INTRODUCTION TO WEB APPLICATION DEVELOPMENT

KARIM ALIBHAI

TODO :

- Get to know each other
- A welcome to the course
- Your cumulative projects
- **Lesson 1:** What on Earth is a web app?
- **Lesson 2:** How Web Servers Chat
- **Lesson 3:** The Black Screen with the Green Text

INTRODUCTIONS

- Your name
- Your current/future career path
- Something you are passionate about
- An interesting fact about your passion that the average person wouldn't know

THE CATCH

- Everyone will be assigned a random partner.
- You must introduce your partner, not yourself.

BEGIN: 00:10:00.

TIME'S UP!

TEACHING ASSISTANTS

- Diana Dai
- Farhan Kanjiyani
- Jeff Peng

DIANA DAI

- Studying Computer Science.
- I <3 _.
- `TODO:` interesting fact.

FARHAN KANJIYANI (@FARHAN5498)

- Studying Business & Computer Science.
- I <3 Video Games & Technology.
- `TODO:` interesting fact.

JEFF PENG (@JEFFTHEHAUTBOIS)

- Studying Electrical Engineering.
- I <3 _.
- `TODO:` interesting fact.

TEACHER

KARIM ALIBHAI (@KARIMSA)

WHO AM I?

- My name is Karim. (?!)
- I'm a Developer.
- I <3 JavaScript, W3, & teaching.
- Although JavaScript is a web language, it can be used to create:
    - robots
    - complex AIs
    - mobile & desktop apps

WELCOME!

PACE

- In-class teachings will keep up with class average.
- TAs are present for the purpose of helping students who would like extra help.
- TAs as well as the teacher will be available via email outside of class time to answer questions.

ASSIGNMENTS

- There will be regular assignments to help you learn to properly apply the lessons.
- All assignments are mandatory.
- There are supplemental exercises for people that would like extra practice.
- There are advanced exercises for people that would like to challenge themselves.

READINGS & QUIZZES

- There will be regular readings **required** to keep up with the course.
- There will be **no quizzes**.
- Whenever possible, we will live code projects together.
  - Every person will be responsible for creating a separate part of the project.
  - Due to this, *if you do not keep up with readings, your **entire** group will suffer.*

CUMULATIVE PROJECT

- The best practice for a working environment is a working environment.
- Think of a family member or friend that would benefit from a web application.
- Either contact this person and ask if you can create it for them,
- or create it just for fun.

REQUIREMENTS & TIME SPENT

- You may work in pairs of two or alone.
- There will be time after some classes to work on your projects.
- You must also work on your projects at home.
- Try to incorporate as many of the lessons you learn as possible.

THE GRAND PRIZE

- The top 3 projects will receive a *secret grand prize.*

LESSON 1: WHAT ON EARTH IS A WEB APP?

THE LAYERS OF A COMPUTER

- Basic Input and Output System (BIOS) is responsible for communication with all hardware.
- Operating System (OS) is the software responsible for acting as a host.
  - It runs user-level applications.
  - Provides software known as `drivers` for application to communicate with hardware.
  - Controls the remaining layers.

THE FILESYSTEM LAYER

- Filesystem is responsible for long-term data storage and retreival.
- The hardware behind the filesystem is the hard drive(s).
- There are two means of data organization:
  - A packet of data is referred to as a `file`.
  - A group of these packets is known as a `directory` (often referred to as a folder).
  - A file *cannot* be within another file, only within a directory.
  - A directory *can* be within another directory.

NOTES ON THE FILESYSTEM

- The location of a file or directory is referred to as the `path`.
- Within a path, names of files and directories are separated using a `path separator`.
  - On Unix-based systems, the path separator is `/`. (i.e. `/path/to/my/file.txt`)
 - On Windows, the path separator is `\`. (i.e. `\path\to\my\file.txt`)

QUESTION

- Given the principles of the filesystem, how would you tell what names belong to files and which ones are directories?

- For instance: `/a/b/c/d` (name the type for each of the letters).

THE FILESYSTEM TREE

- The top-most directory in any OS is referred to as the `root` directory.
- The name of the directory is not `root`, that is simply what it is referred to as.
- The name of the root directory is the same as the path separator.
- When a file or directory is nested in another directory, we can say that it is the child of the other directory.
- Therefore, the other directory is the parent of this file or directory.

THE FILESYSTEM TREE
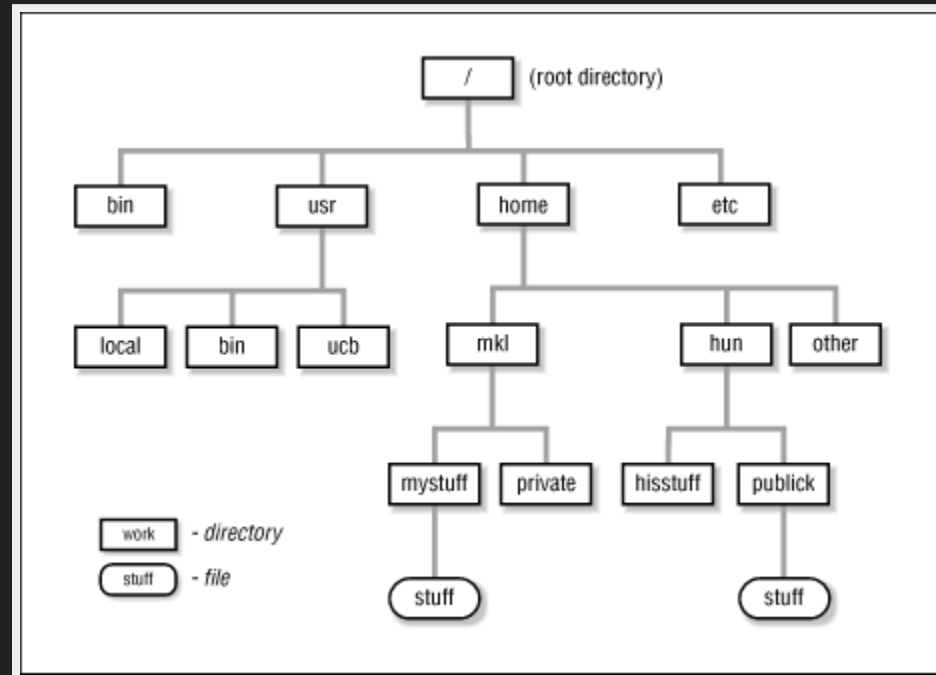
THE NETWORKING LAYER

- Responsible for inter-device communication.
- Underlying *internal* hardware is known as the Network Interface Card (NIC).
- Networking involves much *external* hardware as well.

INTERNET PROTOCOL ADDRESS

- An IP address is an address of a device on a network.
- The format of an IP address is predefined.
- IPv4 addresses (most common) format: four groups of decimal digits (8 bit groups).
  - For example: 192.168.1.101
- IPv6 address format: eight groups of four hexadecimal digits (16 bit groups).
  - For example: 2001:0db8:85a3:0000:0000:8a2e:0370:7334

THE ROUTER

- Creates a local network for inter-device communication for all connected devices.
- Manages communication between local network and the internet.
- For devices, communication on the local network and devices on the internet is almost identical.
- Assigns a local IP address to all devices, including itself.
- Routers have a local IP **and** a public IP.

# THE LOCAL AREA NETWORK (LAN + WLAN)

Wireless Home Network Diagram
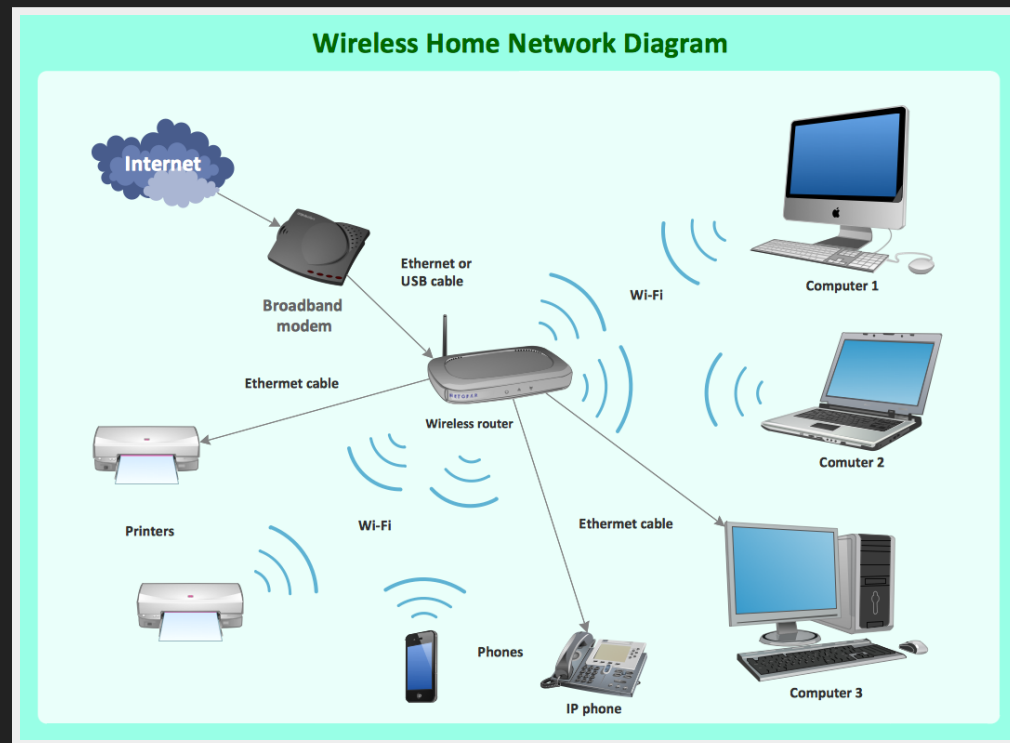
PORTS

- There are 65536 (or 2^16) `ports` on any given device.
- Ports are where all data between devices are sent from and to.
- Ports are specified by integers. (i.e. the first port is 1, the last is 65536)
- Ports < 1024 can only be accessed by applications with administrative access.
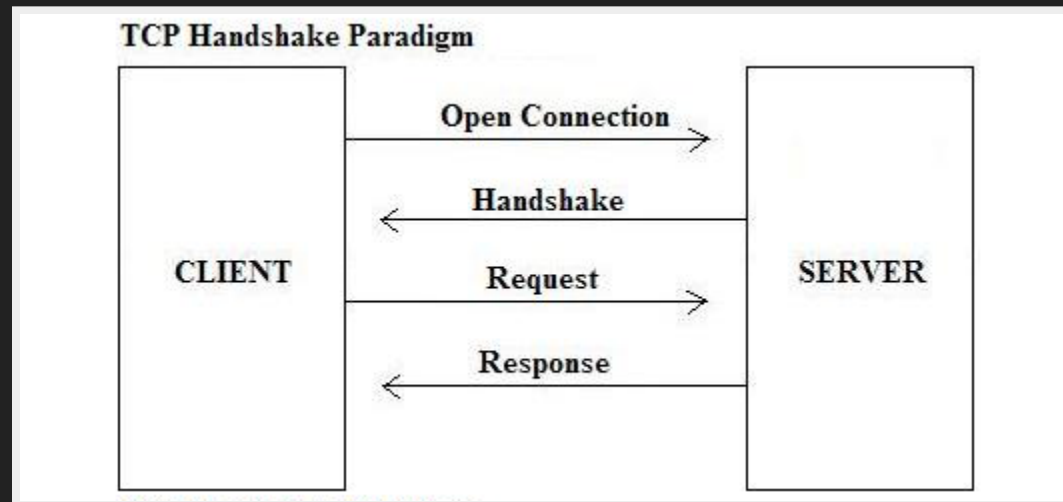- There are two main protocols which can be used to network: TCP & UDP.

USER DATAGRAM PROTOCOL (UDP)

- Known as a `best-effort` protocol.

UDP Request / Response Paradigm

CLIENT → Request → SERVER

SERVER → Response → CLIENT
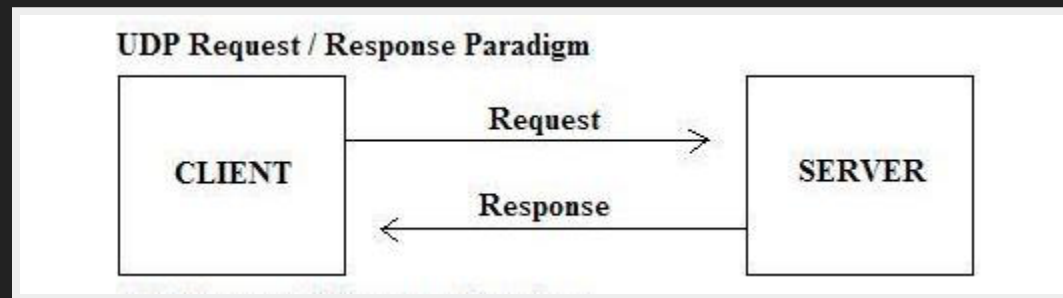
TRANSMISSION CONTROL PROTOCOL (TCP)

- Requires an on-going connection between hosts.
- Each host must continuously acknowledge received and sent data.
- This keeps both hosts up-to-date on each other's presence.

TCP Handshake Paradigm

CLIENT — Open Connection → SERVER
CLIENT ← Handshake — SERVER
CLIENT — Request → SERVER
CLIENT ← Response — SERVER

NOTES ON NETWORKING

- For a device to receive data over a network, it must have an application that binds onto a port.
- For a device to send data over a network, it needs the IP address and port of the receiver.

PUBLIC ROUTING

- When communicating with devices over the internet, you specify the external device's router's public IP.
- The router then forwards (or *routes*) the connection through to the device on the LAN.
- But what about when IP addresses change? And aren't IPs too complicated for users to specify?

DOMAIN NAME SYSTEM (DNS)

- DNS is responsible for resolving domain names to IP addresses.
- Domain names are alpha numeric names that can be bought from Domain Registrars.
- There are only a set number of top-level domains. (i.e. `com`, `ca`, `org`, `co`)
- Your domain name must have a top-level domain attached to it.
- Your domain may have as many sub-domains as you would like (i.e. `images`.google.com, `docs`.google.com)
- Your sub-domains can have sub-domains. (i.e. `scontent`.xx.fbcdn.com)

LESSON 2: HOW WEB SERVERS CHAT

LESSON 3: THE BLACK SCREEN WITH THE GREEN TEXT