

Welcome to your monthly

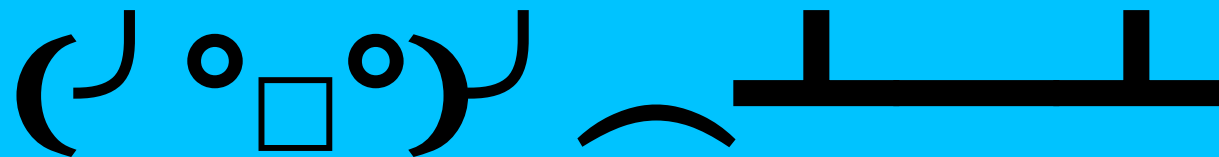
**DRUG**

Meetup

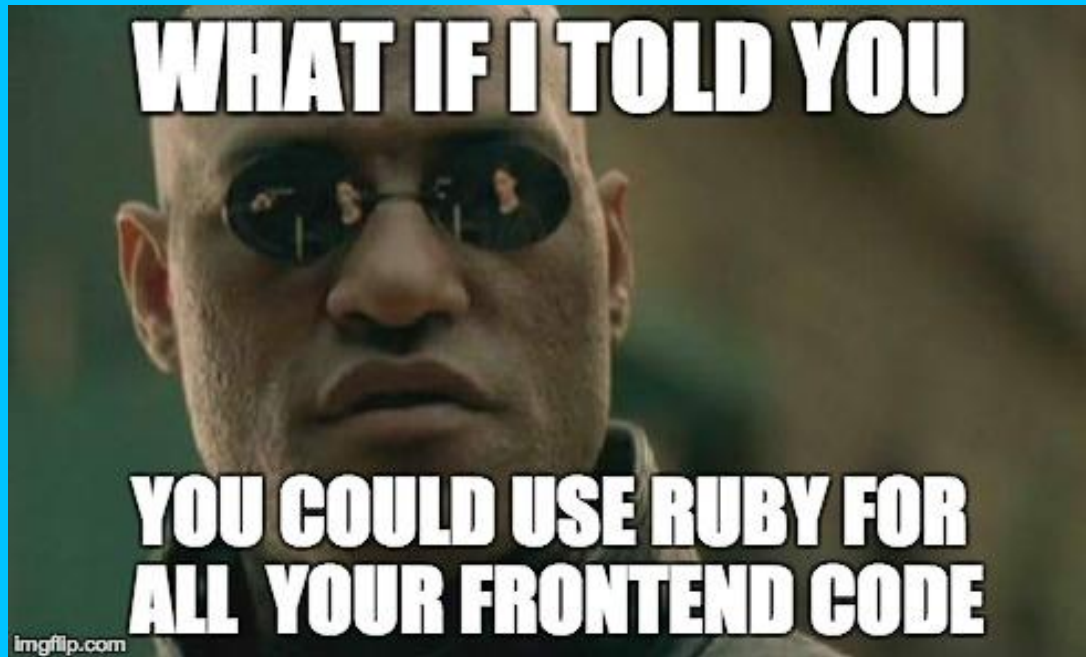
Presentation by:

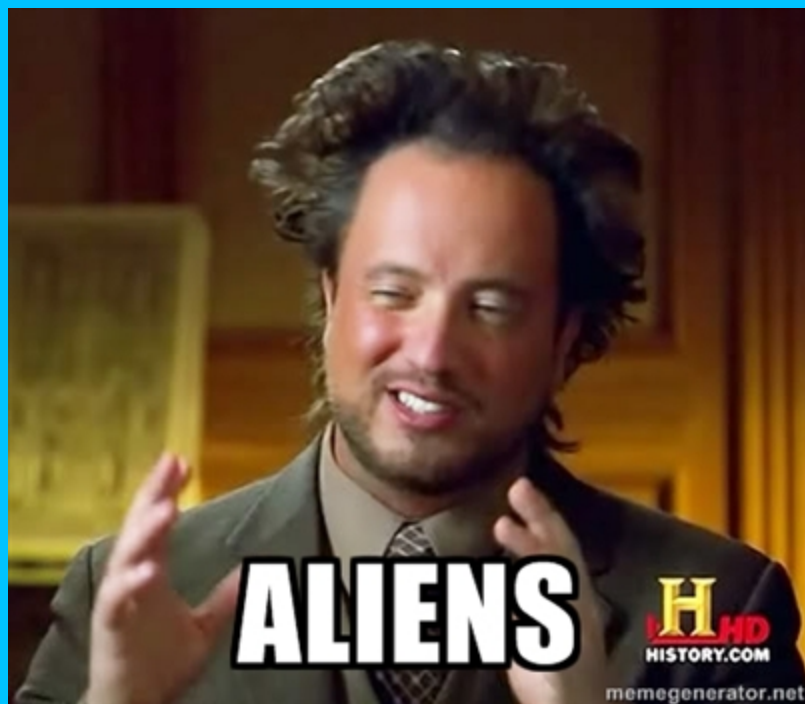
Cody Stringham & Quintin Adam

When I'm working  
with JS I be like...




# Calm down, Bro





Nope,  
introducing...

# Volt

  $\sim /(^{.^{.^{.}}}/)$

# What is Volt?

# D-Day

Introducing ManaVolt



# Getting Started

Here are some basics

# MongoDB

Currently, Volt is dependant on MongoDB.

install:

```
brew install mongodb
```

# Volt Commands

Volt commands are shockingly similar to Rails commands, no pun intended.

Volt Console:

```
volt c
```

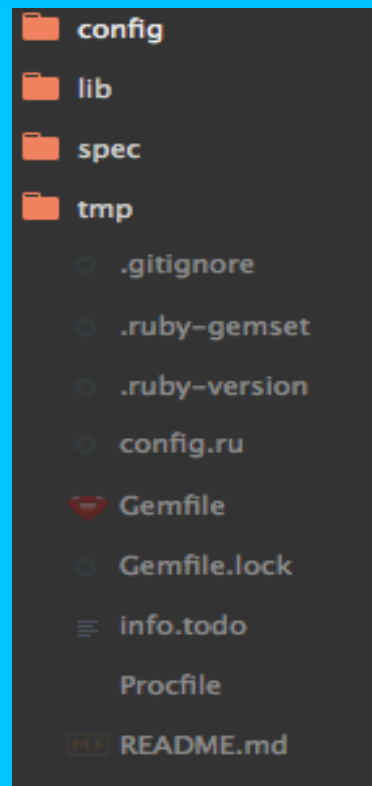
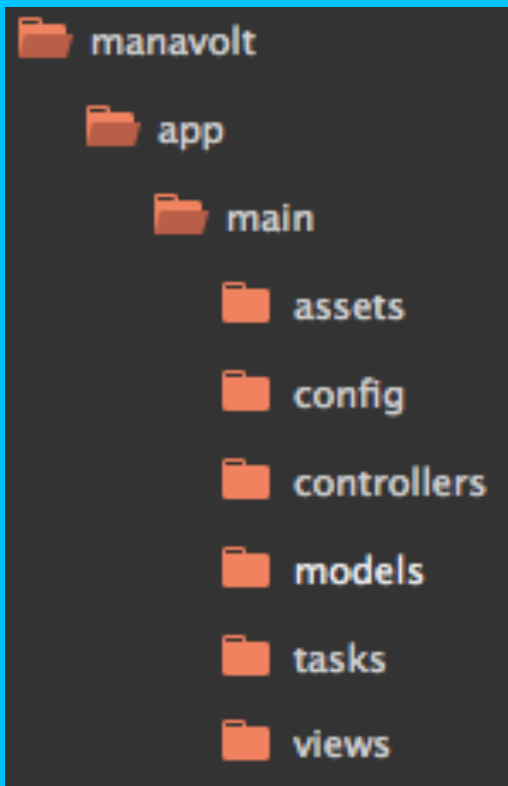
Volt Server:

```
volt s
```

New Project:

```
volt new app_name
```

# Folder Structure



# Templating Language

Sections:

```
<:Title>
  About

<:Body>
  <h1>About</h1>
```

Partials:

```
<:nav href="/about" text="About" />
<:nav href="/teams" text="Teams" />
```

```
<:Nav>
  <li class="{{ 'active' if active_tab? }}">
    <a href="{{ attrs.href }}">{{ attrs.text }}</a>
  </li>
```

# Content Binding

Ruby code in-between two fancy braces

```
{{ Ruby.new(code) }}
```

is evaluated

Conditional:

```
... {{ if daily.editing == true }}  
...   {{ page._task_name }}  
... {{ else }}  
...   {{ daily.name }}  
... {{ end }}
```

# Attribute Binding

Bindings can also be set inside of classes!

Class names:

```
.....<button
.....  class="{ { 'active' if current_difficulty(daily, 3) }}"
.....  e-click="set_daily_difficulty(daily, 3)">
.....  Hard
.....</button>
```

# Routes

```
# See https://github.com/voltrb/volt#routes for more info on routes
get '/about', _action: 'about'

# Teams Controller
get '/teams', _controller: 'teams', _action: 'index'
get '/teams/{__id}', _controller: 'teams', _action: 'show'

# Routes for login and signup, provided by user-templates component gem
get '/signup', _controller: 'user-templates', _action: 'signup'
get '/login', _controller: 'user-templates', _action: 'login'

# The main route, this should be last. It will match any params not
# previously matched.
get '/', {}
```



# Controllers

Controllers keep the peace, views and models never agree!

```
def add_team
  _teams << { name: page._new_team, level: 1, experience: 0, health: 50, mana: 0 }
  page._new_team = ''
end

def remove_team(team)
  _teams.delete(team)
end
```

# Default Models

Models provide “field” definitions and validations

Page Model:

```
<input  
  class="form-control"  
  type="text"  
  value="{{ page._task_name }}"  
  placeholder="New Daily Task"  
>
```

Store Model:

```
model :store
```

```
def save_daily(daily)  
  daily.name = page._task_name  
  daily.editing = false  
end
```

# Custom Models

Models provide “field” definitions and validations

```
class Team < Volt::Model
  field :name
  field :level
  field :experience
  field :health
  field :mana
end
```

# Coming Up...

Rails Integration (third-rail)

RethinkDB / PostgreSQL

REST Controller / Routes

HAML support

Error REPL

# Live Code

like a boss