PROGRAMMING FUNDAMENTALS

LESSON #1

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. BECOME A GOD

- Put it all together

GOALS FOR TODAY

- What is programming?
- What is Web Development?
- How is a Website made?
- Understand some key terminology
- The basics of HTML and CSS
- Mad HTML hacking session

WHAT IS PROGRAMMING?

PROGRAMMING

The process of writing instructions that can be understood by a computer.

PROGRAMMING LANGUAGE

An intermediary language which can be understood by both computers and by Human beings.

1837

Charles Babbage publishes a paper describing a "computing machine" called The Analytical Engine.

He was 100 years ahead of his time. And very grumpy...

1842

Ada Lovelace writes algorithms for the machine Babbage described, making her the first computer programmer.

1937

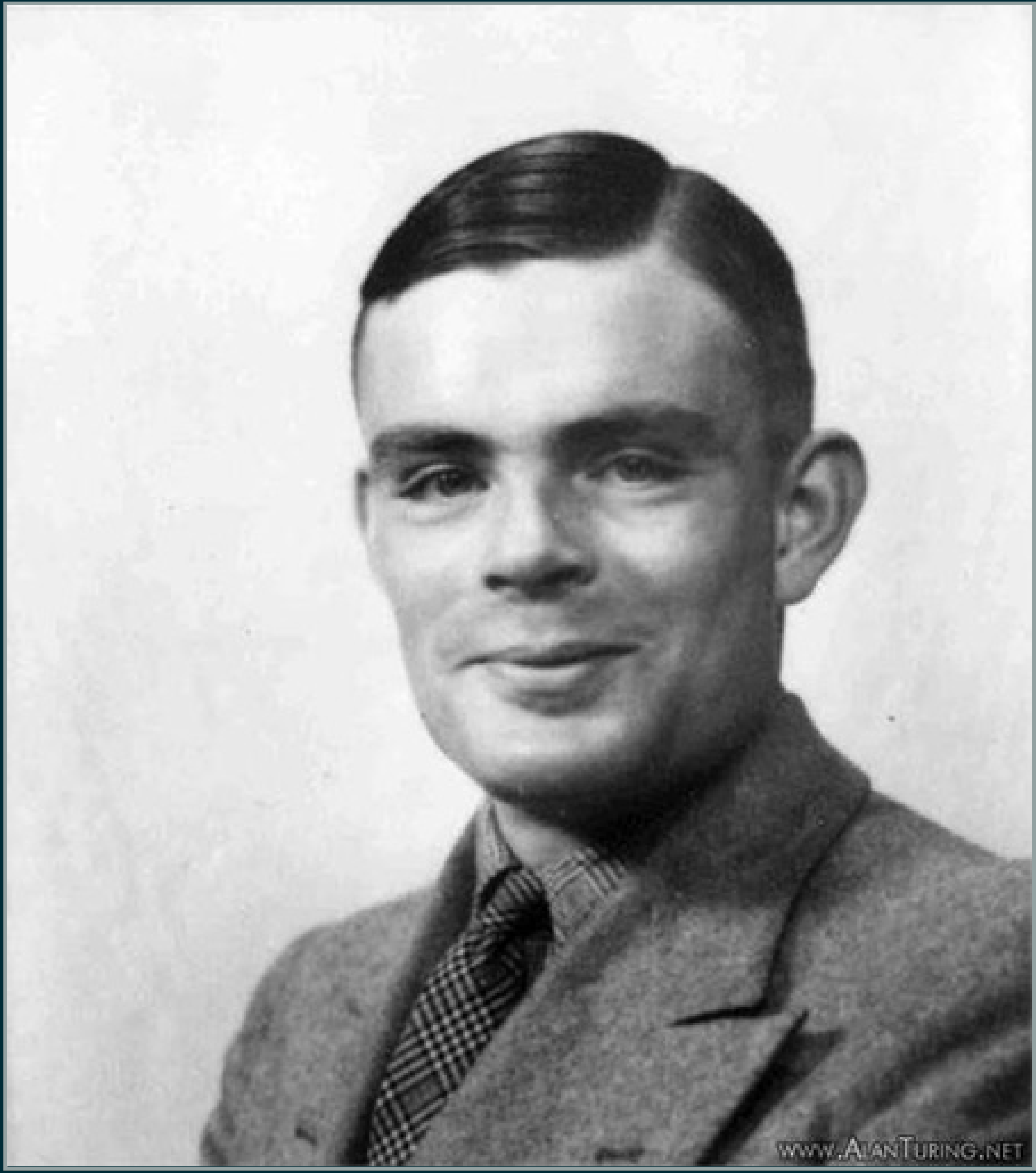
Alan Turing introduces the Turing Machine - a theoretical mathematical device that represents a computing machine.

He defines the notion of computation and opens the gates for the modern computer.

1946

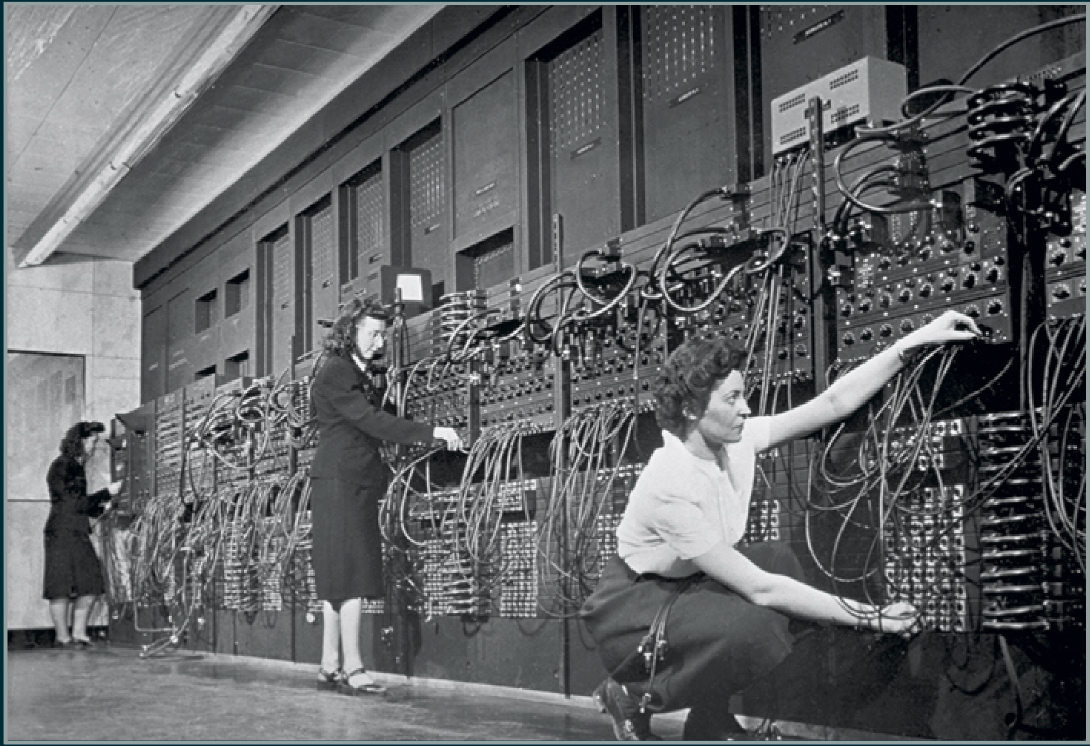
ENIAC, the first fully-programmable digital computer, was completed in Philadelphia.

It was **HUGE**.



1950'S

Grace Hopper and co. create FLOW-MATIC and COBOL, the first programming languages to use English-like instructions.



1971

Intel introduces the first microprocessor, the Intel 4004. Computers shrink drastically.

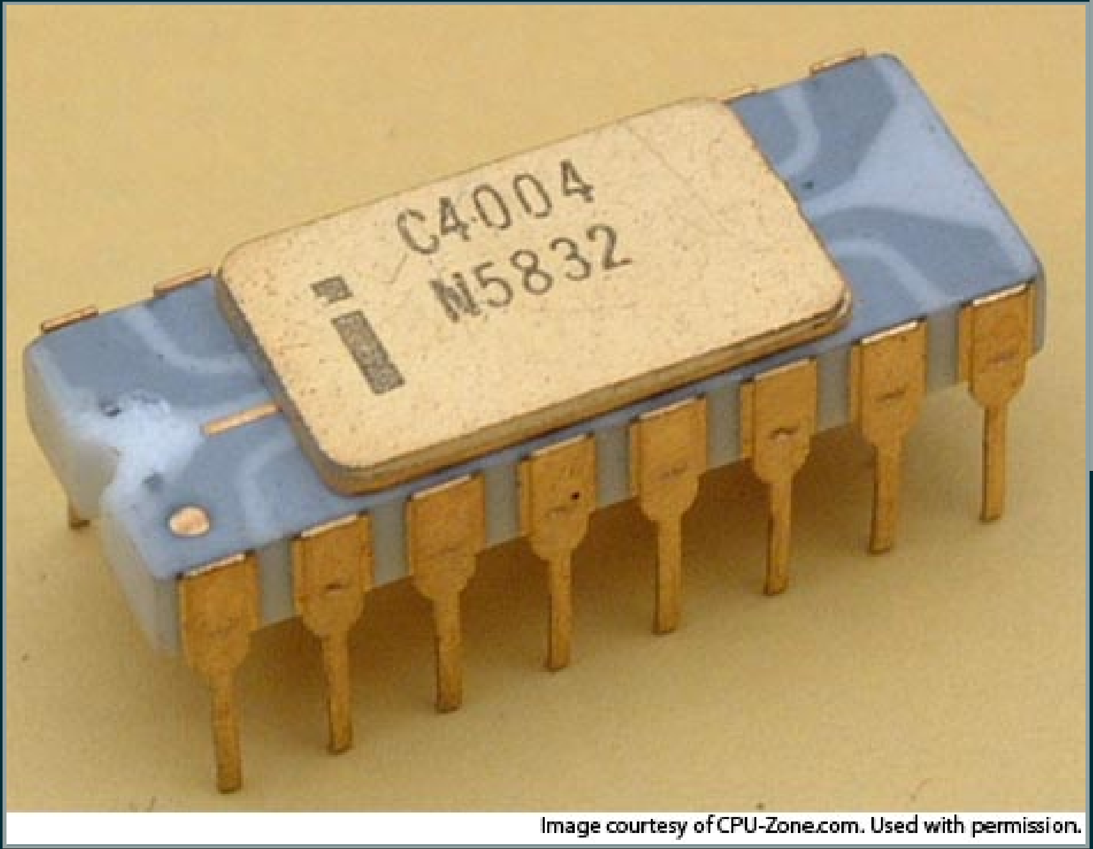


Image courtesy of CPU-Zone.com. Used with permission.

1982

Existing Computer-to-Computer communication protocols are standardised. The Internet is born.

1985

IBM, Apple and Microsoft dominate the emerging home computing industry.

1991

Tim Berners-Lee and his team, of Switzerland, develop HTTP, HTML, a web server and a web browser. The WWW is born.

WHAT CAN PROGRAMMING LANGUAGES DO?

NAME THINGS (VARIABLES)

```
name = "Andrew"
```

TEST THINGS (FORMAL LOGIC)

```
age = 29

age < 50 and age > 18
```

MAKE DECISIONS (BRANCHING)

```
if age !== 21
  print "Cool"
end
```

CHANGE THINGS (STATE MUTATION)

```
age = 29
age = age + 1
```

REPEAT OURSELVES (LOOPING)

```
while true
  print "Endless loop!"
end

print "I never get executed..."
```

FOLLOW THE RULES (TYPE SYSTEMS)

```
# No! Behave yourself, hacker!

name = "Andrew"
name = name + 1
```

HIDE THINGS (ABSTRACTION)

```
fn double_or_n(n)
  if n < 10
    return n
  else
    return n * 2
  end
end

print double_or_n(5)   # 5
print double_or_n(15)  # 30
```

STEAL THINGS (LIBRARIES)

```
require "datetime"

current_time = DateTime.now

print current_time.hour
```

BREAK THINGS (BUGS)

```
n = 0

if 10 / n > 1
  print "WTF?"
end
```

AND MUCH MORE!

WHAT IS WEB DEVELOPMENT?

WEB DEVELOPMENT

- The technical process of creating software that runs on the Web
- It's a broad term that covers several separate jobs!
- Typically, all of the non-design aspects of creating web applications

FRONT-END WEB DEVELOPMENT

- Translate designs into web pages
- Work primarily in the Web Browser
- Main tools are (typically) HTML, CSS and Javascript

BACK-END WEB DEVELOPMENT

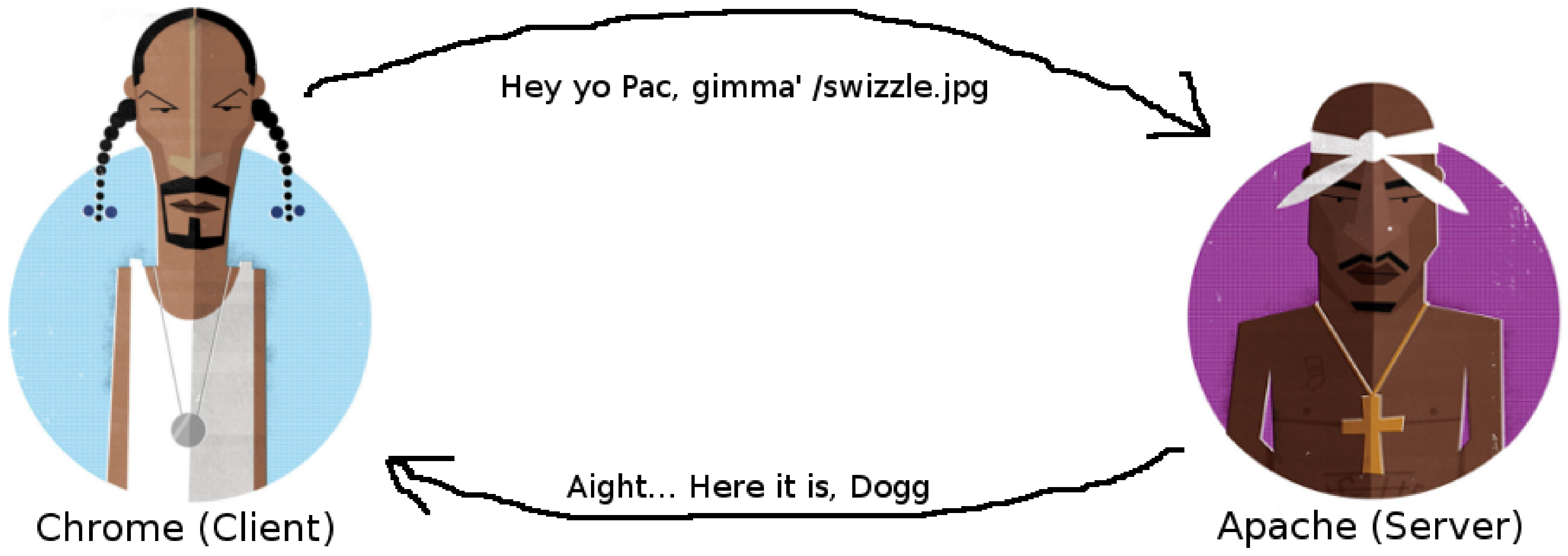
- Create software that provides Website functionality
- Work primarily on the server
- Main tool vary greatly

BEWARE THE FULL-STACK

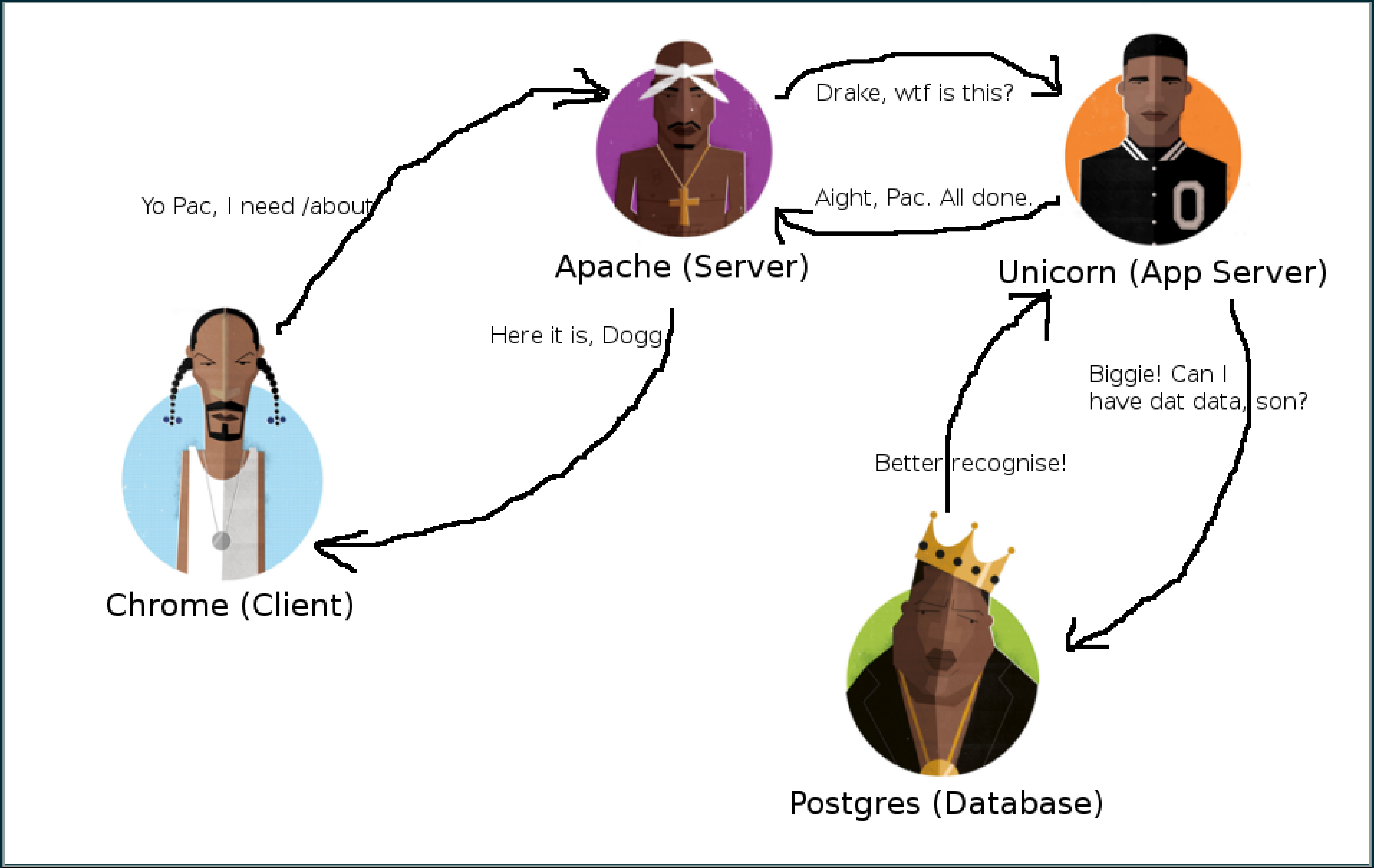
- You can't know everything, pick your niche!
- **Not good:** "Jack of all trades, master of none"
- **Good:** "Jack of some trades, master of one, delegated the rest"

HOW DOES THE WEB WORK?

STATIC REQUEST


Hey yo Pac, gimma' /swizzle.jpg
Aight... Here it is, Dogg
Chrome (Client)
Apache (Server)

DYNAMIC REQUEST


Drake, wtf is this?
Yo Pac, I need /about
Aight, Pac. All done.
Apache (Server)
Unicorn (App Server)
Here it is, Dogg.
Biggie! Can I have dat data, son?
Better recognise!
Chrome (Client)
Postgres (Database)

THE BIRTH OF A WEBSITE

PRESENTED BY SIR DAVID ATTENBOROUGH

1. DESIGN / UX

- Wireframing
- "Click" prototyping
- Visual design
- 9,000,000 rounds of amendments

2. COPYWRITING

- Site content gathering/writing
- Set tone of voice
- Audit existing copy

3. FRONT-END DEVELOPMENT

- Translate design into build
- Device testing
- Program interactions

4. BACK-END DEVELOPMENT

- Construct database
- Create CMS
- Program website functionality

5. TESTING

- Bug fixing
- User acceptance testing
- Quality assurance

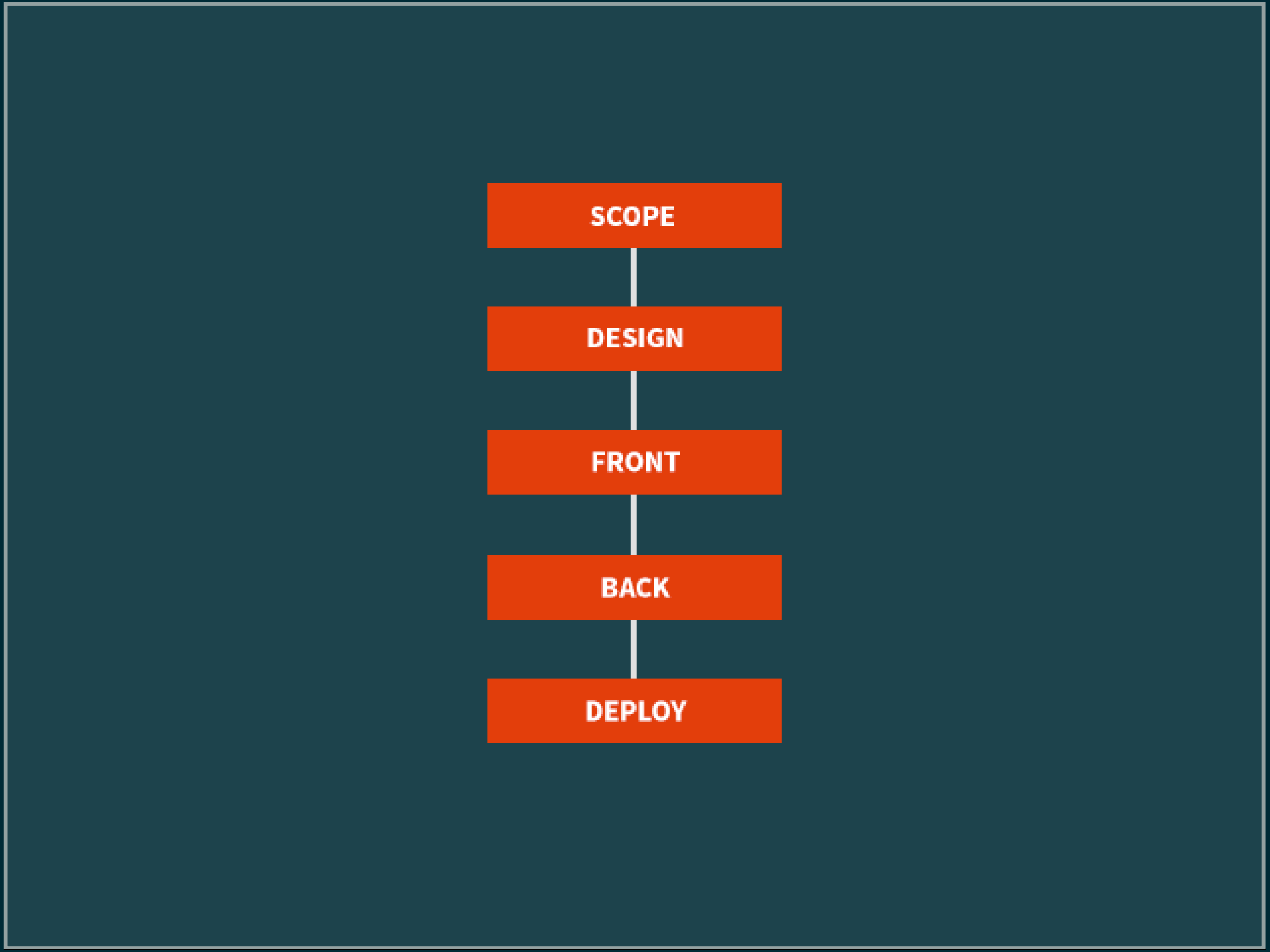
6. DEPLOYMENT

- Infrastructure setup
- Domain / DNS setup
- Code deployment

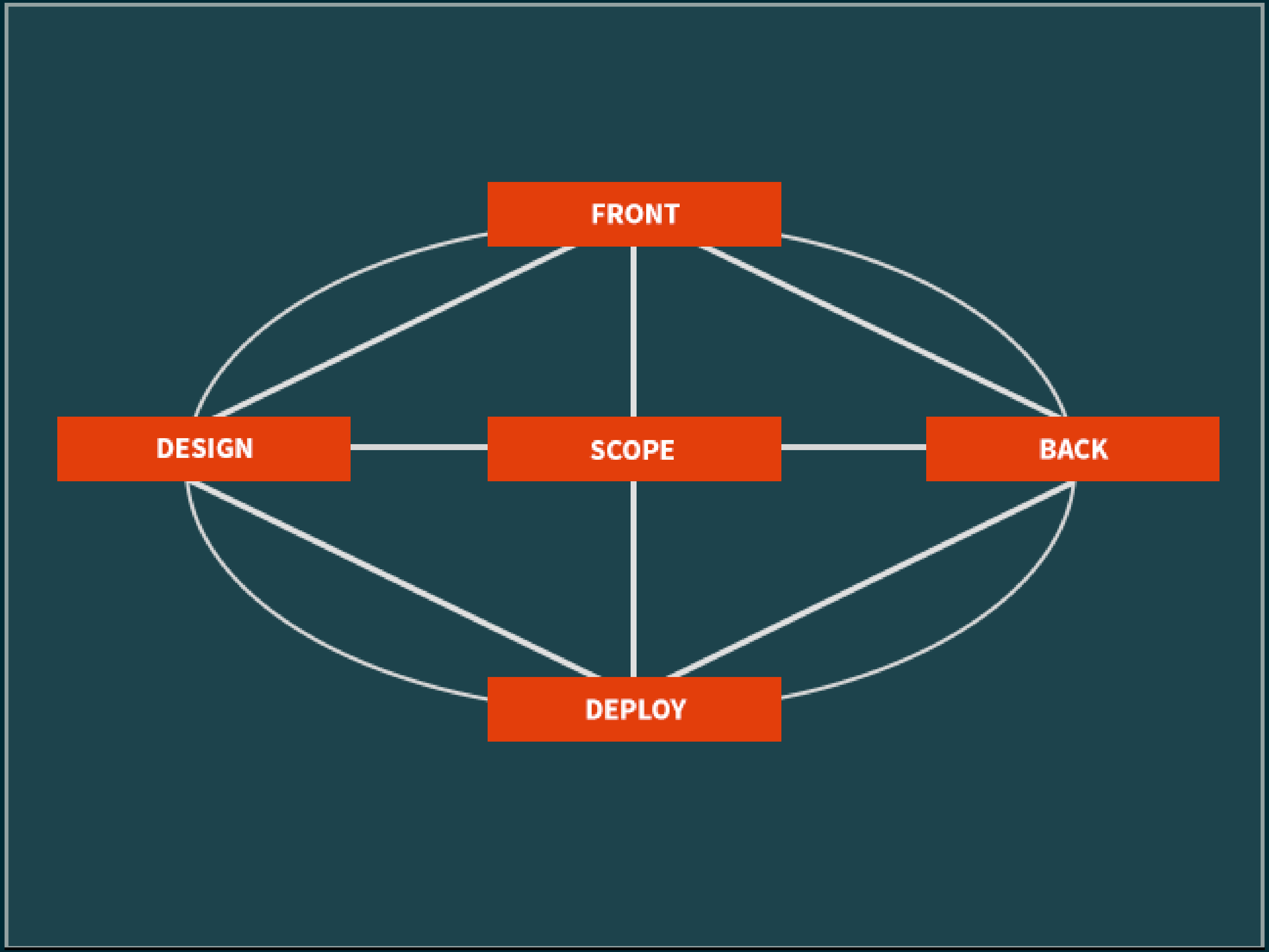
7. OPTIMISATION

- Asset minification
- Image compression
- SEO
- Refactoring algorithms

WATERFALL



AGILE



KEY TERMS WE WILL USE

LET'S HAVE A QUICK LOOK AT

- HTML
- CSS
- Javascript
- Ruby
- Version control (Git)
- API (REST, JSON, WTF?)

HTML

WHAT IS IT?

- Hypertext Markup Language
- Defines Webpage structure
- Parsed by the Web Browser and turned into visual elements
- Not a "programming" language:
  - No control flow
  - No internal state
  - No arithmatic or logic

WHAT DOES IT LOOK LIKE?

THIS IS A HEADING

Alternate text

- List item goes here
- Another list item

CSS

WHAT IS IT?

- Cascading Style Sheets
- Style / enrich HTML documents
- Provides a simple syntax for applying rules to sets of Webpage elements
- Not a programming language, either!

WHAT DOES IT LOOK LIKE?

```
div {
  background-color: #ab34ed;
  font-family: Sans-Serif;
}

ul li.special {
  color: green;
}
```

JAVASCRIPT

WHAT IS IT?

- Provides Webpage behaviour
- Interacts with the user
- Manipulates Webpage elements
- It is a programming language
    - To be exact: A multi-paradigm, dynamic, mildly-strong/duck typed scripting language supporting prototype-based inheritance and first-class functions
- Executed on the users machine

WHAT DOES IT LOOK LIKE?

```
var name = "Andrew",
    age = 29;

if (age >= 21) {
  console.log("You are an adult " + name + ", behave like one!");
} else {
  console.log("Carry on...");
}
```

RUBY

WHAT IS IT?

- A general-purpose programming language
- Created by Yukihiro Matsumoto (Matz) in Japan in 1994
- One of many popular options for server-side programming
- Allows for "dynamic" webpages that may be generated from external sources (databases, etc)
- Executed on the web server

WHAT DOES IT LOOK LIKE?

```
cities = ["Melbourne", "Sydney",
          "Los Angeles", "Ulan Bator"]

cities.each do |c|
  puts "Let's go to #{c}"
end
```

VERSION CONTROL

(GIT)

WHAT IS IT?

- A system for managing changes in files over time
- Provides a log of all changes, allowing us to rollback and forward safely
- We can see who did what, when, where and why
- Very important for collaborating effectively with others
- Git is a popular option, but there are many
- Github.com allows you to publish and share git repositories. A social network centered around coding
    - Check me out: github.com/buntine

API

WHAT IS IT?

- Application Programming Interface
- A Web API is a way of communicating to other systems over the Internet
- An API provides a set of operations that we can call upon
- Many APIs use JSON, a lightweight data format, for accepting instructions and returning results
- This is how we "integrate" with Facebook, Twitter, LinkedIn, etc, etc, etc

REVIEW

QUESTIONS?

WHAT'S NEXT?

FRONT-END WEB DEVELOPMENT

THANK YOU!