



# Lesson 1:

# Front-end web development

# INTRODUCTION

- Traditionally, when web developers started building websites in the 90s, they would wear all the hats.
  - they would set up the server,
  - write all the server-side code that would eventually output HTML and
  - finally sprinkle on some CSS to make things look pretty.

# INTRODUCTION

- Due to the advancement of modern web technologies and computer hardware, we're able to create more sophisticated experiences than ever.
- Because of this complexity, the role of a web developer has naturally evolved and split into separate disciplines.
- Specifically, a line was drawn between the back-end and front-end.
  - Back-end being server-side programming and front-end being client-side.

# INTRODUCTION

- A front-end dev't deals with technologies used to design and develop the user interface of the web application and its interaction with the users.
- It is the practice of producing HTML, CSS and JavaScript for a website or Web Application so that a user can see and interact with them directly.
- Also known as “client-side development”

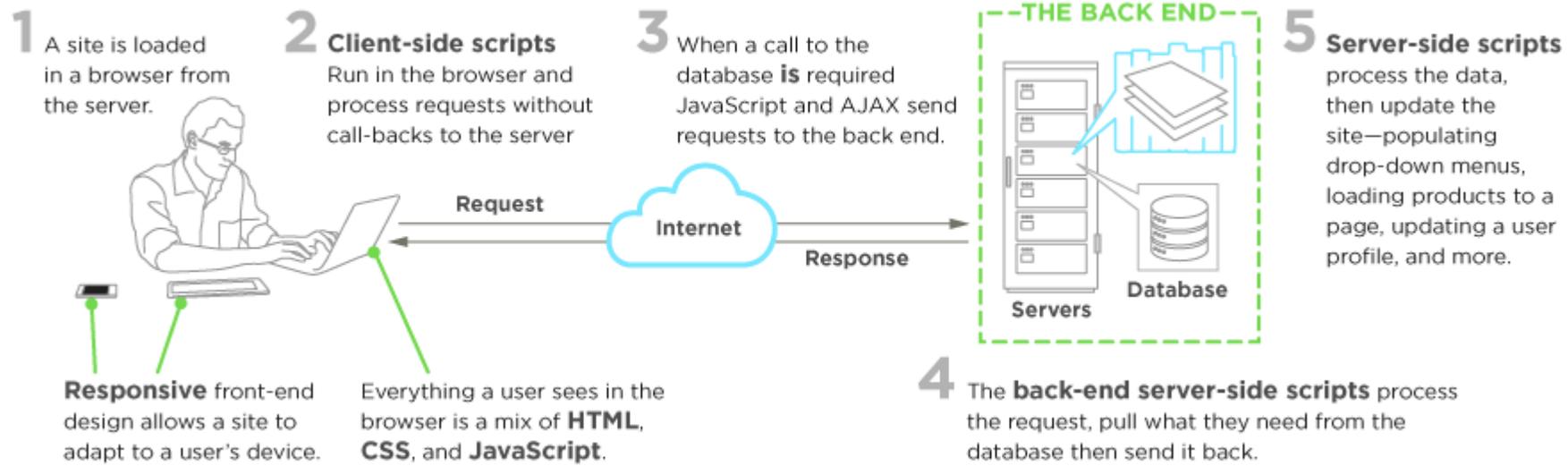
# INTRODUCTION

- Objective of designing a site is to ensure that when the users open up the site they see the information in a format that is easy to read and relevant.
- This is further complicated by the fact that users now use a large variety of devices with varying screen sizes and resolutions
  - thus forcing the designer to take into consideration these aspects when designing the site.

# INTRODUCTION

- They need to ensure that their site comes up correctly in different browsers (cross-browser), different operating systems (cross-platform) and different devices (cross-device)

# INTRODUCTION



## Front end web development

# FRONT-END LANGUAGES

- **HTML — Hypertext Markup Language**
  - Website pages and documents are written in HTML, which consists of **base content** combined with inter-linear tags that provide **semantic information** about the content they enclose.
- **CSS — Cascading Style Sheet**
  - Provides a set of detailed instructions to the browser (or a printer) about how the content of an HTML document should be **displayed**. It includes details like font declarations, sizing, color, on-page placements, and layering precedence.

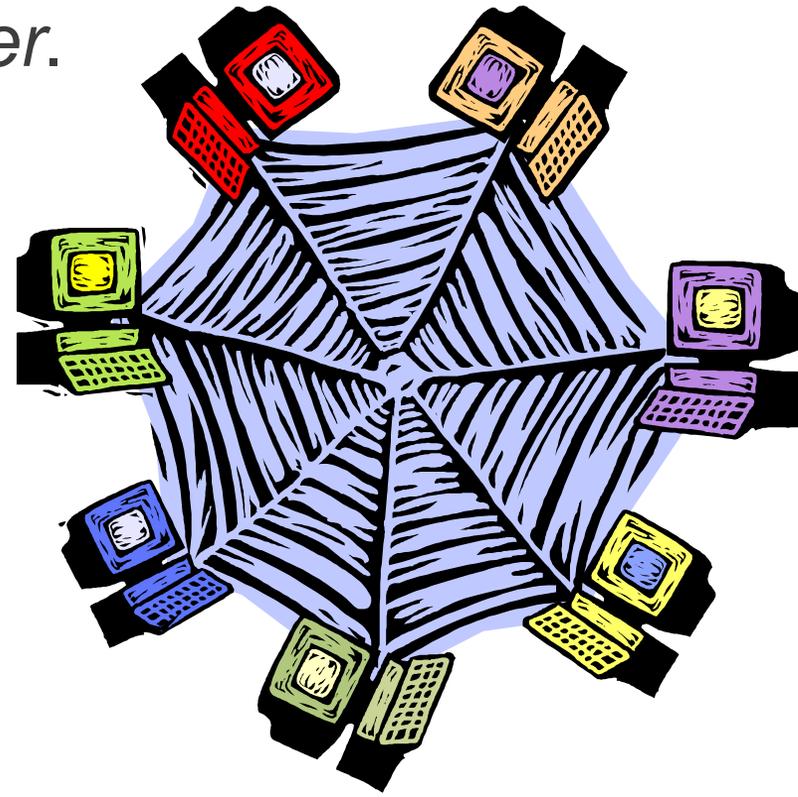
# FRONT-END LANGUAGES

- **JS — JavaScript**

- A scripting language that can be used to provide rich web interaction inside a browser. It is event-driven, **responding to actions taken by the user** (such as clicking, mousing, hovering, and typing).

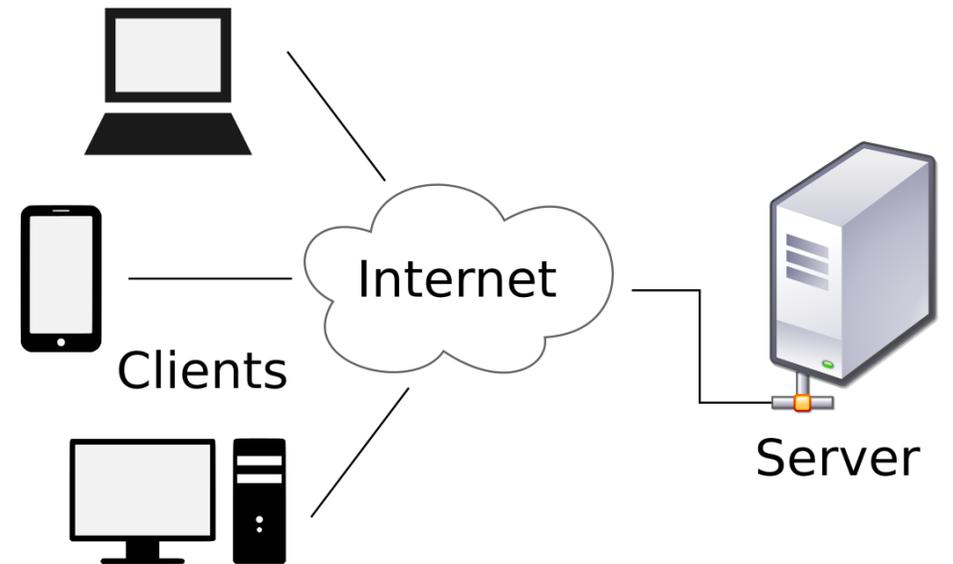
# CLIENT-SERVER ARCHITECTURE

- A network architecture in which each computer or process on the network is either a *client* or a *server*.



# CLIENT-SERVER ARCHITECTURE

- Components
  - Clients
  - Servers
  - Communication networks and protocols



# CLIENT-SERVER ARCHITECTURE

## Clients

- Applications that run on user's computer
- Rely on servers for
  - Files, devices and processing power
- Example: E-mail client, Web client (browser)
  - An application that enables you to send and receive e-mail

**Clients are Applications**

# CLIENT-SERVER ARCHITECTURE

## Servers

- Computers or processes that manage network resources
  - Disk drives (file servers), Printers (print servers), Network traffic (network servers), Web Servers
- Example: Database Server
  - A computer system that processes database queries

**Servers Manage  
Resources**

# CLIENT-SERVER ARCHITECTURE

## Communication Networks

- Connect clients and servers

# CLIENT-SERVER COMPUTING

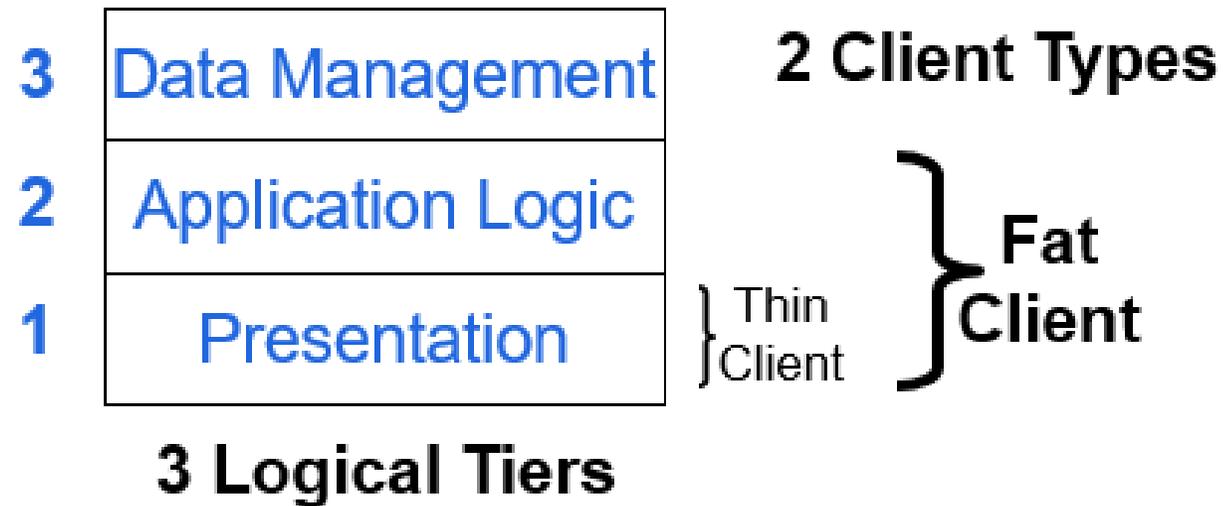
- Process takes place on the server and on the client
- Servers
  - Store and protect data
  - Process requests from clients
- Clients
  - Make requests
  - Format data on the desktop

# CLIENT-SERVER COMPUTING

## Application Components

- Software application functions are separated into three distinct parts
  - Presentation, application logic and data management

# CLIENT-SERVER COMPUTING



Database Applications:

Most common use of client-server architectures

# REVISION OF CLIENT SIDE SCRIPTS

- HTML
  - Structure, elements and forms
- CSS
  - How to, using CSS as internal, external and inline
- JavaScript
  - Structure, using js with HTML (internal and external js)

# REVISION OF CLIENT SIDE SCRIPTS

# Exercises

# FRONT-END WEB FRAMEWORKS

- What is a web framework and why we use it?
- Examples of front-end frameworks
  - CSS?

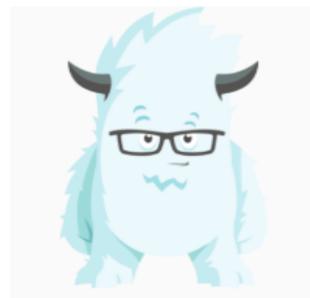
Bootstrap



Semantic-  
UI



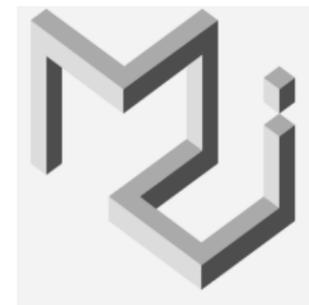
Foundatio  
n



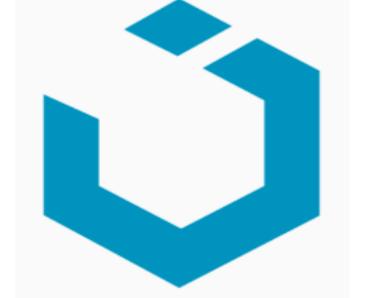
Materializ  
e



Material  
UI

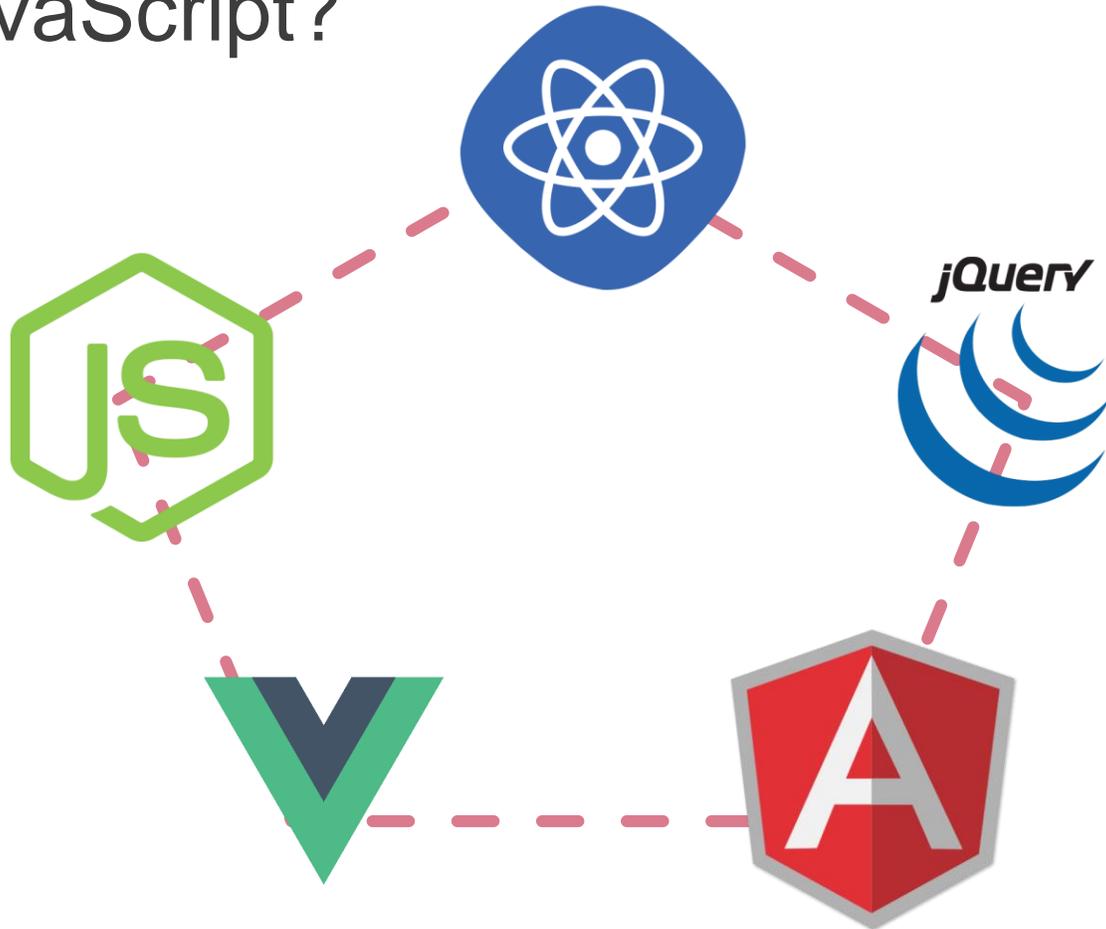


UIKit



# FRONT-END WEB FRAMEWORKS

## ■ JavaScript?



- React JS
- Vue.js
- Angular.js
- Node.js
- jQuery



**Thank You!**