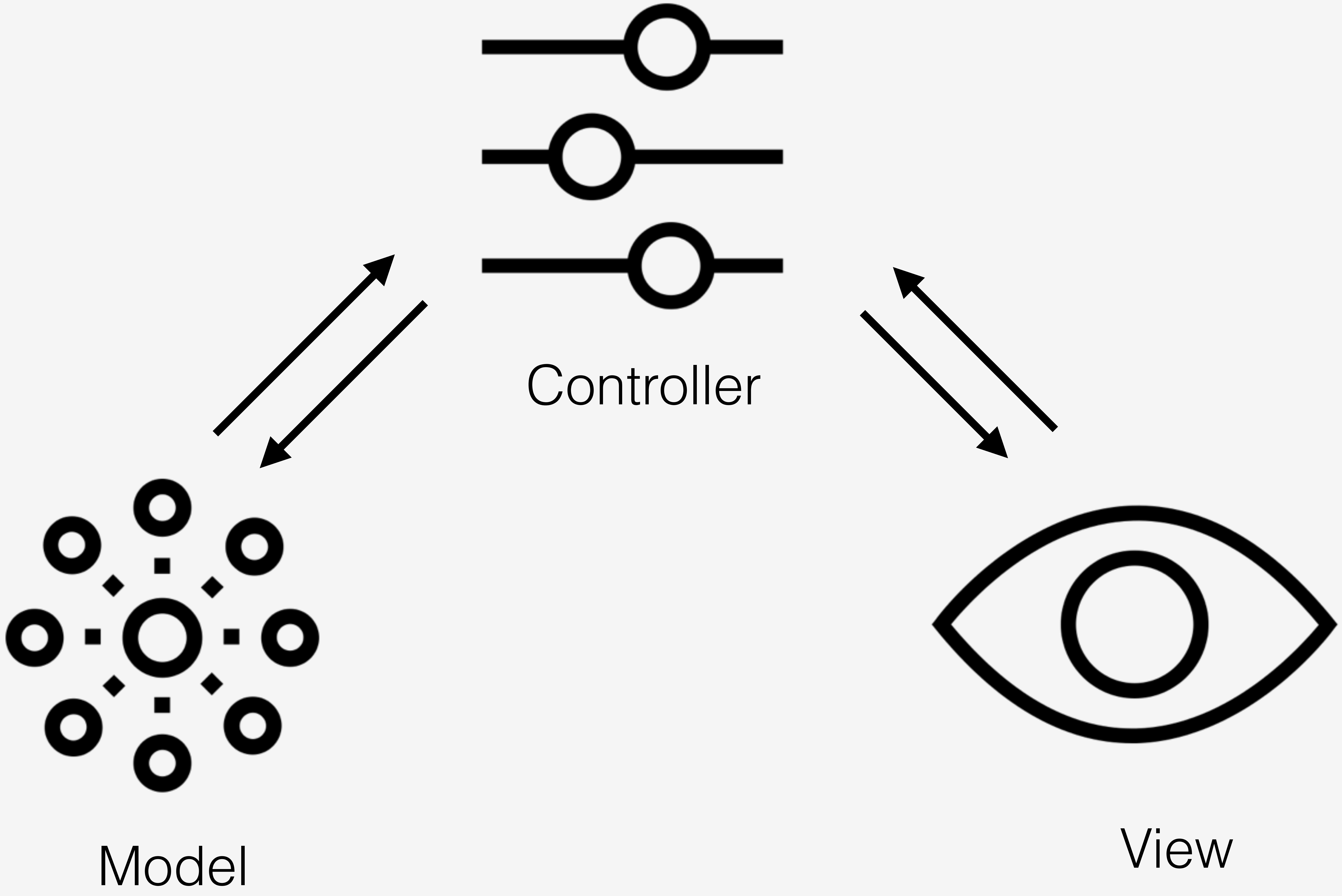
Let’s us store a ton of information in the form of JSON (BSON)

It’s a document based data store

Allows us to insert JSON, update it, and retrieve it later.

What is MVC?

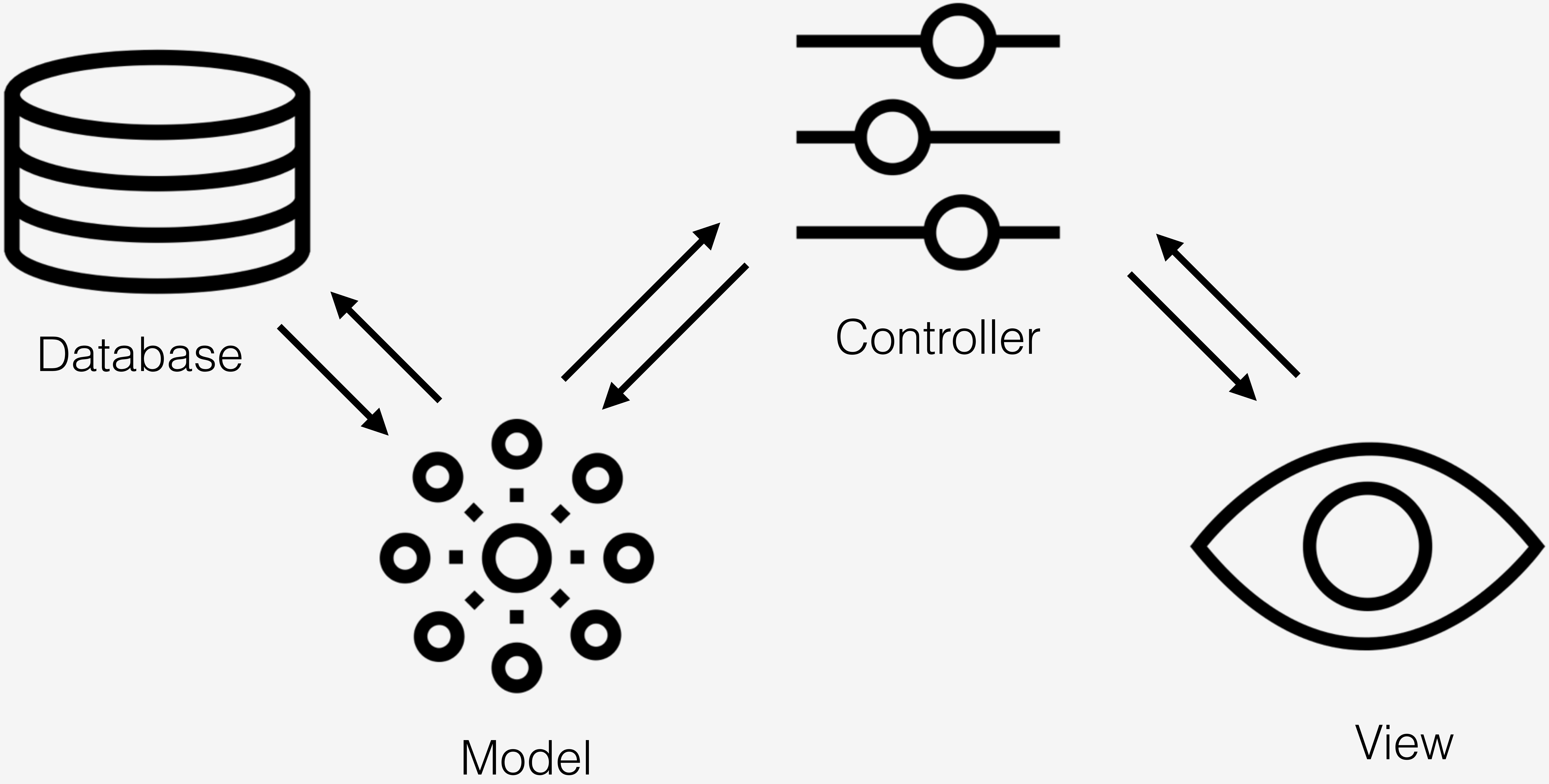
MVC is a way to organize and structure your code



Controller

Model

View



Database

Controller

Model

View

Recap

- MongoDB is a super JavaScript friendly database
- MVC is a great way to organize code

Questions?

Thanks!