SE 3003– Web Engineering

Laiba Imran

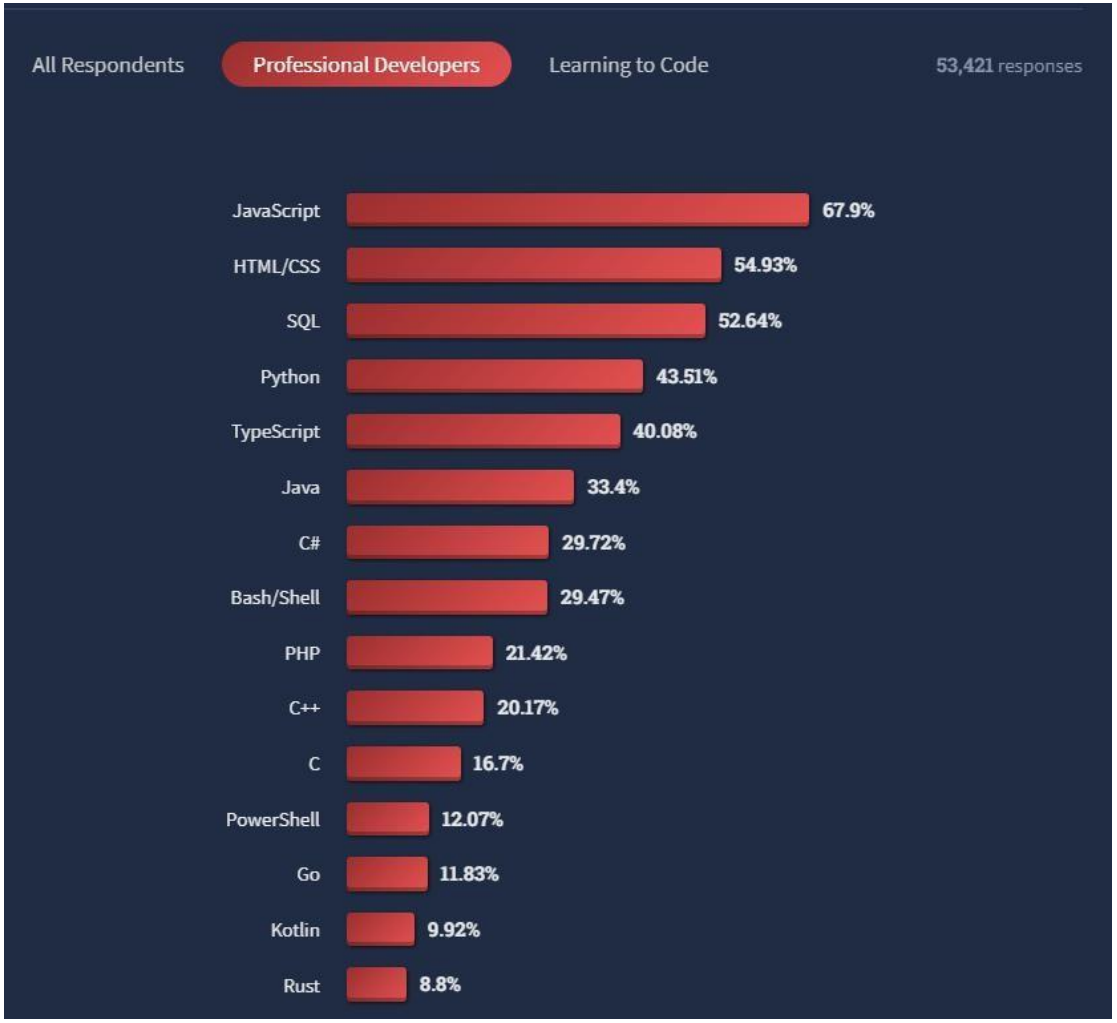
About Me

- Lecturer at NUCES-FAST (Aug 2024 -)
- Instructor at NUCES-FAST (Jan 2023 -July 2024)
- MS Software Engineering – NUCES-FAST (2022-2024)
- BS Computer Science - NUCES-FAST (2018-2022)
- Email: laiba.imran@nu.edu.pk
- Office: A-304D
- Contact Hours:

Course Objectives

- Understanding of modern web application development technologies
- Develop and design Web interfaces using latest UI frameworks
- Understand the best web development practices being followed in the industry
- Training on industry-oriented web frameworks
- Utilize the incredible power of web technologies
  - Develop web based of MERN, and Serverless Stack

Most Popular Technologies *(Stackoverflow Survey)*



https://survey.stackoverflow.co/2022/#technology-most-popular-technologies

Course is Different (mini-courses)

Frontend Technologies

HTML5, CSS3, JavaScript, TypeScript, ReactJS

Backend Technologies

NodeJS, ExpressJS, MongoDB

Web Service Technologies

Cloud, Serverless, IaaS, PaaS, Database Services, Storage Services

Target Students

- Want to pursue good career as a **Full Stack Web Developer/Web Designer**
- Want to learn cutting-edge Web development technologies
- Want to get hands-on practice on industry-oriented web frameworks
- Feel Comfortable in coding (e.g., C++, Java, Python...)
  - Has built, or could build, a single-user application
- Ready to take coding challenges

Contents and Organization

Class Policies and Guidelines

Attendance policy:

- **Will be taken at the start of the class**. Students ***appearing late in the class after the attendance*** will be marked **"Absent"**
- **80%** attendance is compulsory

Plagiarism policy:

- Plagiarism in **midterm/final** exam may result in **F grade** in the course.
- Plagiarism in an assignment items (**assignments, quizzes & project**) will result in zero marks in the whole assignments items category. If fore mentioned act is repeated more than once the instructor can refer a case to the Department Disciplinary Committee (the maximum punishment can be award of **'F' grade** in that course.)

Class Policies and Guidelines

Course retake policy:

- Midterm/final exam retake
  - The examination assessment and retake committee decide the exam retake/pretake cases.
- Assignments/quizzes retake
  - There will be **no retake** of any assignment or quiz.

Assignments and Projects

- Where **~80%** of your learning will take place
- For learning, not evaluation -> low marks (~25%)
- Posted to **Google Classroom** and **Slate**
- All **assignments** will be individual and **project** will be in group (max. 2-3 students)
- Program must work, compile errors / runtime errors lose all correctness points
- Copying solution code or giving code to someone else is **CHEATING -> F** in the course

Class Policies and Guidelines

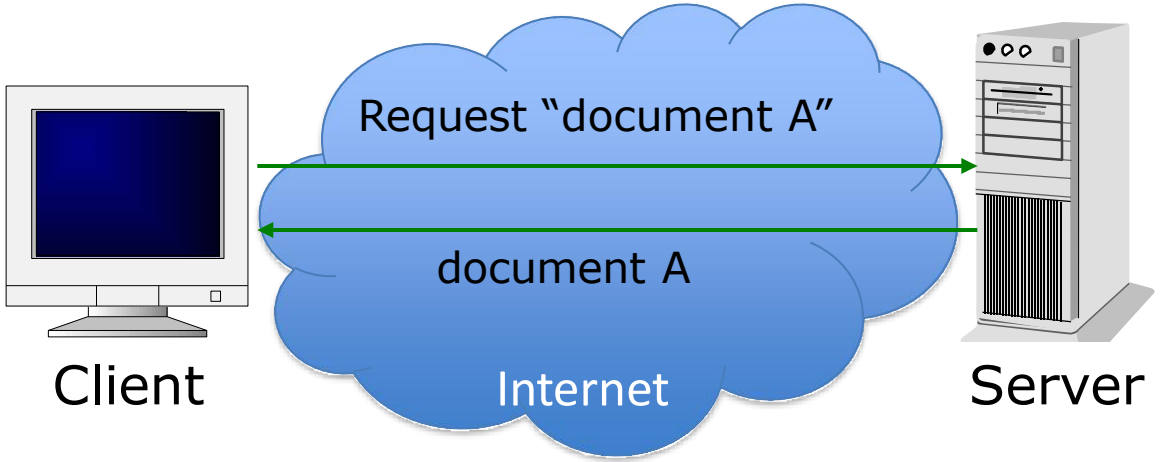
- **Don’ts**
  - Use of cell phones
  - Discussion with fellows during class
  - Early leave
  - Frequent movements In-out of class

- **Do’s**
  - Be interactive, ask questions
  - Participate in the lecture especially during hands-on practice
  - Be prepared for practice sessions

Intíoduction to Web

Web Essentials

- **Client**: web browsers, used to surf the Web
- **Server systems**: used to supply information to these browsers
- **Computer networks**: used to support the browser-server communication



Internet v.s. Web

The Internet: a inter-connected computer networks, linked by wires, cables, wireless connections, etc.

Web: a collection of interconnected documents and other resources.

The world wide web (**WWW**) is accessible via the Internet, as are many other services including email, file sharing, etc.

How does the Internet Work?

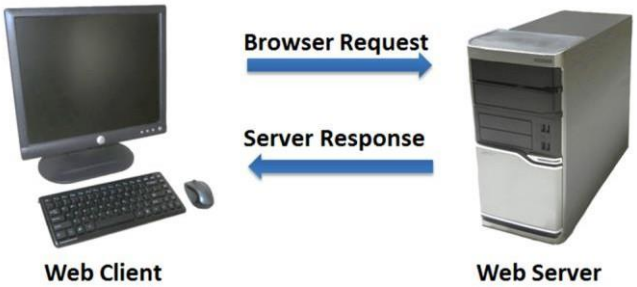
Through communication protocols

A **communication protocol** is a specification of how communication between two computers will be carried out

IP (Internet Protocol): defines the packets that carry blocks of data from one node to another	**TCP** (Transmission Control Protocol) and **UDP** (User Datagram Protocol): the protocols by which one host sends data to another.	Other application protocols: **DNS** (Domain Name Service), **SMTP** (Simple Mail Transfer Protocol), and **FTP** (File Transfer Protocol)

The World Wide Web (WWW)

- WWW is a system of interlinked, hypertext documents that runs over the Internet
- Two types of software:
  - **Client**: a system that wishes to access the information provided by servers must run client software (e.g., web browser)
  - **Server**: an internet-connected computer that wishes to provide information to others must run server software
  - Client and server applications communicate over the Internet by following a protocol built on top of TCP/IP – HyperText Transfer Protocol (HTTP)



Basics of the WWW

- **Hypertext**: a format of information which allows one to move from one part of a document to another or from one document to another through hyperlinks
- **Uniform Resource Locator (URL)**: unique identifiers used to locate a particular resource on the network
- **Markup language**: defines the structure and content of hypertext documents

Web Client: Browser

- Makes HTTP requests on behalf of the user
  - Reformat the URL entered as a valid HTTP request
  - Use DNS to convert server's host name to appropriate IP address
  - Establish a TCP connection using the IP address
  - Send HTTP request over the connection and wait for server's response
  - Display the document contained in the response
    - If the document is not a plain-text document but instead is written in HTML, this involves rendering the document (positioning text, graphics, creating table borders, using appropriate fonts, etc.)

Web Servers

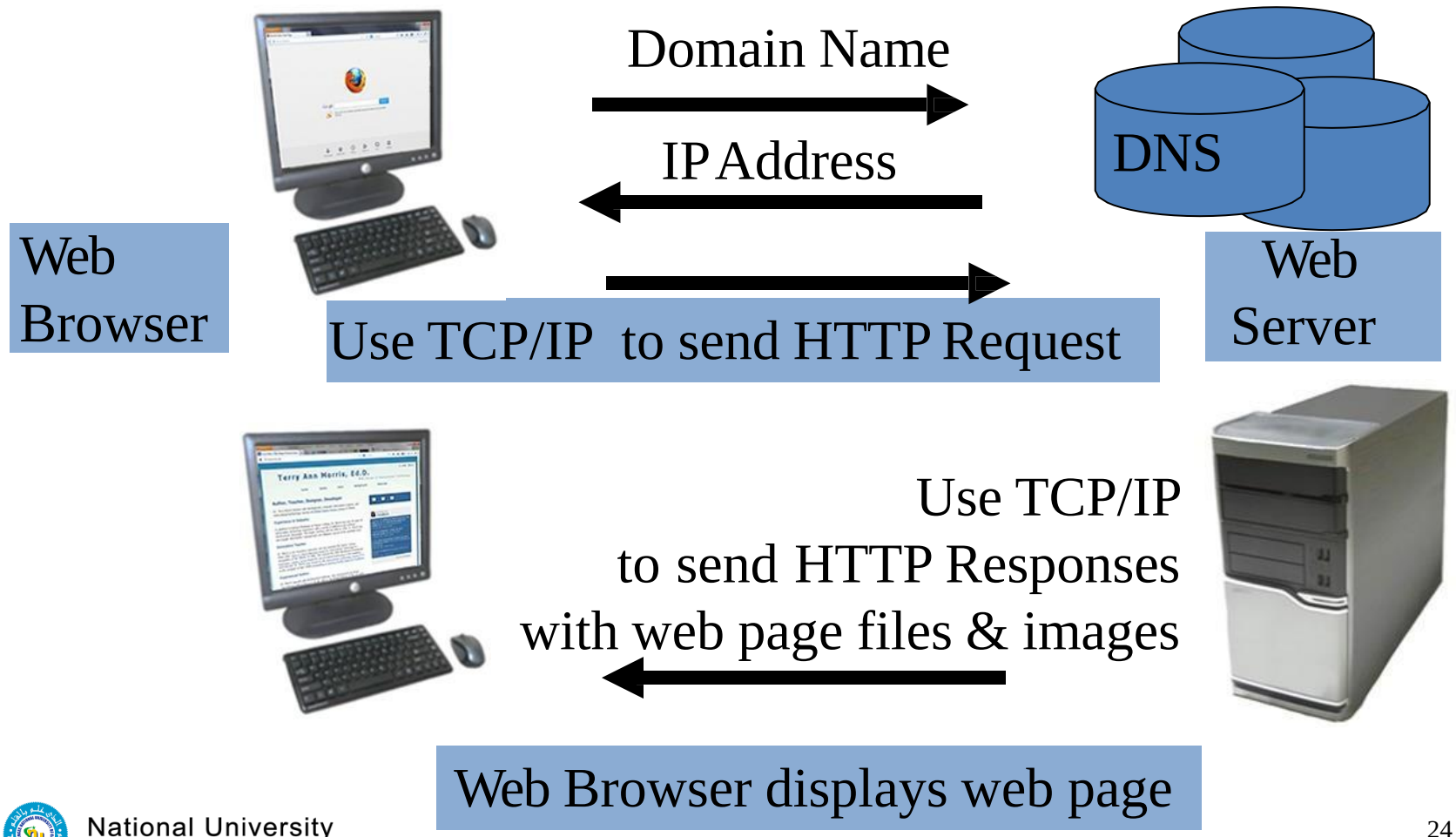
- Main functionalities:
  - Server waits for connect requests
  - When a connection request is received, the server creates a new process to handle this connection
  - The new process establishes the TCP connection and waits for HTTP requests
  - The new process invokes software that maps the requested URL to a resource on the server
  - If the resource is a file, creates an HTTP response that contains the file in the body of the response message
  - If the resource is a program, runs the program, and returns the output

Domain Name

- Locates an organization or other entity on the Internet
- Domain Name System (DNS)
  - Divides the Internet into logical groups and understandable names
  - Associates unique computer IP Addresses with the text-based domain names you type into a web browser
  - Browser: http://google.com
  - IP Address: 173.194.116.72

Domain Name System

- The Domain Name System (DNS) associates Domain Names with IP addresses.



Top-Level Domain (TLD) Name

- A top-level domain (TLD) identifies the right-most part of the domain name.

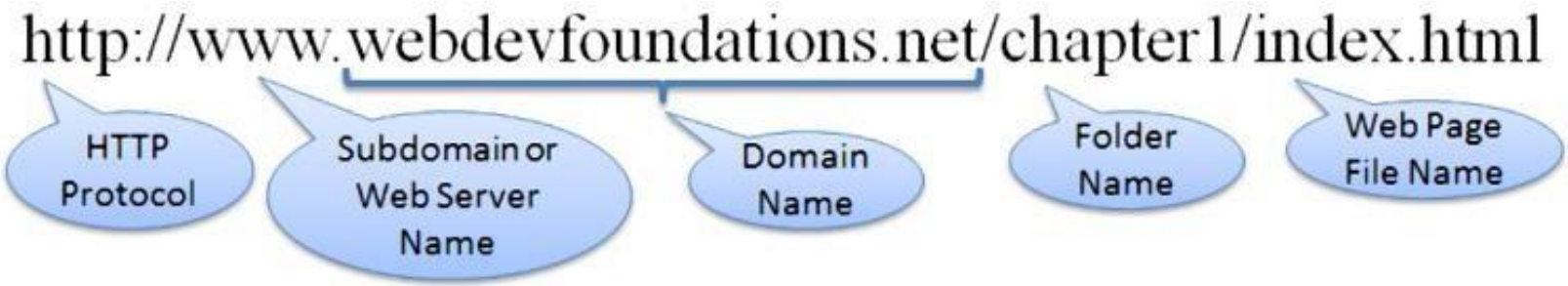
- Examples of generic TLDs:
  .com, .org, .net, .mil, .gov, .edu, .int, .aero, .asia, .cat, .jobs, .name, .biz, .mobi, .museum, .info, .coop, .post, .pro, .tel, .travel

Country Code TLDs

- Two character codes originally intended to indicate the geographical location (country) of the web site.
- In practice, it is fairly easy to obtain a domain name with a country code TLD that is not local to the registrant.
- Examples:
  - .tv, .ws, .au, .jp, .uk
  - See http://www.iana.org/cctld/cctld-whois.htm

Uniform Resource Identifier

- URI – Uniform Resource Identifier
  - identifies a resource on the Internet
- URL – Uniform Resource Locator
  - a type of URI which represents the network location of a resource such as a web page, a graphic file, or an MP3 file.



Static Web: HTML/XHTML, CSS

- **HTML** stands for **H**yper**T**ext **Ma**rkup **L**anguage
  - It is a text file containing small markup tags that tell the Web browser how to display the page

- **XHTML** stands for e**X**tensible **H**yperText **M**arkup **L**anguage
  - It is identical to HTML 4.01
  - It is a stricter and cleaner version of HTML
  - E.g., <!DOCTYPE>, <html>, <head>, and <body> are mandatory

- **CSS** stands for **C**ascading **S**tyle **S**heets
  - It defines how to display HTML elements

Static web limitations

- What is the drawback to simple document model?
  - Static
  - Assume that documents are created before they are requested

- What are examples of information that might be part of web documents that may not be known before they are requested?

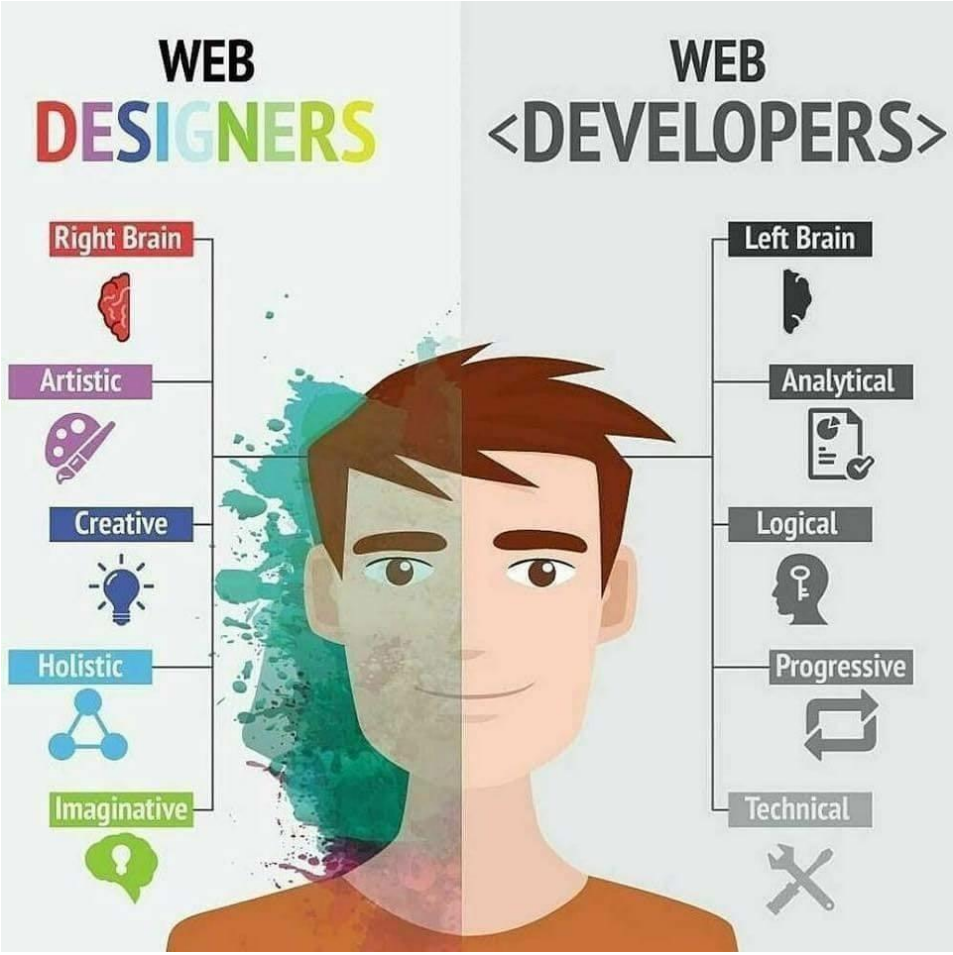
Client-Side Programming

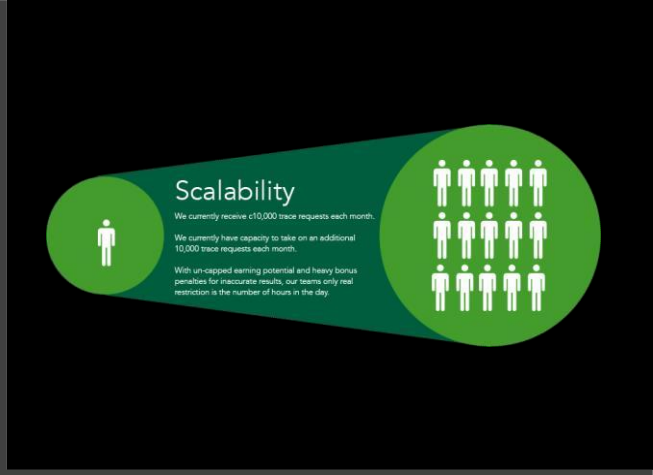
- Scripting language: a lightweight programming language
- Browser scripting: **JavaScript**
  - Designed to add interactivity to HTML pages
  - Usually embedded into HTML pages
  - What can a JavaScript Do?
    - Put dynamic text into an HTML page
    - React to events
    - Read and write HTML elements
    - Validate data before it is submitted to a server
    - Create cookies
    - …

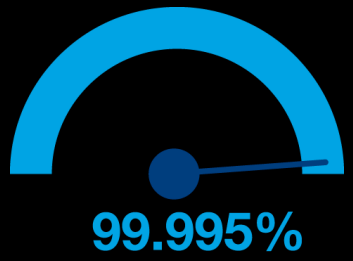
Server-Side Programming

- The requests cause the response to be generated
- Server scripting:
  - ASP.Net MVC: Microsoft product, uses .Net framework (*.asp)
  - CGI/Perl: Common Gate Way Interface (*.pl, *.cgi)
  - PHP: Open source, strong database support (*.php)
  - Java via JavaServer Pages (*.jsp)
  - …

Web Development vs Designing





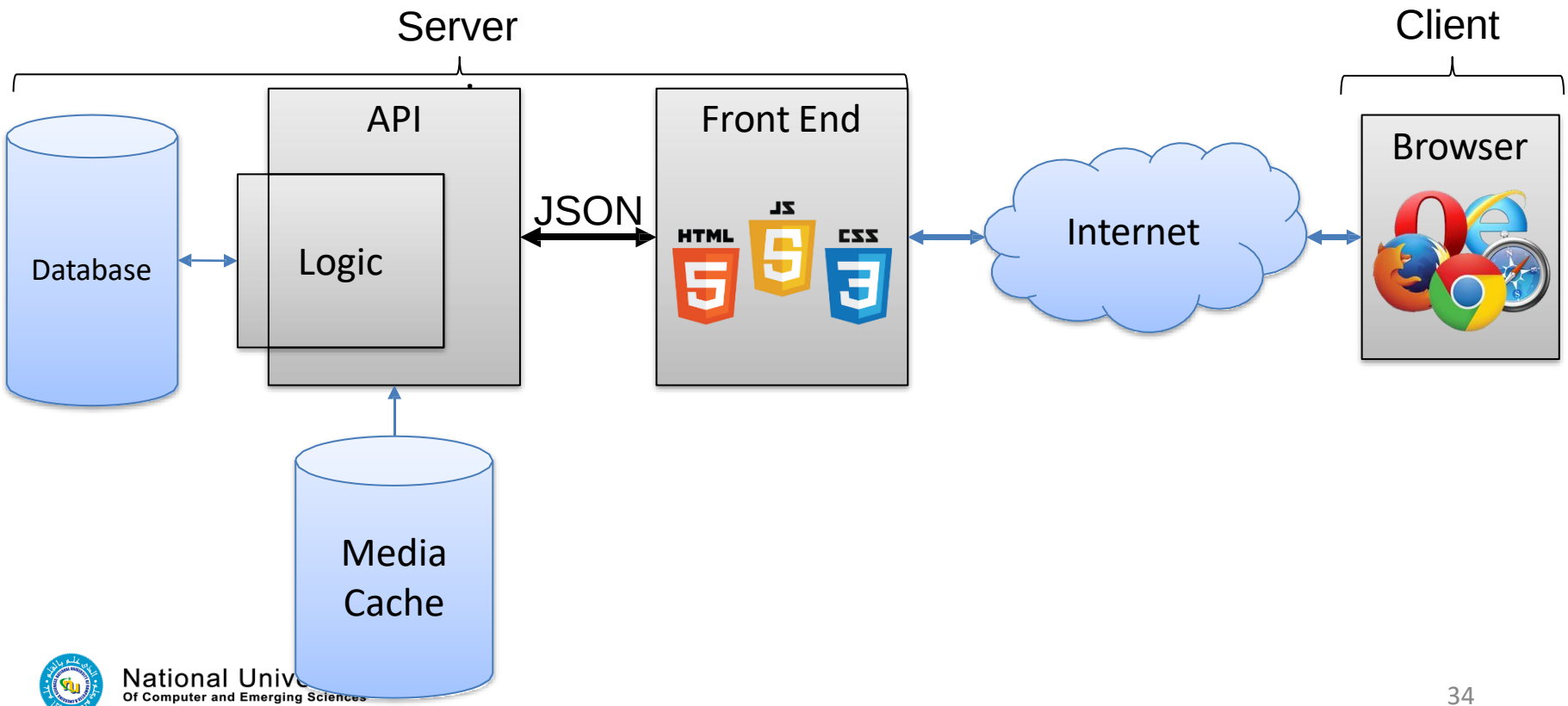


Principles of Web Design

- Availability
- Performance
- Reliability
- Scalability
- Manageability
- Cost

Web Applications

- UI (Front End (DOM, Framework))
- Request Layer (Web API)
- Back End (Database, Logic)



FRONTEND DEVELOPMENT

National University
Of Computer and Emerging Sciences

Front End Languages

- HTML/CSS
- JavaScript
- Java (applets)

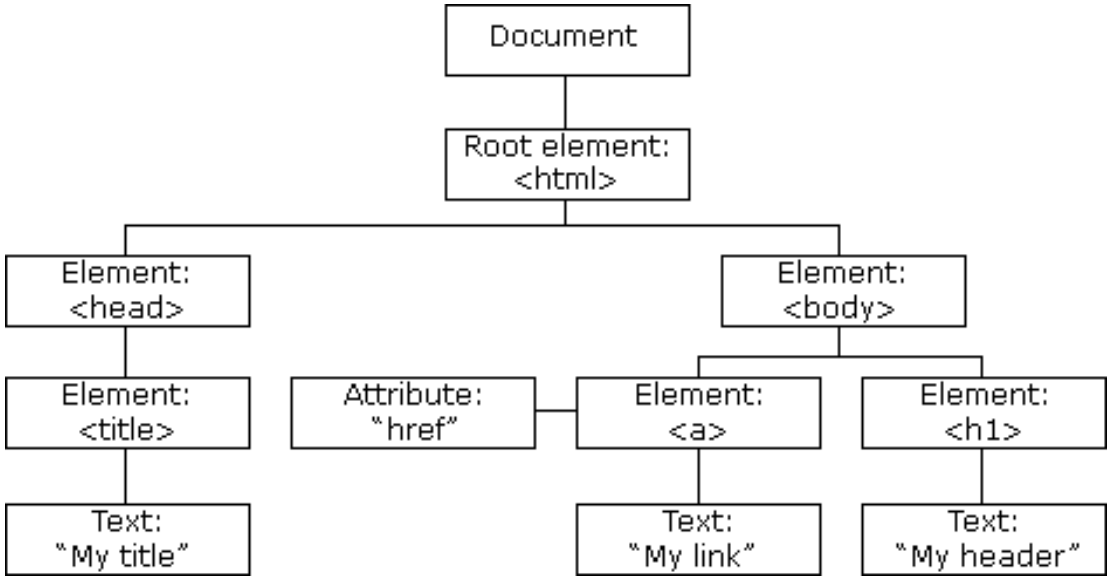
- What is the most popular?
- Answer: JavaScript/HTML/CSS is the only real option for front-end native languages and is basically the standard. But there are many variations on JavaScript that are used.

2009

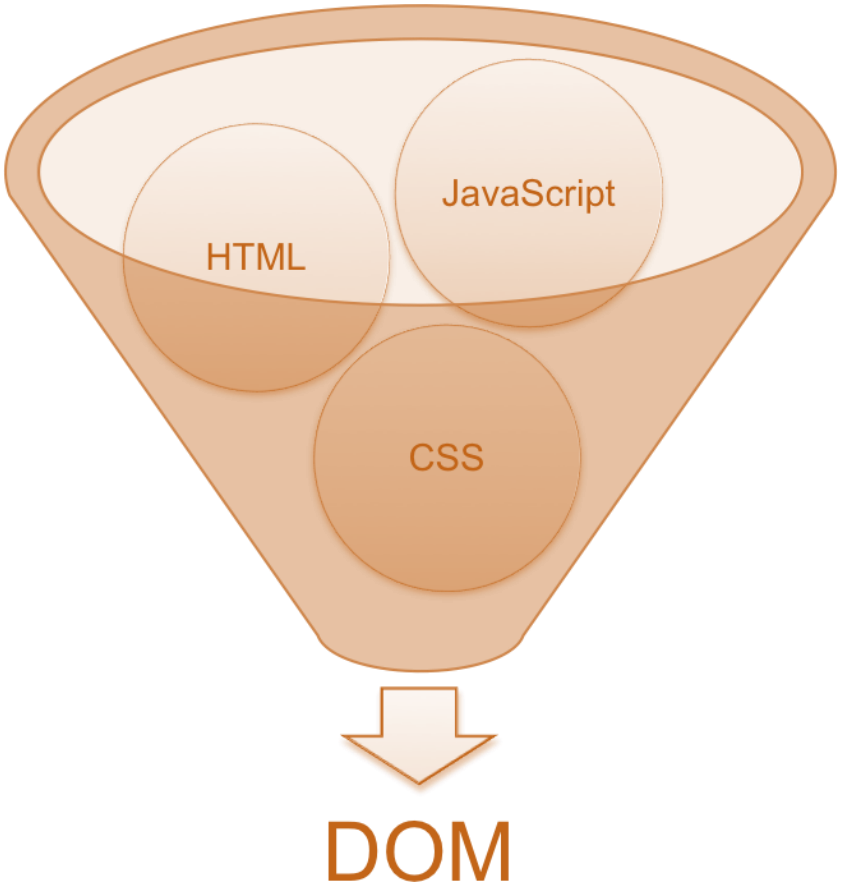
Today

DOM (Document Object Model)

- Document Object Model makes every addressable item in a web application an Object that can be manipulated for color, transparency, position, sound and behaviors.
- Every HTML Tag is a DOM object



DOM (Document Object Model)



Three representations of same page

Software Architectural Pattern and Framework

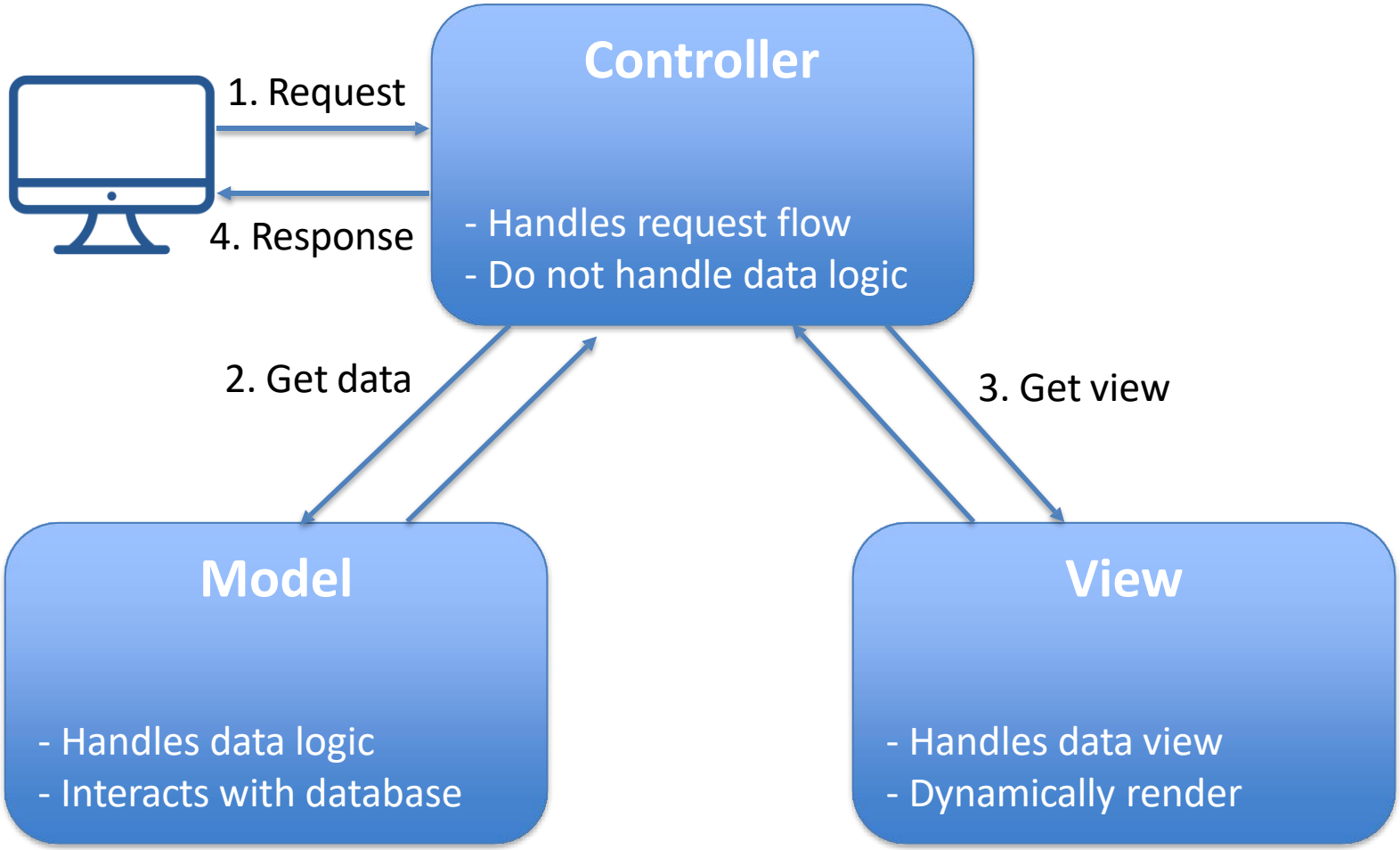
- Software Architectural Pattern
  - An architectural pattern expresses a fundamental structural organization schema for software systems
  - It provides a set of predefined subsystems, their responsibilities, and includes rules and guidelines for organizing the relationships between them

- Software Framework
  - A platform for developing software applications
  - It provides a foundation on which software developers can build programs for a specific platform
  - May include predefined classes and functions or APIs that can be used to process input, manage hardware/software components, etc.

MVC (Model View Controller)

- Software architectural design pattern
- Designed in 1970
- Originally developed for desktop applications
  - Most frequently used for web applications
- Separates application functionality
- Promotes modular programming

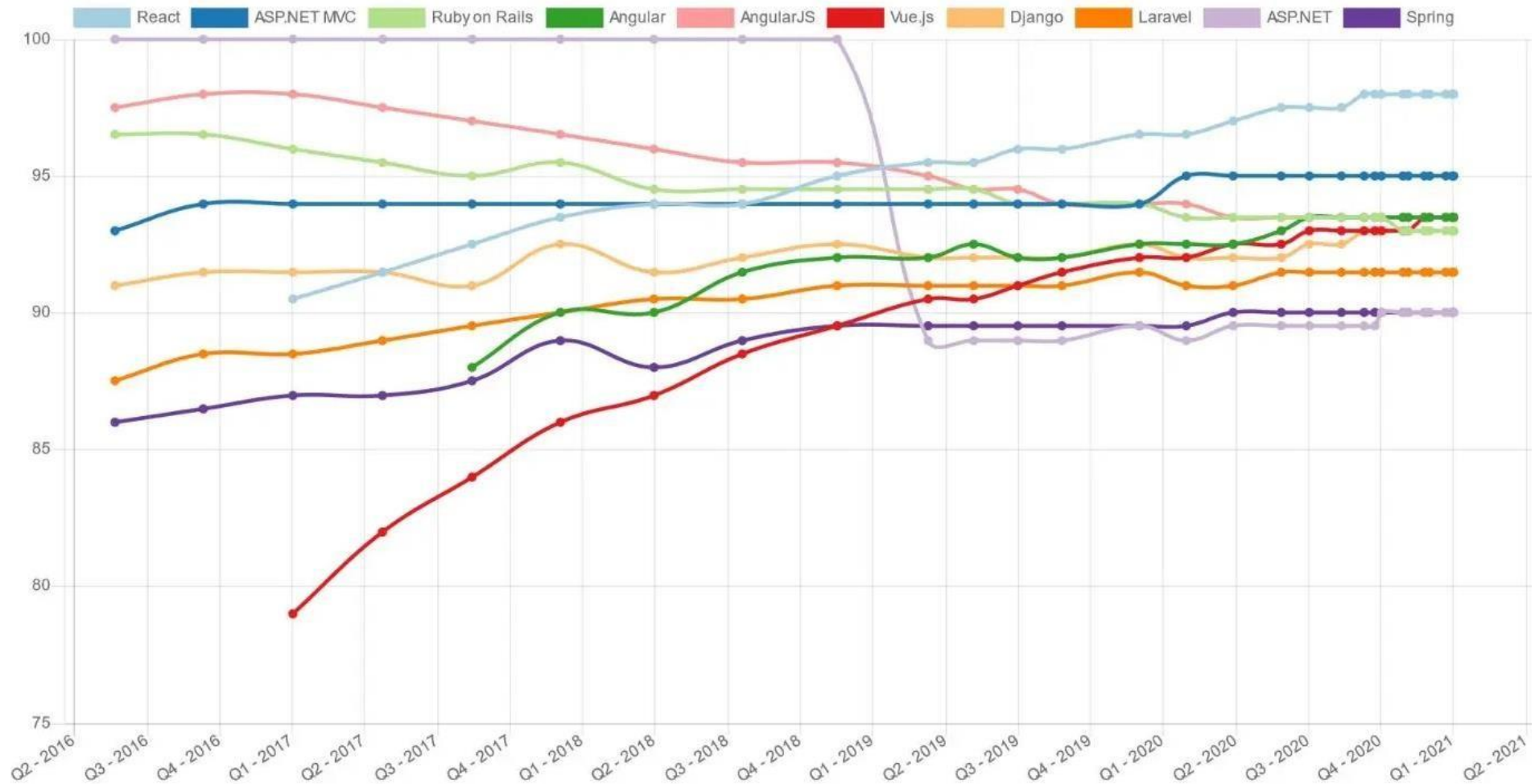
MVC Overview



Web Frameworks That Use MVC

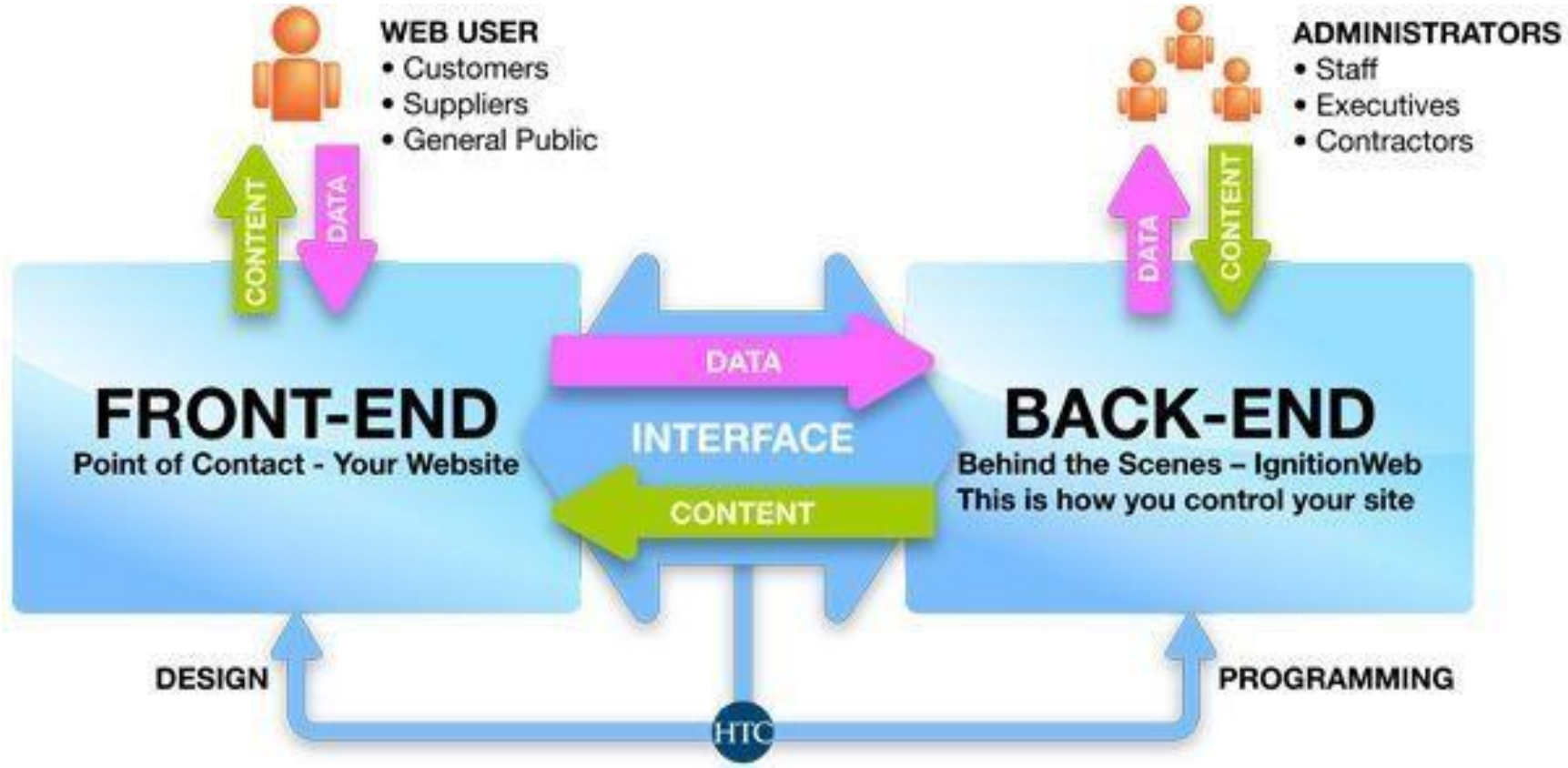
- Spring (Java)
- ASP.NET MVC (C#)
- Ruby on Rails (Ruby)
- Laravel (PHP)
- Django (Python)
- Flask(Python)
- AngularJS
- React (JS)
- Ember (JS)
- Express (JS)
- Vue (JS)
- Polymer (JS)

Popular frameworks



http://hotframeworks.com

BACKEND DEVELOPMENT



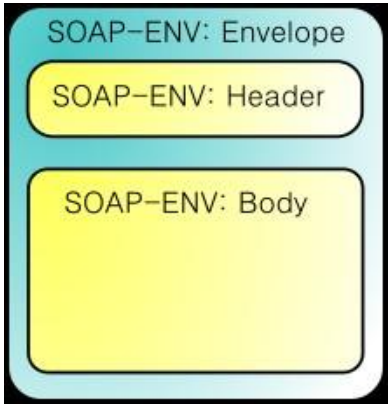
What is a Backend?

- All of the awesome that runs your application.
- Web API
  - Connection layer between the frontend and backend
  - Connected through API calls (POST, GET, PUT, etc.)
  - Transmit Content from the Backend to the Frontend commonly in JSON Blobs
- Service Architecture that drives everything (Where all the logic is)

Web APIs

- Application Programming Interface (API)
  - Serves as an interface between different software programs and facilitates their interaction
  - A particular set of rules and specifications that a software program can follow to access and make use of the services and resources provided by another particular software program
- Web API
  - The intermediate layer between front-end and back-end systems
  - Typically a defined set of HTTP request messages expressed in SOAP or REST along with a definition of the structure of response messages, typically expressed in JSON or XML.

SOAP



- Simple Object Access Protocol (SOAP)
- SOAP Introduction:
  - http://msdn.microsoft.com/en-us/library/ms995800.aspx
- A SOAP message is an ordinary XML document containing the following elements:
  - An Envelope element that identifies the XML document as a SOAP message
  - A Header element that contains header information
    - contains application-specific information about the SOAP message
      - optional
      - must be the first child element of the Envelope element
  - A Body element that contains call and response information
  - A Fault element containing errors and status information

SOAP

```
<soap:Envelope
  xmlns:soap="https://schemas.xmlsoap.org/soap/envelope/">
 <soap:Header> <!-- optional -->
   <!-- header blocks go here... -->
 </soap:Header>
 <soap:Body>
   <!-- payload or Fault element goes here... -->
 </soap:Body>
</soap:Envelope>
```

REST

- Representational State Transfer (REST)
  - Use HTTP method to invoke remote services (not XML)
- The response of remote service can be in XML/JSON or any textual format
- Benefits:
  - Easy to develop
  - Easy to debug (with standard browser)
  - Leverage existing web application infrastructure

```
GET /articles/23 HTTP/1.1
Accept: text/html, application/xhtml
```

```
HTTP/1.1 200 (OK)
Content-Type: text/html
```

Server Responses

- JavaScript Object Notation (JSON)
  - Lightweight data-interchange format
  - Human readable and writable and also machine friendly
  - Wide support from most languages (Java, C, C#, PHP, Ruby, Python…)

```
1 {
2 "firstname":"Hassan",
3 "lastname":"Sartaj"
4 }
```

Json: 48 bytes

```
1 <person>
2 <first-name>Hassan</first-name>
3 <last-name>Sartaj</last-name>
4 </person>
```

XML: 83 bytes

Thank You!