Lesson 1
Dan, Madhu, Su Min

The Accelerator

- The goal: to take students from knowing nothing about app development to a level where they can build and ship their own apps.
- 3 Tracks: iOS, Android, and Backend development
- Receive close mentorship from experienced mentors
- Receive career and academic guidance to help ensure success at Rutgers and beyond
- 1st semester: learn the fundamentals of your track
- 2nd semester: work with 2 mentees (one from iOS and Android) and a mentor to build an app of your choosing

How it's Built

We currently tentatively have 4 talks scheduled for this semester:

AppNexus — October 7th

facebook — Early November

mongoDB — Late October

TWO SIGMA — Late November

11

Semester Goals

- Fundamental understanding of client-server model
- API Structure
- Use Flask, MongoDB to build a simple backend project
- Authentication (Google OAuth, Flask middleware)
- Hosting (Heroku, AWS)
- Using 3rd Party APIs/SDKs
- Create a simple UI for the frontend

Today's Agenda

- How the internet works
- Servers and Clients
- Overview of the tools we will be using
- APIs
- Databases
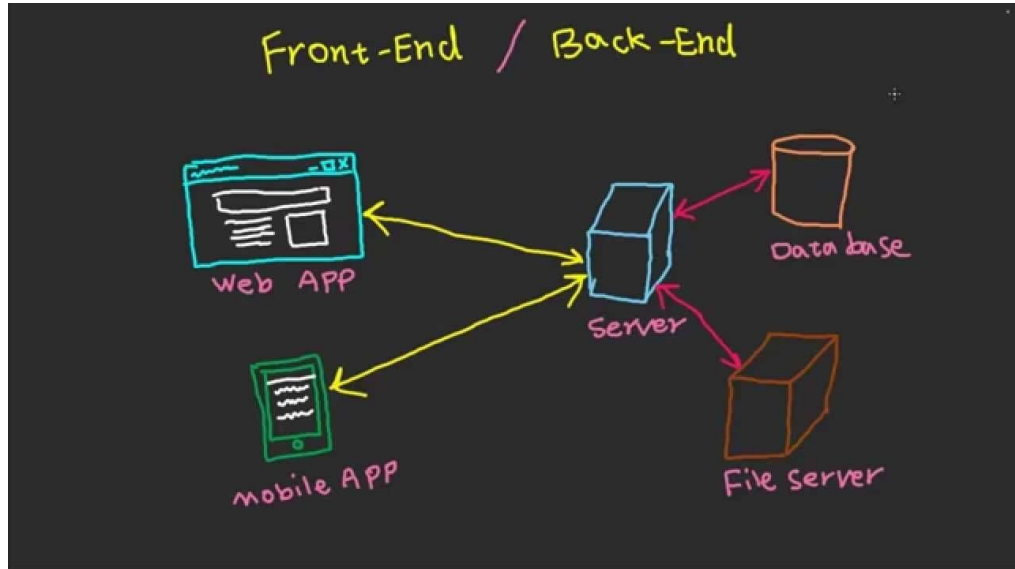- Frontend vs Backend

How does it work?

Client

Server

Your browser connects to a server and requests a page.

The server sends back the requested page.

©2003 HowStuffWorks

Your machine running a Web browser

Server machine running a Web server

Web applications

- Client-server program where the client runs in a web browser
  - Facebook, Twitter, Gmail, Google
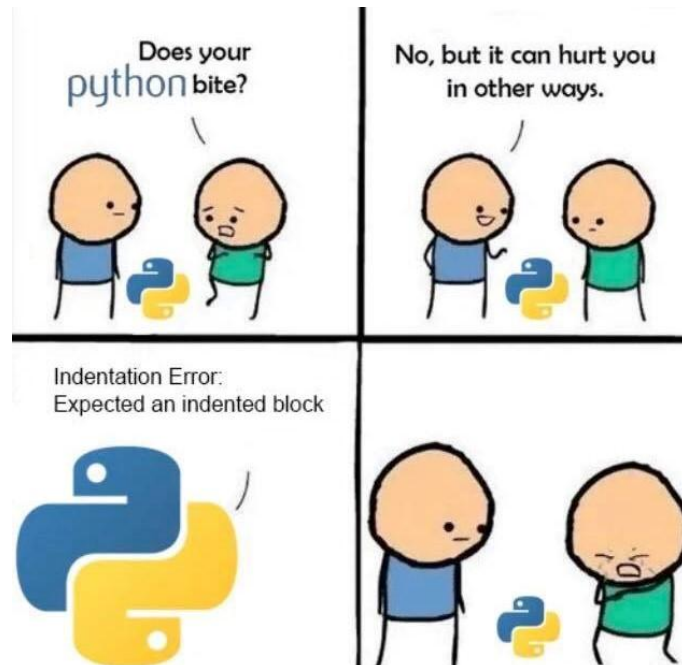- Flask
  - Web app framework
  - Run server

Backend

- Server,
- application,
- database

Python

- High-level language
- Excels in
  - String manipulation
  - File Manipulation
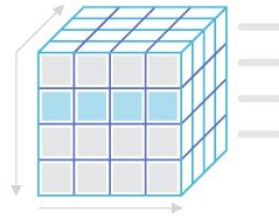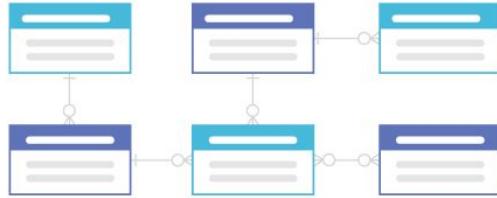 - Machine Learning
  - Data Science

Flask

- Microframework for python
- Easy to learn
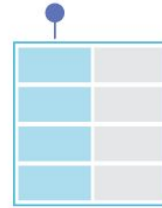- stages, hosts and handles your complete web application

Databases

SQL

NoSQL

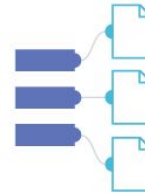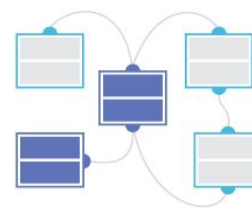Relational Database Management Systems (RDBMS)

Key-Value

Graph
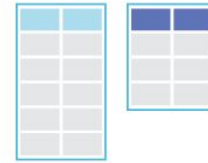
Online Analytical Processing (OLAP) Cube
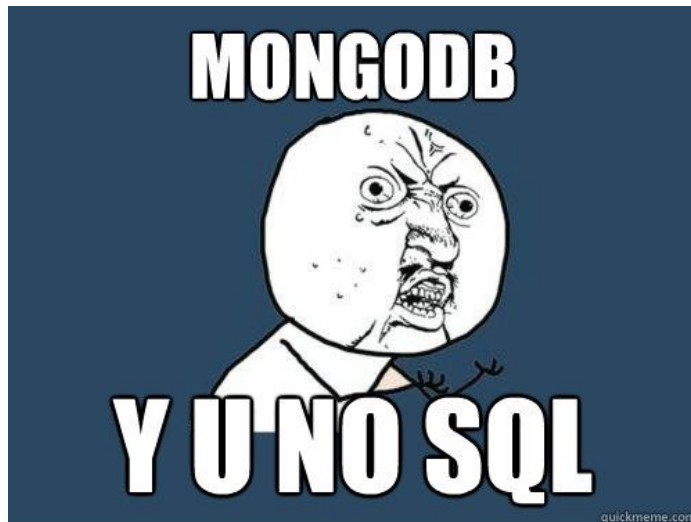
Document

Column store

MongoDB

- NoSQL
  - Json format
  - No tables
- Flexible
- Lower latency

Response/Request
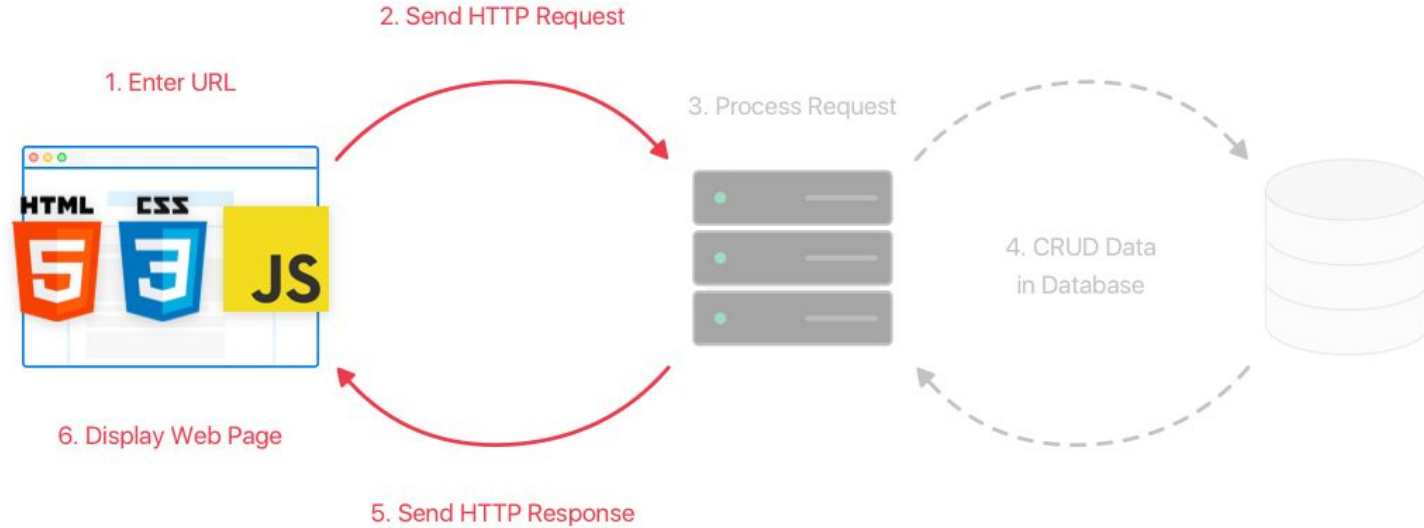

Browser   Server   Database

2. Send HTTP Request

1. Enter URL

3. Process Request

4. CRUD Data in Database

6. Display Web Page

5. Send HTTP Response

CREATE READ UPDATE DELETE

CRUD

request

header
- request line
  - http method
  - request-uri
- host
- name-value pairs

body

client

server

response

header
- status line
- name-value pairs

body
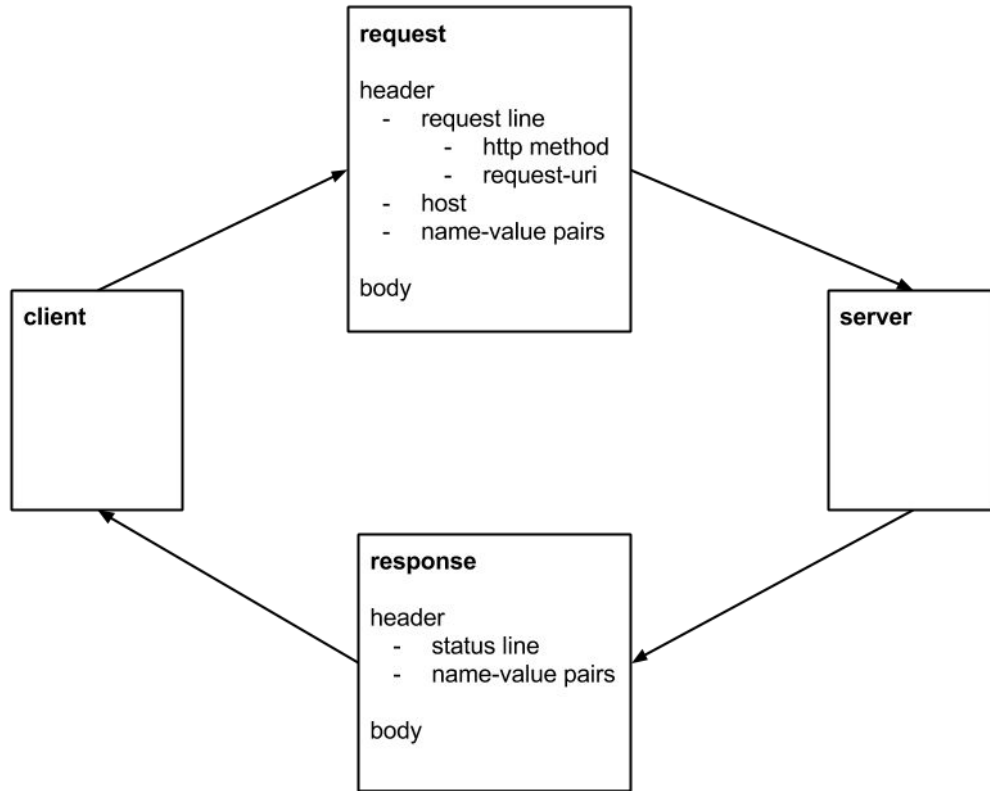
T O D O

- Introductions And Stuff
- Overview of our goals
    - Building a skill set etc..
    - Ask what they want to achieve
- Go over what skills they already have
- Technical portion
    - How the internet works
        - Data is routed to servers
        - HTTP is the base of the language that the computer talks in with the server
        - Server vs Client
    - Architecture overview
    - Overview of the tools we will be using
        - Python
        - Flask
        - Linux
        - Mongodb
    - REST APIs