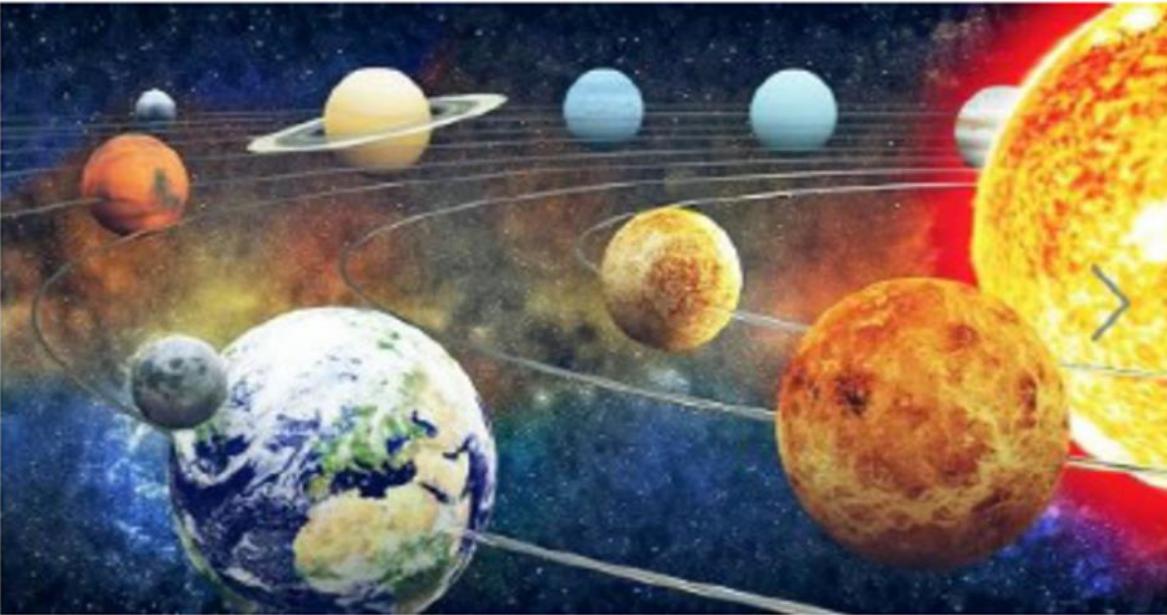


Starting in the name of Allah,

*the most beneficial,
the most merciful.*



۲۳
تہنیٰ

کیا انسان کو ہر وہ چیز حاصل ہے جس کی اس نے تمنا کی؟



UNIVERSITY OF
KARACHI

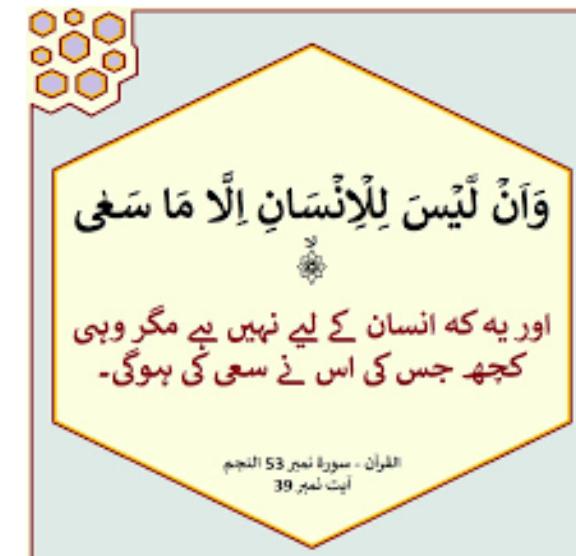
Surah An-Najm Chapter 53 Verse 39

اور یہ کہا سان گروہی ملنا ہے جس کی دُکوٹش کر رہے

(القرآن ۵۳:۳۹)



And there is not for man except that [good] for which he strives.



UNIVERSITY OF
KARACHI

Read, Read, Read..... Practice, Practice, Practice

<https://www.geeksforgeeks.org/difference-between-internet-and-www/>

<https://cs.lmu.edu/~ray/notes/webapps/>

<https://codesandbox.io/p/sandbox/>

<https://sandilands.info/sgordon/teaching/its323y07s1/protected/ITS323Y07S1L18-InternetApplications.pdf>

<https://www.geeksforgeeks.org/difference-between-web-application-and-website/>

<https://bitbytesoft.com/11-different-types-of-web-applications/>

<https://www.theengineeringknowledge.com/full-form-of-internet/>

<https://datatracker.ietf.org/doc/html/rfc2151>

Who am I?

Humera Tariq

PhD, MS, MCS (Computer Science), B.E (Electrical)

Postdoc (Medical Image Processing, Deep Neural Networks)

Email: humera@uok.edu.pk

Web: <https://humera.pk/>

Discord: <https://discord.gg/xeJ68vh9>

Week 01-Updated

Internet Application Development



Copyright © 2025, Humera Tariq

*Department of Computer Science (DCS/UBIT)
University of Karachi
January 2025*

Before we formally start! DCS Alumni success story

Jazakallah" (جزاک اللہ) to DCS/UBIT Alumni for sharing resources and offering us license to the paid web development course.



<https://www.codewithahsan.dev/courses/web-dev-basics>

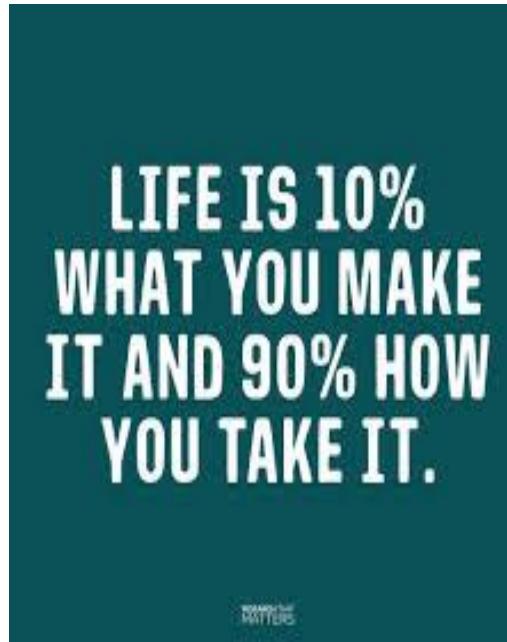


How to Access the Angular/react Course? → IOMECHS will reach us soon!



Grading

7



Share if you have a
better plan

Concepts/practice/In-class discussions	35%
Team project	20% + Bonus
Challenge Time: Team A vs Team B	20%
Final exam	25%
Total	100%

Progress Tracking/ Meeting Plans

- (1) Reading material and resources will be posted on Discord.
- (2) You are supposed to demonstrate learning by showing up in front of the class and participating in Discord individually and as a team.

~ Total Weeks = 12-15 No. of Students per section= ~ 80

~Team size = $80/8 = 10$ persons ~Team count = 8

Weekly lab meet-up = 2 teams => 20 students

Round 1 Progress: 5 weeks

Round 2 Progress: 10 weeks

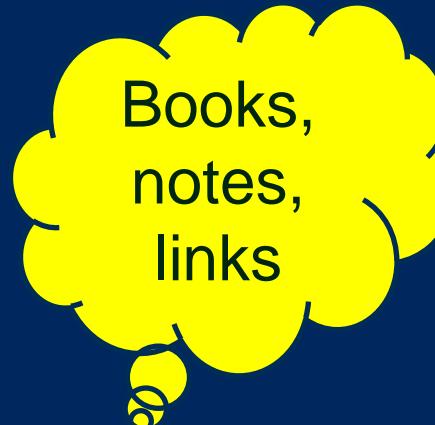
Course Objectives

At the end of this course, you will be able to:

- 1) Web page **structure and appearance** with HTML5 and CSS
- 2) Client-side **interactivity and behavior** with JavaScript
- 3) Using web services (APIs) **on the front-end** with asynchronous JavaScript
- 4) Writing web services with **server-side** JavaScript via Node.js
- 5) Storing and retrieving information in a **database** with SqLite

Resources

Internet Application Development



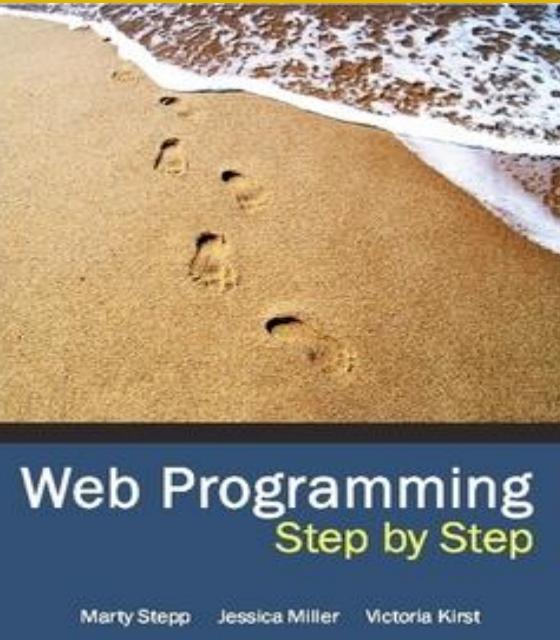
Books,
notes,
links

Copyright © 2025, Humera Tariq

Department of Computer Science (DCS/UBIT)

University of Karachi

January 2025



- [Chapter 2: HTML Basics](#) (excerpt)
- [Chapter 3: CSS for Styling](#) (excerpt)
- [Chapter 6: Forms](#) (excerpt)
- [Chapter 7: Web Design](#) (excerpt)
- [Chapter 14: Cookies and Sessions](#) (excerpt)
- [Chapter 15: Web Security](#) (excerpt)
- [Appendix A: Syntax Reference](#) (excerpt)

Browser: Document, Events, Interfaces



Table of contents

Main course contains 2 parts which cover JavaScript as a programming language and working with a browser. There are also additional series of thematic articles.

PART 1

The JavaScript language

PART 2

Browser: Document, Events, Interfaces

PART 3

Additional articles

Available
on discord
resources

[The Modern JavaScript Tutorial](#)

[The Modern JavaScript Tutorial BOM UI part 2.pdf](#)

[The Modern JavaScript Tutorial part 3.pdf](#)

[The Modern JavaScript Tutorial. additional.pdf](#)

Background/Context

Internet Application Development

INTERNET

Internet is a global network of networks.

Internet is a means of connecting a computer to any other computer anywhere in the world.

Internet is infrastructure.

Internet can be viewed as a big book-store.

At some advanced level, to understand we can think of the Internet as hardware.

Internet is primarily hardware-based.

It originated sometimes in late 1960s.

Internet is superset of WWW.

The first version of the Internet was known as ARPANET.

Internet uses IP address.

It consists of interconnected computers, copper wires, fiber optics, and wireless system.

WWW

WWW stands for World wide Web.

World Wide Web which is a collection of information which is accessed via the Internet.

WWW is service on top of that infrastructure.

Web can be viewed as collection of books on that store.

At some advanced level, to understand we can think of the WWW as software.

WWW is more software-oriented as compared to the Internet.

English scientist Tim Berners-Lee invented the World Wide Web in 1989.

WWW is a subset of the Internet.

In the beginning WWW was known as NSFNET.

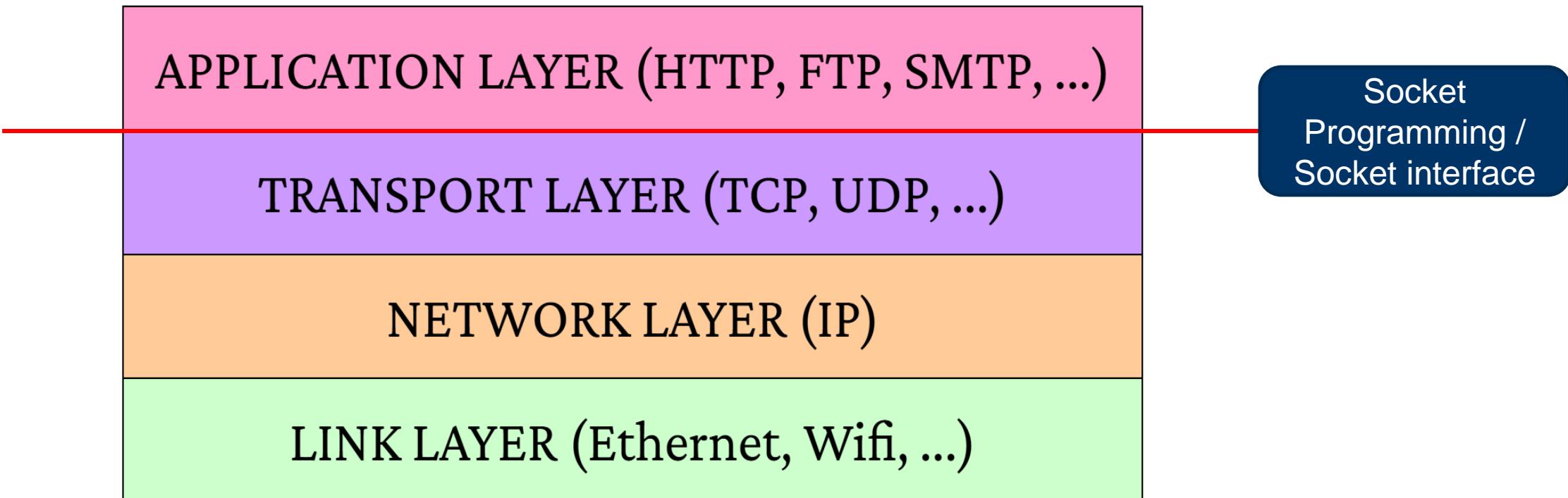
WW uses HTTP.

It comprises documents, files, and folders saved in various linked computers.

Review of Internet Layers

All Internet applications work over the exact same transport layers. The Internet says nothing about how these applications should work. It provides IP, TCP, and UDP, and that's it.

You can build anything on top of the Network & Transport Layer.



Internet Applications pretty much just need to know:

1) the **IP address** of the other party (what *host* the other party is running on—a network layer concept),

(2) the **port number** of the application running at the other end (because the other machine might be running multiple services—a transport layer concept).

The application passes those two pieces of information to the transport layer to make the communication happen.

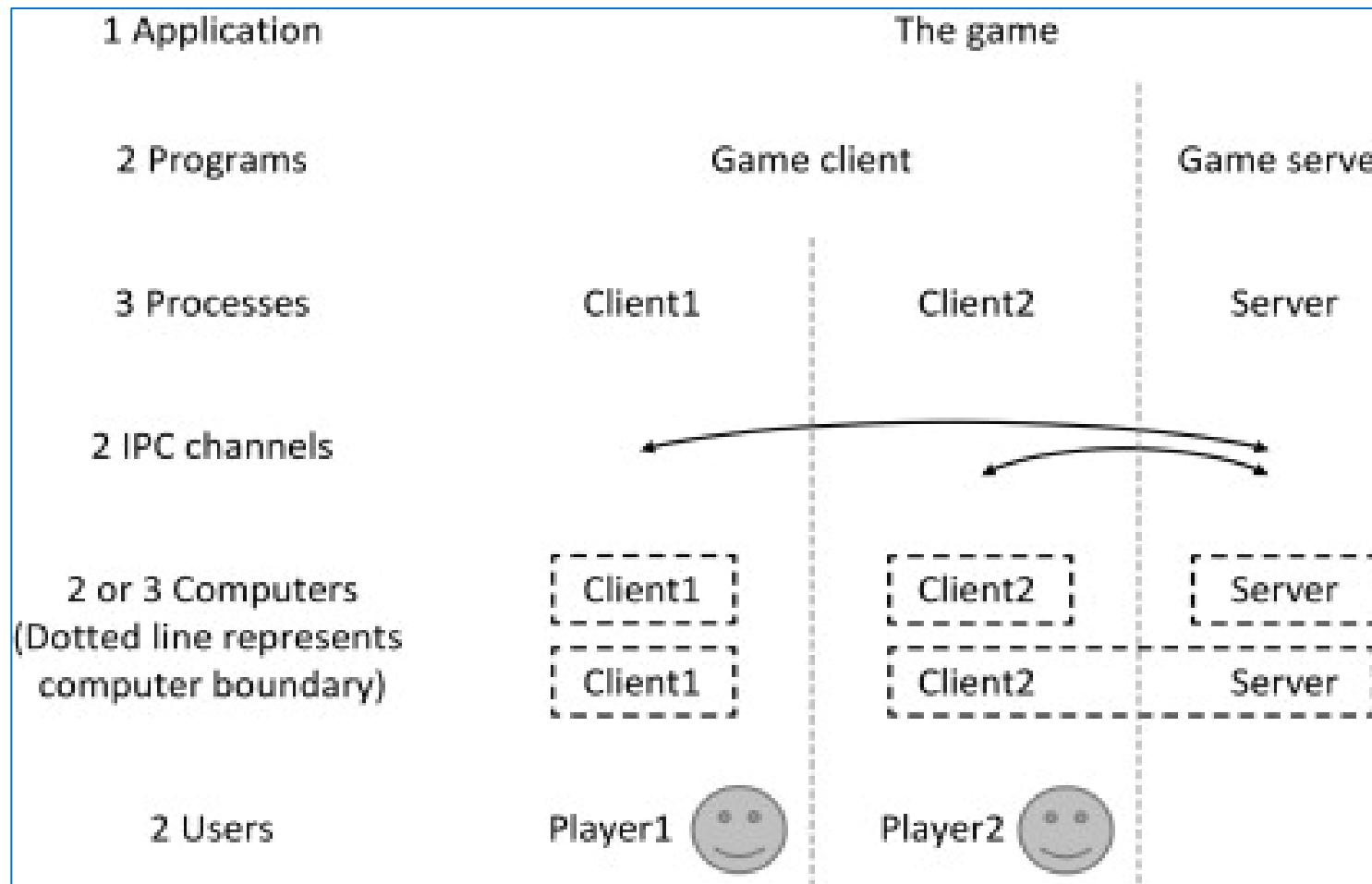
So, hosts have IP addresses and applications run on specific ports.

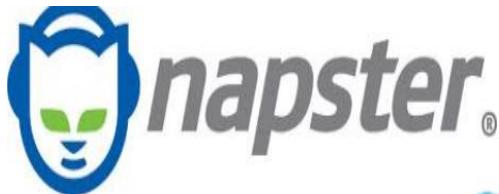
Writing Your Own Internet Applications

<https://www.sciencedirect.com/topics/computer-science/client-server-paradigm>

Current IoT solutions mostly rely on the centralized client-server paradigm where all the devices are identified, authenticated, and connected via cloud servers

Client Server paradigm





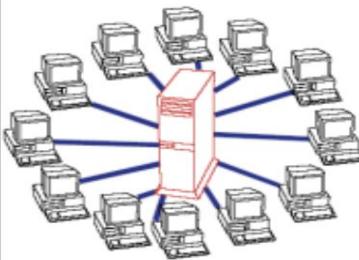
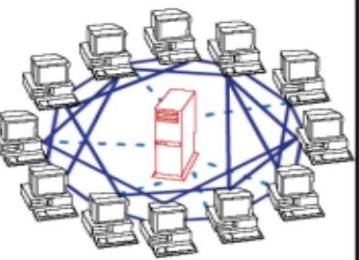
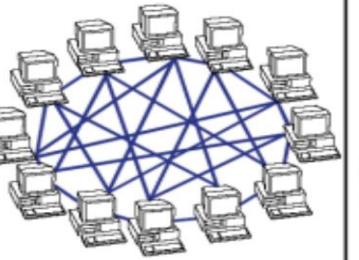
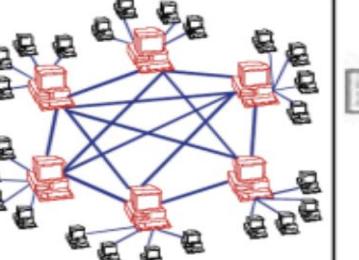
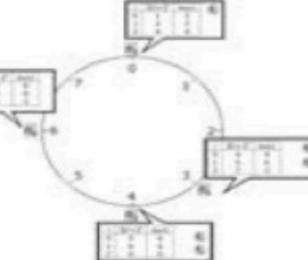
P2P paradigm

Peer-to-peer systems (P2P systems) represent a paradigm for the construction of distributed systems and applications in which data and computational resources are contributed by many hosts on the Internet.

P2P systems enable the sharing of data and resources on a very large scale by eliminating any requirement for separately managed servers and their associated infrastructure.

P2P systems have been used to provide **file sharing, web caching, information distribution and other services**, exploiting the resources of tens of thousands of machines across the Internet.

Summary of the characteristic features of Client-Server and P2P

Client-Server	Peer-to-Peer			
	Unstructured P2P		Structured P2P	
	1st Generation	2nd Generation		
<p>1. Server is the central entity and only provider of service and content. → Network managed by the Server</p> <p>2. Server as the higher performance system.</p> <p>3. Clients as the lower performance system</p> <p>Example: WWW</p>	<p><i>Centralized P2P</i></p> <p>1. All features of Peer-to-Peer included</p> <p>2. Central entity is necessary to provide the service</p> <p>3. Central entity is some kind of index/group database</p> <p>Example: Napster</p>	<p><i>Pure P2P</i></p> <p>1. All features of Peer-to-Peer included</p> <p>2. Any terminal entity can be removed without loss of functionality</p> <p>3. → No central entities</p> <p>Examples: Gnutella 0.4, Freenet</p>	<p><i>Hybrid P2P</i></p> <p>1. All features of Peer-to-Peer included</p> <p>2. Any terminal entity can be removed without loss of functionality</p> <p>3. → dynamic central entities</p> <p>Example: Gnutella 0.6, JXTA</p>	<p><i>DHT-Based</i></p> <p>1. All features of Peer-to-Peer included</p> <p>2. Any terminal entity can be removed without loss of functionality</p> <p>3. → No central entities</p> <p>4. Connections in the overlay are "fixed"</p> <p>Examples: Chord, CAN</p>
				

(Eberspächer, & Schollmeier 2005)

https://www.researchgate.net/publication/220829221_A_Content-Addressable_Network_for_Similarity_Search_in_Metric_Spaces#fullTextFileContent

Distributed Applications: browser execution + Server execution + DB execution

Descriptive models for distributed system design

Physical model

Architectural model

Architectural elements

Communicating entities

Processes

Objects

Components

Web Services

Communication paradigm

Inter-process communication

UDP
sockets

TCP
sockets

Multi-
cast

Indirect
communication

Remote
invocation

Roles and responsibilities

Architectural styles

Client-server

Peer-to-peer

Placement

Multiple server

Proxy/Cache

Mobile code

Interaction model

Interaction model

Failure model

Security model

Architectural patterns

Vertical distribution

Multi-tier

Thin/Fat Client

Horizontal distribution

Web site – Web app – Web Service

What is the difference?

Check out the full Wikipedia articles on [web applications](#) , [web servers](#), and [web services](#).

Early Internet Applications!

A Primer on Internet TCP/IP Tools and Utilities 1997

Application	RFC	Port	Description
Daytime	867	13	When a client connects, the server sends back a string with the current date and time, then immediately closes the connection. (From the RFC: “There is no specific syntax for the daytime. It is recommended that it be limited to the ASCII printing characters, space, carriage return, and line feed. The daytime should be just one line.”)
Time	868	37	When a client connects, the server sends back a 32-bit time value, then immediately closes the connection. (From the RFC: “The time is the number of seconds since 00:00 (midnight) 1 January 1900 GMT, such that the time 1 is 12:00:01 am on 1 January 1900 GMT; this base will serve until the year 2036.”)
Quote of the Day	865	17	When a client connects, the server sends back a short message then immediately closes the connection. (From the RFC: “There is no specific syntax for the quote. It is recommended that it be limited to the ASCII printing characters, space, carriage return, and line feed. The quote may be just one or up to several lines, but it should be less than 512 characters.”)

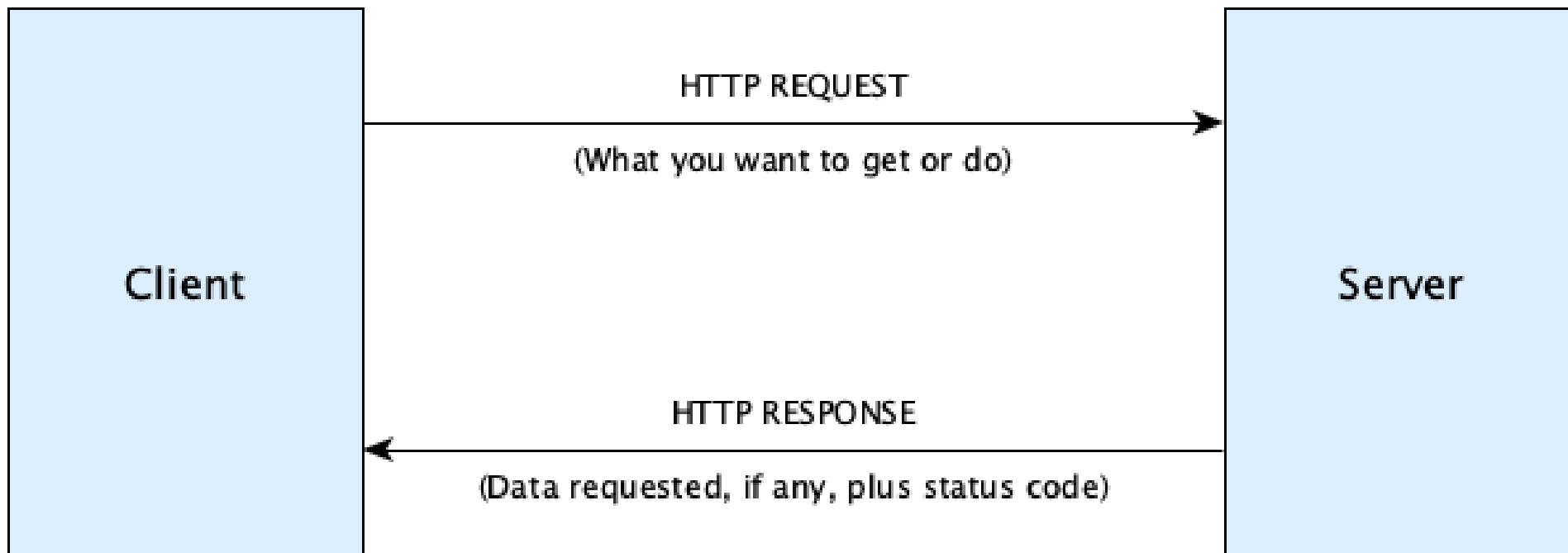
DNS	1035	53	Clients ask the name service questions, usually of the type "what is the ip address for this domain name?" and the server responds with an answer of some sort.
Trivial File Transfer	1350	69 udp	TFTP. Much weaker than FTP, but is lightweight and simple. The sender and receiver exchange files packet by packet in lock step, with acknowledgements being part of the protocol itself.
Simple Mail Transfer	2821	25	SMTP. A very simple protocol for sending email.
Mailbox Access with POP3	1939	110	A simple protocol for managing email.
Mailbox Access with IMAP	3501	143	A modern alternative to POP3. IMAP clients can stay connected for longer times than POP3 clients, can have multiple clients attached to the same mailbox simultaneously, keep state on the server, fetch partial messages, and do other cool things.
File Transfer	959	21	A fully featured file transfer application using the File Transfer Protocol, FTP. Server runs on port 21 with data transferred through port 20.

World Wide Web	7230-7235	80	Servers usually run on port 80 (clear) or port 443 (secure) but really could run anywhere. Clients request resources via a uniform resource identifier (URI) and the server responds to the request. Request and response structures are quite detailed. The resources can be absolutely anything (so the responses usually contain a media type in their header).
News	3977	119	Network News Protocol, NNTP. Used for the reading and writing news articles structured into newsgroups.
Telnet	854	23	A very, very, generic communication protocol.
Secure Shell	4250-4256	22	A protocol for secure remote login, file transfer, and more.

Web Application Program

A **web application** is a program that runs on a computer with a **web server** and is accessed via a **web browser** (the “client”) that communicates with the server via the HTTP or HTTPS protocol.

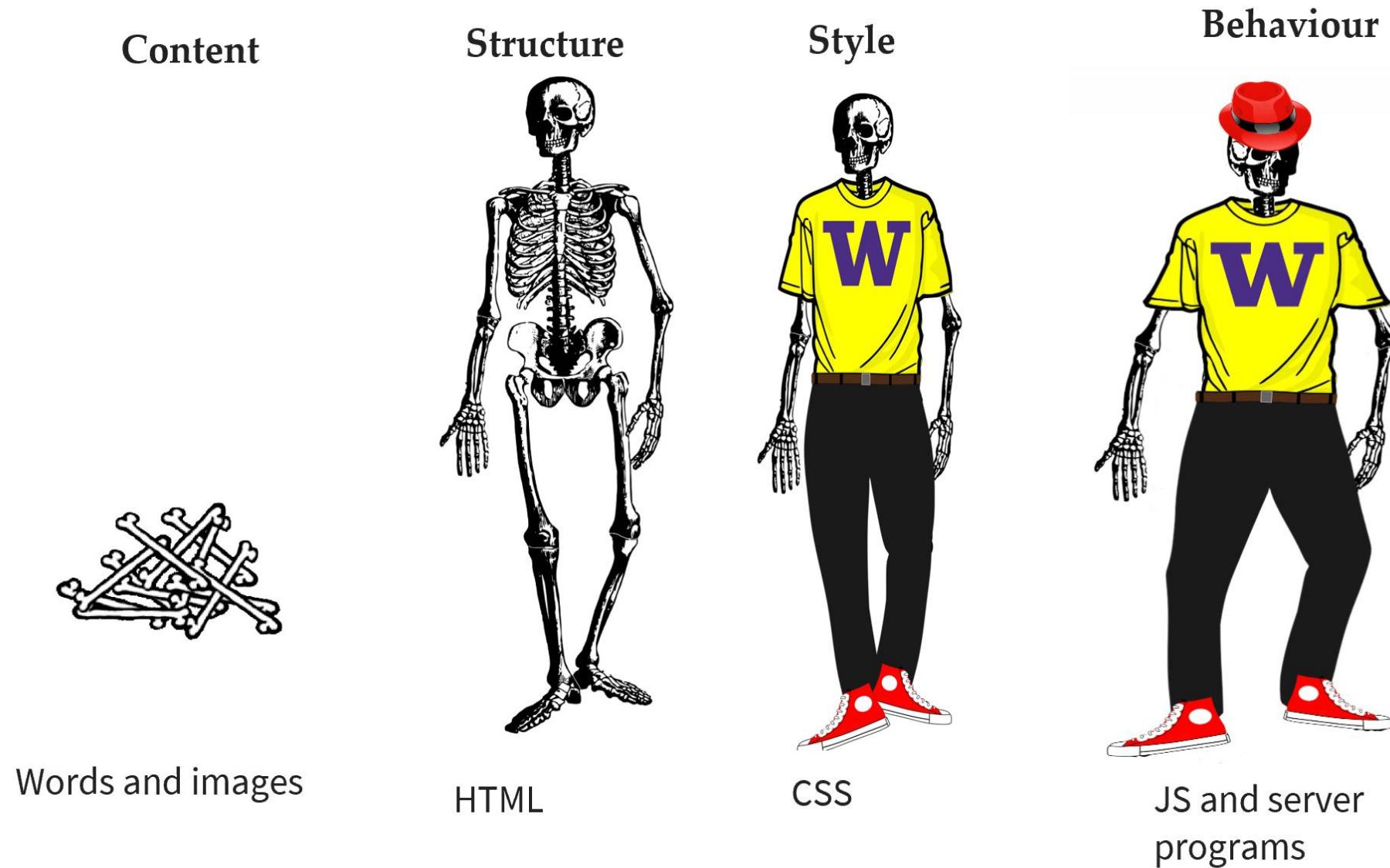
In HTTP(S), the client sends a **request** to the server and the server replies with a **response**:



Website - Web app - WebService

- ✓ If the server always sends back the same data in the form of **HTML**, **CSS**, **images**, **videos**, and similar resources *without any user interaction*, we call the app a **web site**.
- ✓ If there is user interaction (**mouse clicks**, **tapping and swiping**, **phone shaking**, **keypresses**, etc.) that affect the execution of the program, we have a **web app**.
- ✓ If the server only produces raw data (**generally in text or JSON**), then we speak of a **web service**.

Web site – Web apps – web service



A **layout plan** helps you determine your site's usability and navigation, which affect **user experience**. Some elements to consider when planning include the sites' **header**, **footer**, and **navigation**. You can use pen and paper or web design software such as **Figma** to design a website layout. It doesn't have to be detailed as long as it roughly represents the look and feel of your site.

Website title

Navigation Navigation Navigation

Newest Post

First Post

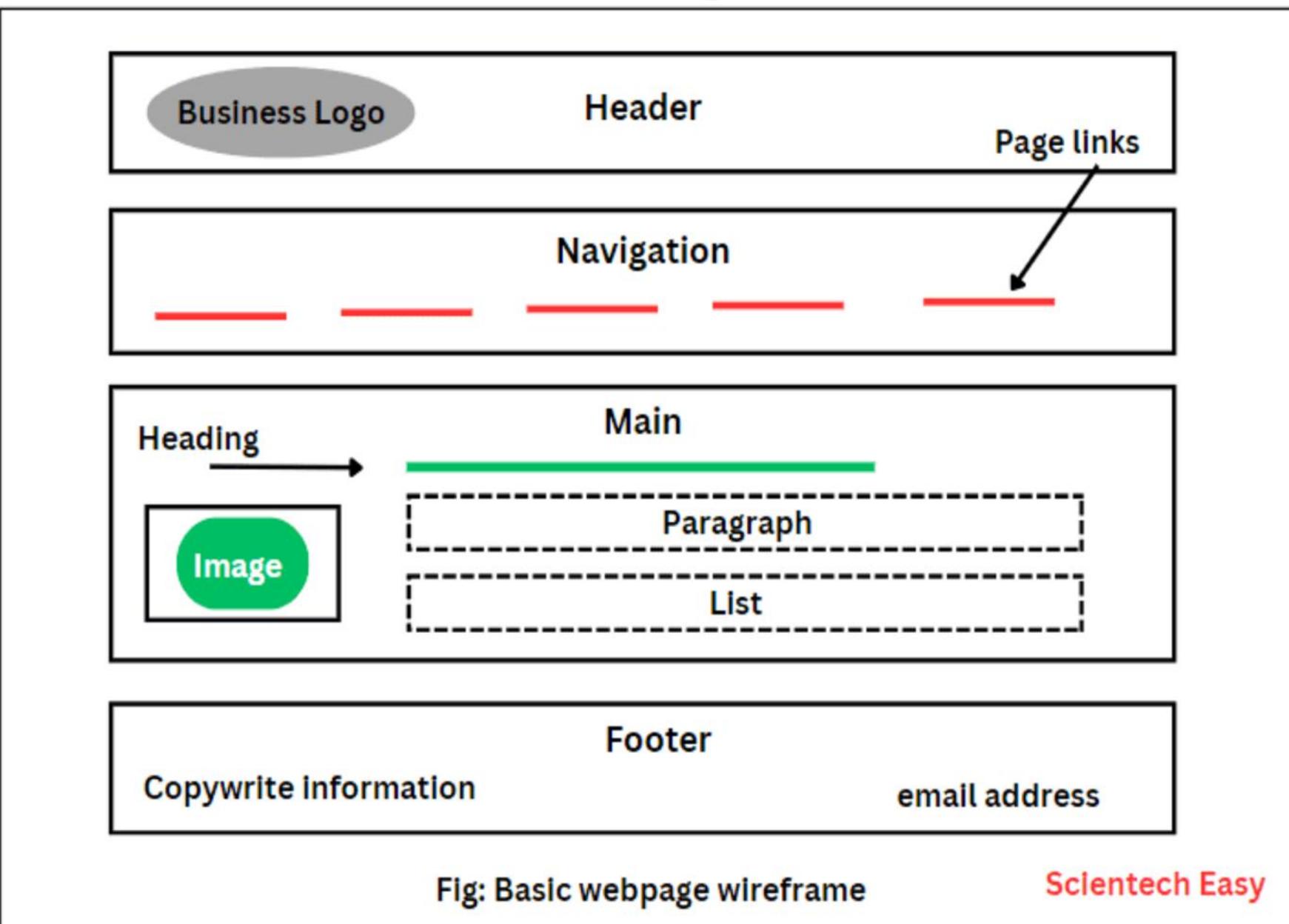
Content goes here

About Me

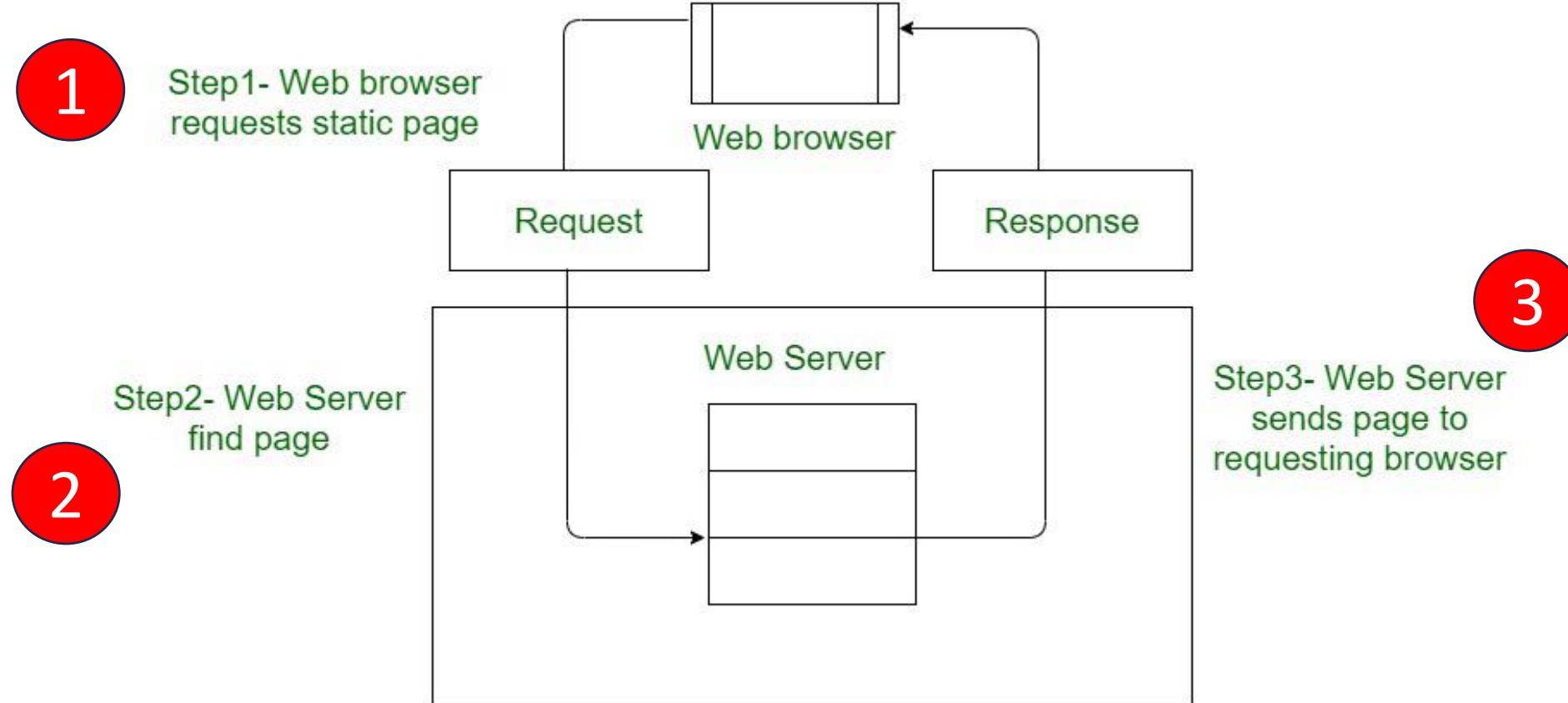
Author description goes here

Social-media Social-media Social-media

Basic Web Page WireFrame

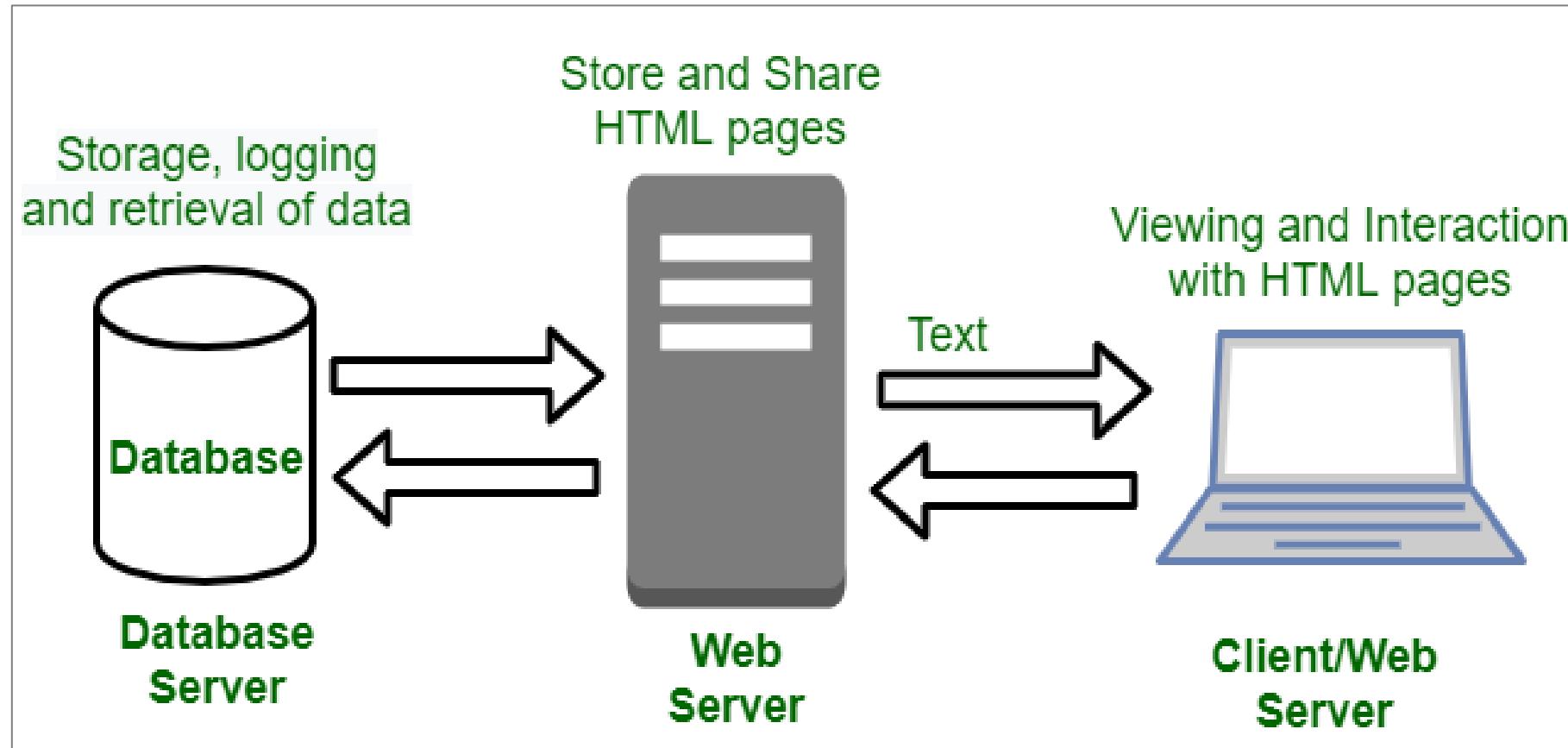


Architecture of Static Website

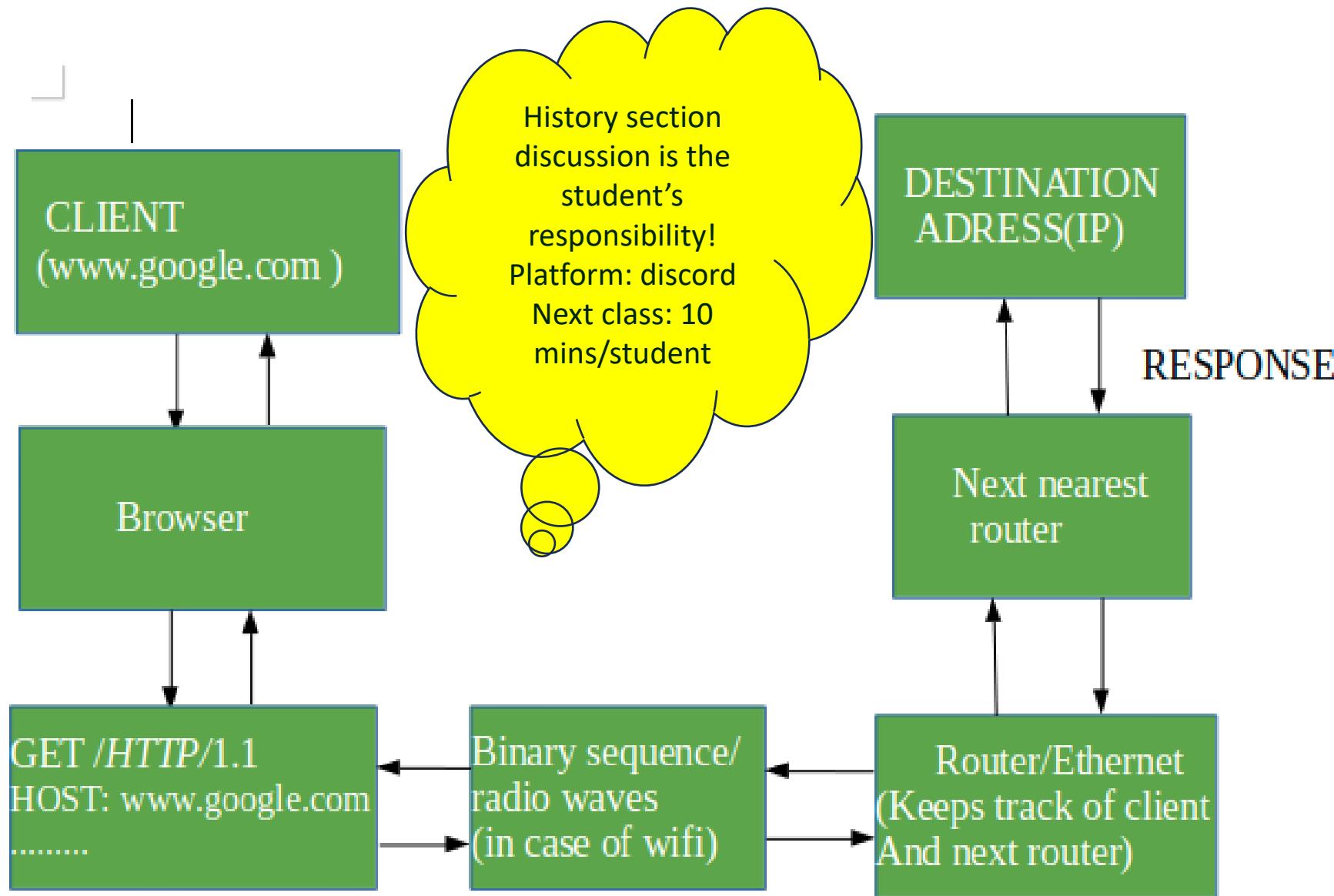


Note: Static does not mean that it will not respond to user actions,

Architecture of Dynamic Website



How Does Internet and Web Programming Work?



Client-side

- ✓ First, when we type a URL like **www.google.com**, the browser converts it into a file containing
 - GET /HTTP/1.1 (where GET means we are requesting some data from the server and HTTP refers to a protocol that we are using, 1.1 refers to the version of the HTTP request)
 - Host: www.google.com
 - And some other information
- ✓ Now this file is converted to binary code by the browser and it is sent down the wires if we are connected through Ethernet and if we are using WiFi, first it converts it to a radio signal which is decoded by a router in a very low level. It is converted to binary and then sent to the servers. This information or ‘binary codes’ go to the destination and responds if it is received by the sender only because of the IP address. One router will send the information to another and this keeps on going until the binary codes reach the destination.

Server-side

✓ Now the server receives the binary code and decodes it and sends the response in the following manner:

- HTTP/1.1 200 ok (where 200 ok is the status)
- Content-type : type/HTML
- Body of page

Now, this is converted back to binary by the server and sent to the [IP address](#) that is requesting it. Once the codes are received by the client, the browser again decodes the information in the following way

- First, it checks the status
- It starts reading the document from the HTML tag and constructs a Tree-like structure.
- The HTML tree is then converted to corresponding binary code and rendered on the screen.
- In the end, we see the website front-end.
- Soon we will understand the tree structure of the HTML document

The future of the internet?



Now it is time for..... !

Writing and Deploying Simple Web Apps

THERE'S NO
ONE RULE
FITS ALL, NOT
IN ART, NOR
IN LIFE

<https://hackr.io/documents/webdev-ebook.pdf>

Preface

“Most good programmers do programming not because they expect to get paid or get adulation by the public, but because it is fun to program.”

- Linus Torvalds

Creator of Linux Operating System

Where to start & Where should stop!

Data in academia



Data in the real world



Angular or React?



INSTAGRAM



WHATSAPP



GENERAL MOTORS



UPWORK



FACEBOOK



AIRBNB



GOOGLE



HBO

Uber

UBER

NETFLIX

NETFLIX



Forbes

NIKE

FORBES



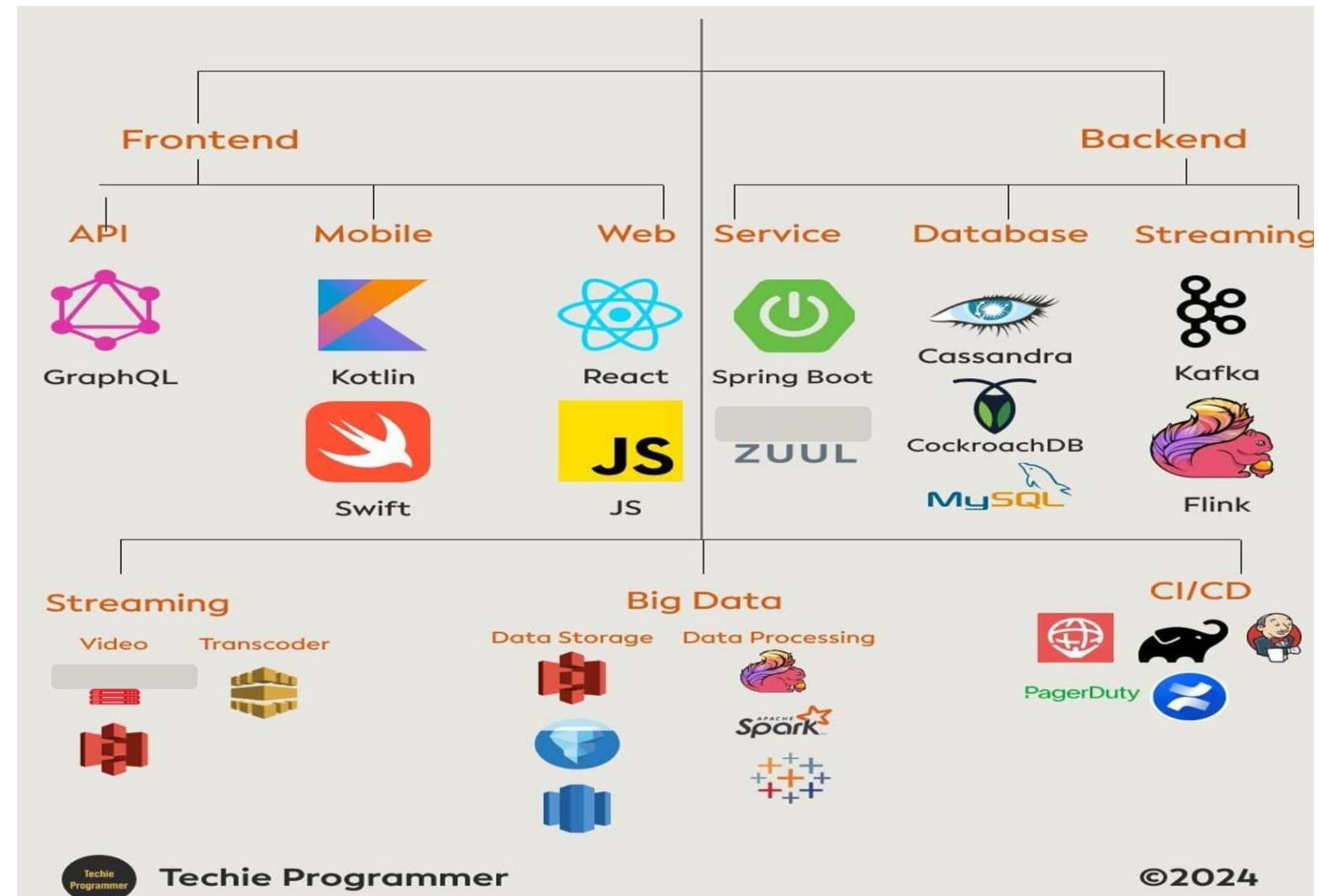
DROPBOX

SONY

SONY

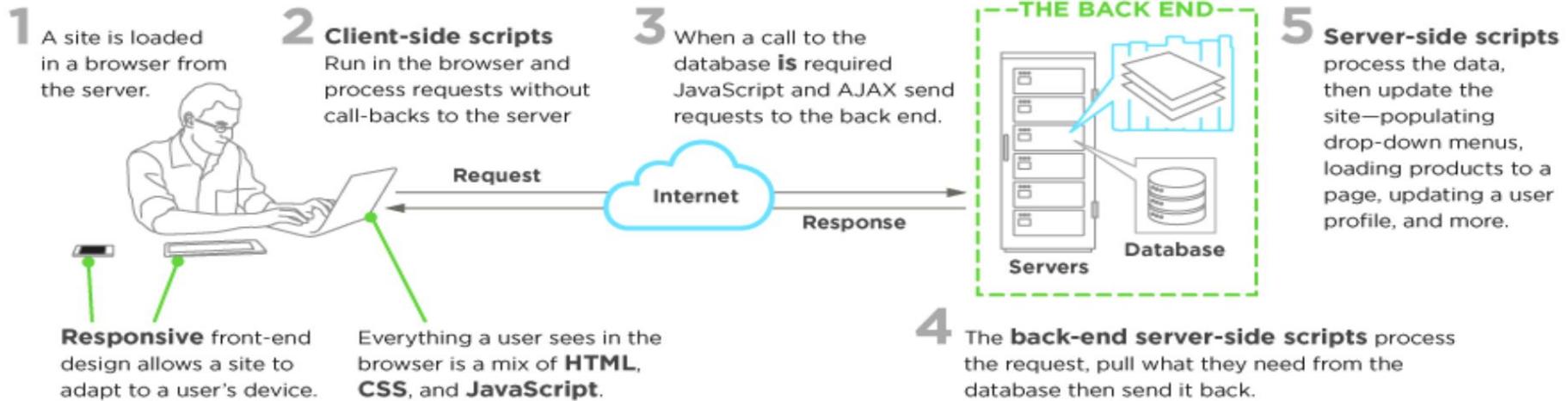
A web application is a software program that runs on a web server and is accessed by users through a web browser. Unlike traditional desktop applications, web applications do not need to be installed on the user's device and can be used on any device with a compatible browser. Web applications typically involve a combination of frontend and backend technologies to deliver interactive, dynamic, and responsive user experiences.

Example Real world Architecture



3 web development specialties

FRONT-END DEVELOPMENT



2- Backend development: The “backend” is like the portion of the iceberg below the surface. Without it, the website can’t function. The backend deals with things like servers, applications, and databases.

3- Full-stack development: This is the combination of both backend and frontend development.



UNIVERSITY OF
KARACHI

Popular Programming Languages 2000--2023

2000



Client-side Language?

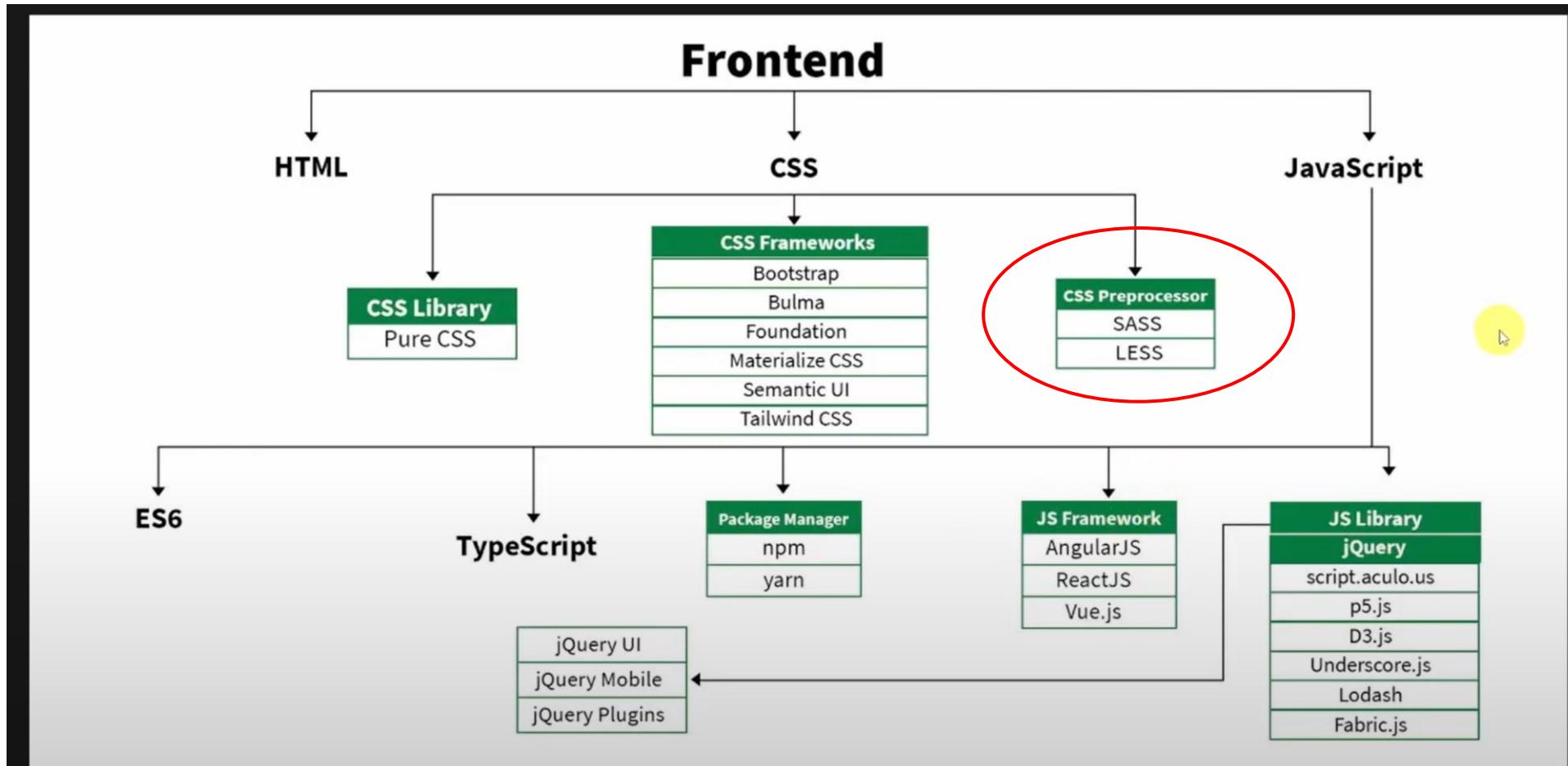
- ✓ Single threading
- ✓ Limited precision of numbers
- ✓ Limited multithreading
- ✓ Limited concurrency
- ✓ Limited low-level operations
- ✓ Lack of type safety
- ✓ Limited OOP support

Front end development overview

SAAS: Software as Service

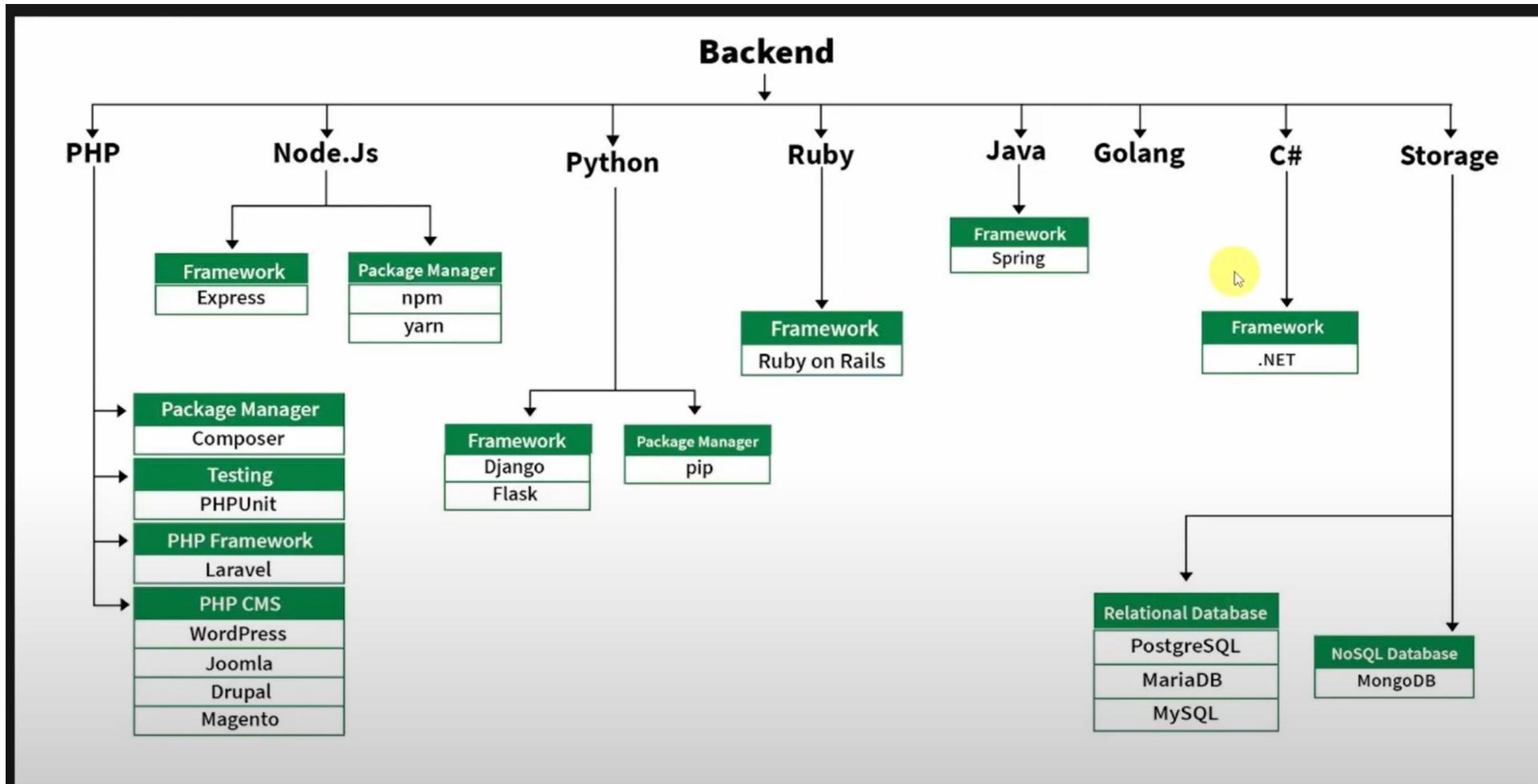
LESS: ???

<https://www.codewithahsan.dev/courses/web-dev-basics/what-is-web-development>



Back end development overview

<https://www.codewithahsan.dev/courses/web-dev-basics/what-is-web-development>



FRONT END DEVELOPMENT

HTML- The Skeleton of a webpage

CSS- Beauty of a webpage

JAVASCRIPT- Behaviour of a webpage

User Interface Design/User Experience Design

Responsive Web Design





Static Website	Dynamic Website
Content of Web pages can not be change at runtime.	Content of Web pages can be changed.
No interaction with <u>database</u> possible.	Interaction with <u>database</u> is possible
It is <u>faster</u> to load as compared to dynamic website.	It is <u>slower</u> than static website.
Cheaper Development costs.	More Development costs.
No feature of Content Management.	Feature of Content Management System (CMS).
HTML, CSS, Javascript is used for developing the website.	Server side languages such as PHP, Node.js are used.
Same content is delivered everytime the page is loaded.	Content may change everytime the page is loaded.

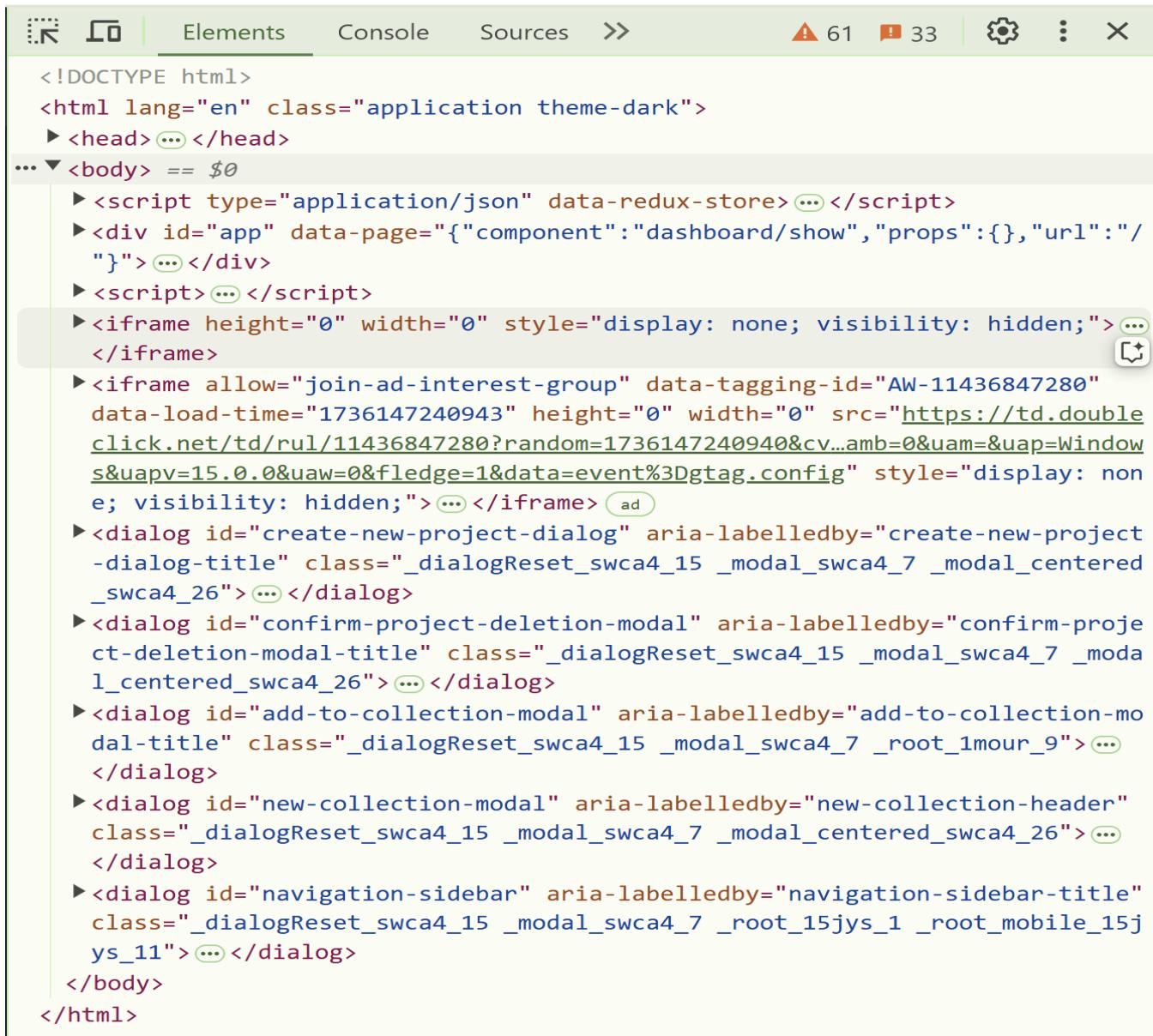
Can we identify and name arbitrary html elements on this web page ??

The screenshot shows the StackBlitz homepage. At the top, there is a dark header with a search bar containing "Search projects...", a lightning bolt icon, and navigation links for "Docs", "Enterprise", "Pricing", "More", and a "NEW bolt.new" button. Below the header, a sidebar on the left shows a user profile "divacomputing" (Free plan) and links for "Dashboard", "Projects", "Collections", and "Settings". In the center, there is a "New project" button with the text "from one of StackBlitz starter templates" and a preview image showing various programming logos like Angular, Node.js, React, and Vue.js. To the right, a large yellow thought bubble contains the text "Ctrl +shift+I". Below these sections, there is a "Recent projects" table with two entries:

	Title	Description	Views	Forks	Updated
<input type="checkbox"/>	Static Starter	HTML/CSS/JS Starter	1	0	31 minutes ago
<input type="checkbox"/>	Static Starter	HTML/CSS/JS Starter	1	0	3 days ago

The Squid Game Set

Ctrl +shift+I



The screenshot shows the browser's developer tools with the "Elements" tab selected. The DOM tree is displayed, starting with the DOCTYPE declaration and the HTML root element. The body contains various components like a Redux store script, an app div, and several dialog elements for project management. A specific iframe is highlighted, which contains a URL related to a doubleclick.net advertisement.

```
<!DOCTYPE html>
<html lang="en" class="application theme-dark">
  <head> ...
  </head>
  <body> == $0
    <script type="application/json" data-redux-store> ... </script>
    <div id="app" data-page="{"component": "dashboard/show", "props": {}, "url": "/"}> ...
    </div>
    <script> ...
    </script>
    <iframe height="0" width="0" style="display: none; visibility: hidden;"> ...
    </iframe>
    <iframe allow="join-ad-interest-group" data-tagging-id="AW-11436847280" data-load-time="1736147240943" height="0" width="0" src="https://td.doubleclick.net/td/rul/11436847280?random=1736147240940&cv...amb=0&uam=&uap=Windows&uapv=15.0.0&uaw=0&fledge=1&data=event%3Dgtag.config" style="display: none; visibility: hidden;"> ...
    </iframe> ad
    <dialog id="create-new-project-dialog" aria-labelledby="create-new-project-dialog-title" class="_dialogReset_swca4_15 _modal_swca4_7 _modal_centered_swca4_26"> ...
    </dialog>
    <dialog id="confirm-project-deletion-modal" aria-labelledby="confirm-project-deletion-modal-title" class="_dialogReset_swca4_15 _modal_swca4_7 _modal_centered_swca4_26"> ...
    </dialog>
    <dialog id="add-to-collection-modal" aria-labelledby="add-to-collection-modal-title" class="_dialogReset_swca4_15 _modal_swca4_7 _root_1mour_9"> ...
    </dialog>
    <dialog id="new-collection-modal" aria-labelledby="new-collection-header" class="_dialogReset_swca4_15 _modal_swca4_7 _modal_centered_swca4_26"> ...
    </dialog>
    <dialog id="navigation-sidebar" aria-labelledby="navigation-sidebar-title" class="_dialogReset_swca4_15 _modal_swca4_7 _root_15jys_1 _root_mobile_15jys_11"> ...
    </dialog>
  </body>
</html>
```





The screenshot shows the StackBlitz web interface. At the top, there's a search bar with the placeholder "Search projects...", a "New Project" button, and a "D" icon. Below the search bar, there's a "New project" section with a "New project from one of StackBlitz starter templates" button. To the right of this, there are icons for Angular, Node.js, Docker, and other technologies. The main area is titled "Recent projects" with a "Show all >" link. It lists two recent projects:

Title	Views	Forks	Updated
Static Starter HTML/CSS/JS Starter	1	0	about 1 hour ago
Static Starter HTML/CSS/JS Starter	1	0	3 days ago

To the right of the main content, there's a browser developer tools sidebar. The "Elements" tab is selected. The sidebar shows the following CSS code:

```
element.style {  
}  
  
body {  
    font-size: 13px;  
    line-height: 1;  
    overscroll-behavior-y: none;  
}  
  
a, abbr, acronym, address, applet, article, aside, audio, big,  
blockquote, body, canvas, caption, center, cite, code, dd, del, details, dfn, div, dl, dt,  
embed, fieldset, figcaption, figure, footer, form, h1, h2, h3, h4, h5, h6, header, hgroup,  
html, iframe, img, ins, kbd, label, legend, li, mark, menu, nav, object, ol, output, p, pre, q,  
ruby, s, samp, section, small, span, strike, sub, summary, sup, table, tbody, td, tfoot, th,  
thead, time, tr, tt, u, ul, var, video {  
    border: 0;  
    font-size: 100%;  
    font: inherit;  
    margin: 0;  
    padding: 0;  
    text-overflow: ellipsis;  
    vertical-align: initial;  
}  
  
body {  
    line-height: inherit;  
    margin: 0;  
}  
  
*, :after, :before {  
    box-sizing: border-box;  
}  
  
*, ::backdrop, :after, :before {  
    --tw-border-spacing-x: 0;  
    --tw-border-spacing-y: 0;  
    --tw-translate-x: 0;  
    --tw-translate-y: 0;  
    --tw-rotate: 0;  
    --tw-skew-x: 0;  
    --tw-skew-y: 0;  
    --tw-scale-x: 1;  
    --tw-scale-y: 1;  
}
```

Static HTML/CSS

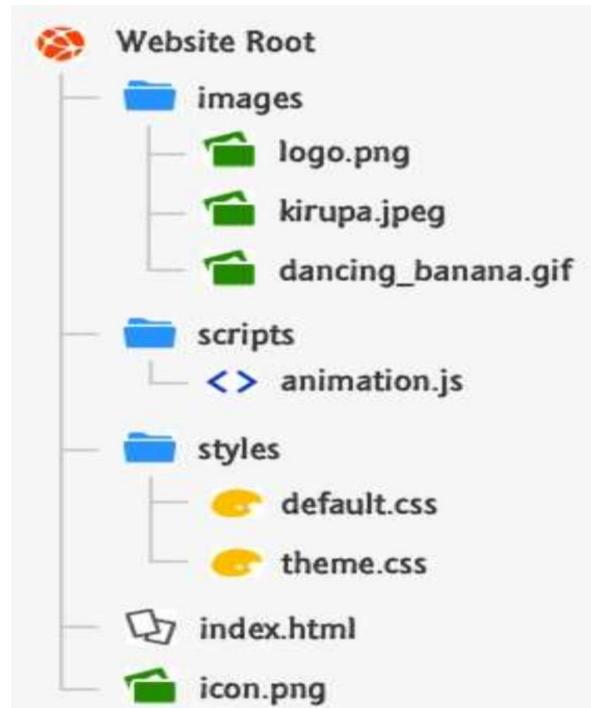
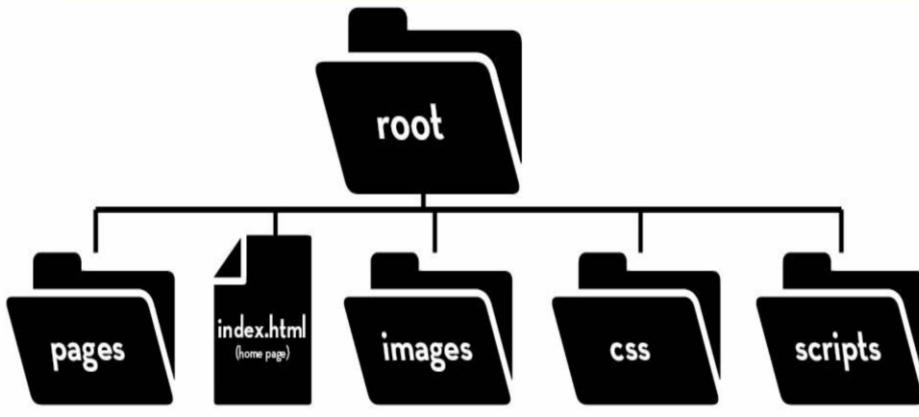
Add to  **diva.computing** ▾ X

Popular Frontend Backend Fullstack Docs, Blogs & Slides Creative Mobile & VR Vanilla Native Languages

 Astro Basics Node.js	 Next.js Node.js	 Nuxt Node.js	 React TypeScript
 Vanilla JavaScript	 Vanilla TypeScript	 Static HTML/JS/CSS	 Node.js Blank project
 Angular TypeScript	 Vue JavaScript	 WebContainer API Node.js	

<https://stackblitz.com/>

html-css-js project directory structure



<https://www.uok.edu.pk/>

The screenshot shows the Network tab of the Chrome DevTools. It lists various resources loaded by the page:

- top
- www.uok.edu.pk
 - alumni
 - css
 - images
 - js
- (index)
- platform.twitter.com
- static.licdn.com
- chromewebdata/

Two yellow thought bubbles with blue outlines are overlaid on the screenshot, pointing to the 'www.uok.edu.pk' entry and the '(index)' entry. The bubble for 'www.uok.edu.pk' contains the text 'Ctrl +shift+I'. The bubble for '(index)' contains the text 'Ctrl +shift+p'.



UNIVERSITY OF
KARACHI

The only file a website actually needs is one called: index.html

typing `WWW.PRACTICALSERIES.COM` into a web browser will cause it to look for and load a file called `index.html` at that address.

OTHER DEFAULT FILE NAMES

`index.html` is not the only default file name for a web site, the following all work:

<code>index.html</code>	<code>default.html</code>	<code>Index.cgi</code>
<code>index.htm</code>	<code>default.htm</code>	<code>Default.html</code>
<code>index.shtml</code>	<code>home.html</code>	<code>Default.htm</code>
<code>index.php</code>	<code>home.htm</code>	<code>Home.html</code>
<code>index.php5</code>	<code>Index.html</code>	<code>Home.htm</code>
<code>index.php4</code>	<code>Index.htm</code>	<code>placeholder.html</code>
<code>index.php3</code>	<code>Index.shtml</code>	
<code>index.cgi</code>	<code>Index.php</code>	

Note: the difference between lower and upper case in the starting letters

Web browsers look for files in the order listed above—starting top left with `index.html` and ending bottom right with `placeholder.html`.

Welcome to HTML5 Black and White, World!

← → ⌂ stackblitz.com/edit/stackblitz-starters-q269pttc?file=index.html

Ebook websites HT Redirection x Humer... DLI Event | Deep Le... YouTube Maps All Bookmarks

Save Fork Share Static Starter

b Open in bolt.new | A

PROJECT Create a repository

1 **index.html**

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>BSCS 633 Week 01 Website</title>
7 </head>
8
9 <body>
10  <h1>BSCS 4th Year 2025!</h1>
11  <p>Welcome to HTML5 Black and White, World!</p>
12 </body>
13 </html>
```

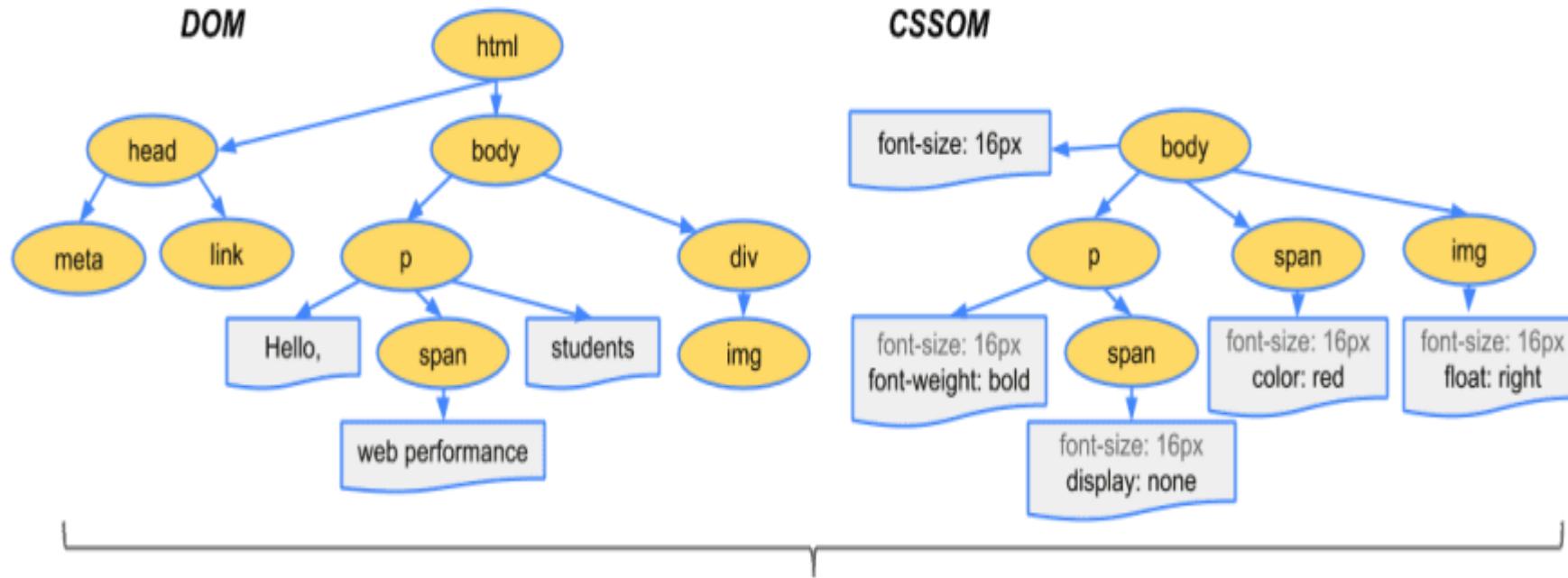
BSCS 4th Year 2025!

Welcome to HTML5 Black and White, World!

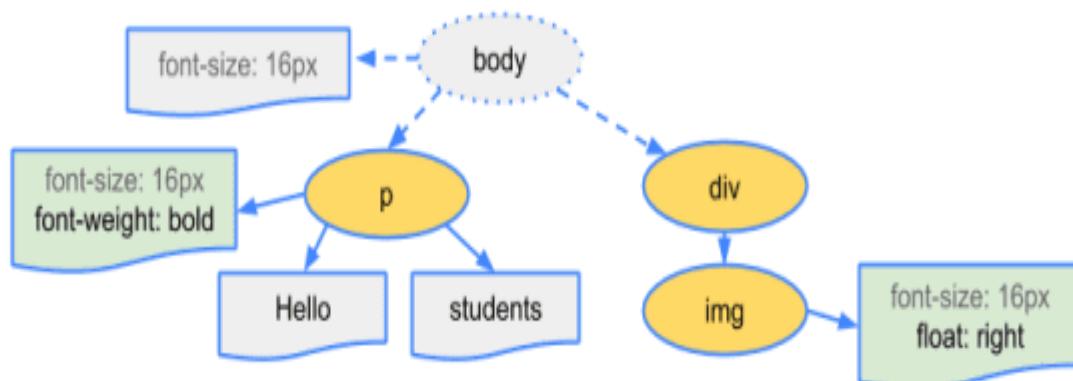


Every Browser knows and process HTML

Parent-child
relationship ??



Render Tree



UNIVERSITY OF
KARACHI

Tags vs Elements



IAD-cheat-sheet_01.pdf

- An HTML Element is everything from the start tag to the end tag

e.g. `<p>Some text</p>` - is referred to as an element, including starting tag- content- ending tag

`<p>` and `</p>` - are referred to as tags.

Tags Used in the `<head>` Section

Tag	Description
<code><title> text </title></code>	title shown on page tab
<code><meta attribute="value" ... /></code>	page metadata
<code><link href="url" type="text/css" rel="stylesheet" /></code>	links to a CSS style sheet
<code><script src="url"></script></code>	link to JavaScript code
<code><!-- comments --></code>	comment (can appear in head or body)



UNIVERSITY OF
KARACHI

Code break down

Element	Purpose
<code><!DOCTYPE html>:</code>	This tells the browser that we're using HTML5
<code><html lang="en">:</code>	This is the root element of the HTML page. The lang="en" attribute specifies that the content is in English.
<code><head>:</code>	This contains meta information about the document.
<code><meta charset="UTF-8">:</code>	This specifies the character encoding for the document (UTF-8 is the standard for most websites).
<code><meta name="viewport" content="width=device-width, initial-scale=1.0">:</code>	This ensures that the website looks good on all devices, including mobile phones.
<code><title>:</code>	This sets the title of the webpage, which appears in the browser tab.
<code><body>:</code>	This is where all the visible content of your webpage goes.
<code><h1>:</code>	This creates a large heading.
<code><p>:</code>	This creates a paragraph of text.

4- Ways to set the background color

BSCS 633 Year 2025!

Welcome to HTML5 colored, World!

Coloring the background of div Element using color name

This is a paragraph

Coloring the background of div Element Using RGB values

RGB values `rgb(0,255,0)`

Coloring the background of div Element Using Hex color

Hex Color `#FF0000`

Coloring the background of div Element Using RGB with Alpha

`rgba(0,0,255,0.5)`

```
<body style="background-color:aqua"> set HTML background color in the <body> element
```

```
  <h1>BSCS 633 Year 2025!</h1>
```

```
  <p>Welcome to HTML5 colored, World!</p>
```

```
<div style="background-color:red">
```

```
  <h2>Coloring the background of div Element using color name</h2>
```

```
  <p>This is a paragraph</p>
```

```
</div>
```

```
<div style="background-color:rgb(0,255,0)">
```

```
  <h2>Coloring the background of div Element Using RGB values</h2>
```

```
  <p>RGB values rgb(0,255,0)</p>
```

```
</div>
```

```
<div style="background-color:#FF0000">
```

```
  <h2>Coloring the background of div Element Using Hex color</h2>
```

```
  <p>Hex Color #FF0000</p>
```

```
</div>
```

```
<div style="background-color:rgba(0,0,255,0.5)">
```

```
  <h2>Coloring the background of div Element Using RGB with Alpha</h2>
```

```
  <p>rgba(0,0,255,0.5)</p>
```

```
</div>
```

```
</body>
```

Save Fork Share

Static Starter

index.html • index_colors.txt

P ⌂ ... ⌂ stackblitzstartersq269pttc-ltw2--8080--c8c182a3.local-cred...

15 <body style="background-color:green" >
16
17 <h1>Welcome to BSCS 633 Website!</h1>
18
19 <h2>Lets Learn About UBIT Architecture !</h2>
20
21
22
23 <p> The Department of Computer Science, University of Karachi, was established by a resolution of Academic Council in its meeting, held on November 27. 1984, and it began functioning in the academic year 1985-86 by offering a Degree Program in Master of Computer Science (MCS) and became one of first institutions in Karachi imparting education in Computer Science and Technology.
The department has started in the year 1996
24

Prompt run, edit, and deploy full-stack web apps.

Welcome to BSCS 633 Website!

Lets Learn About UBIT Architecture !



Tags Used in the <body> Section

Tag	Display	Description
<p> text </p>	Block	paragraph
<h1> text </h1>	Block	(h1 for largest to h6 for smallest)
<h2> text </h2>		
...		
<h6> text </h6>		
	Inline-Block	image

Here are some **highlights** of UBIT Architecture:

- Inspired by Roman Pantheon
- A magnificent temple to all of the gods, the Roman Pantheon has stood on its site on the *Campus Martius* for almost *2,000 years*
- The inscription declares that it was built by **Marcus Agrippa** who built the original Pantheon in **27 BC**

```
<p>Here are some <strong>highlights</strong> of UBIT Architecture:</p>
<ul>

    <li> Inspired by Roman Pantheon </li>
    <li>A magnificent temple to all of the gods, the Roman Pantheon has stood on its site on t
he <em>Campus Martius</em> for almost <em>2,000 years</em></li>
    <li>The inscription declares that it was built by <strong>Marcus Agrippa</strong> who bui
lt the original Pantheon in <strong>27 BC</strong></li>
</ul>
```

 text 	Inline	emphasis (italic)
 text 	Inline	strong emphasis (bold)
 text text nested item nested item 	Block	ordered (ol) and unordered (ul) list; list item (li)

To turn the element to a link? The closest equivalent would be to use an

Want to learn more about UBIT? Visit the official dcs/ubit website.

Tags Used in the <body> Section

Tag	Display	Description
<p> text </p>	Block	paragraph
 text 	Block	anchor (link)
	Inline-Block	image

```
<p>Want to learn more  
about UBIT? <a  
href="https://www.uok.edu.  
pk/faculties/  
computerscience/ubit.  
php">Visit the official  
dcs/ubit website</a>. </p>
```



www.uok.edu.pk refused to connect.

```
<p>Want to learn more  
about UBIT? <a  
href="https://humera.pk/  
">Visit the official dcs/  
ubit website</a>. </p>
```



humera.pk refused to connect.

Html5 Link to Open New Tab Example

```
<!DOCTYPE HTML>
<html>
<head> <meta charset="utf-8">
<title>Creating Links</title>
</head>
```

```
<body>
<h1>How to Create Hyperlinks in HTML 5</h1>
<p><a href=
https://www.uok.edu.pk/faculties/computerscience/
ubit.php
</a> is a static website using php.
</p>
</body>
</html>
```

1999.

Here are some **highlights** of UBIT Architecture:

- Inspired by Roman Pantheon
- A magnificent temple to all of the gods, the Roman Pantheon on *Campus Martius* for almost 2,000 years
- The inscription declares that it was built by **Marcus Agrippa** in the Pantheon in **27 BC**

Want to learn more about UBIT? [Visit the official dcs/ubit website](#)

Open link in new tab

Open link in new window

Open link in incognito window

 Open link as humera

Save link as...

Copy link address

Get image descriptions from Google >

Inspect

Team Assignment 01

Internet Application Development

Copyright © 2025, Humera Tariq

Department of Computer Science (DCS/UBIT)
University of Karachi
January 2025

Online Thrift Store 633

Using HTML 5 only, create a prototype of an online Thrift store by ubit students.

Validate and verify the result on the browser as you proceed. Don't mind about the visual aspect of the result (that will be taken care by using CSS).

- Use the site <http://www.lipsum.com/> if you need to generate example paragraphs.
- Create separate **sections** for the header, menu and body of the Thrift Store. Use semantic tags (**section**, **header**, **nav**, ...) for those sections...
- The header should have a title (**h1**), a subtitle (**h2**) and a logo (**img**).
- The menu should have links to each of the online thrift store sections (Books, Electronics,) organized using an unordered list (**ul**). These links can simply point to example URLs.
- The thrift store body should have a section for each second hand/used item. Use the semantic tag **article**.
- Old items should have a title (**h3**), an introduction (**p**) and an image (**img**). It should also have a footer containing the owner name and age of used item.
- Don't forget to [validate](#) the page when you're done.

Semantic Elements

Semantic element clearly describes its meaning to the browser and the developer.

- Examples: <table>, <form>, <header> and <footer>

- Non-semantic element tells nothing about its content. - Examples: <div> and

<div> ... </div>	Block	block-level section of a page
 ... 	Inline	inline-level section of a page

HTML5 Semantic Grouping Tags

Tag	Display	Description
<header>	Block	Container for a header of a document
<footer>	Block	Container for a footer of a document
<article>	Block	A standalone piece of content (e.g., entire blog post including title, author, etc.)
<section>	Block	A piece of content that is part of another (e.g., a chapter section of a reading)
<aside>	Block	Defines some content aside from the content it is placed in (e.g., a sidebar in an article)
<main>	Block	Specifies the main content of a document. The content inside should be unique to the document and not contain content that is repeated across pages (e.g., sidebars, nav links, search bars, etc.)

Why Semantic Elements in HTML5?

- ✓ In <div>, we give an id or class which tells about what kind of content it is holding; either body, header or footer etc.
- ✓ In case of semantic elements of HTML5, tag name itself clearly defines what kind of code it's holding, and it is for which part of the website.
- ✓ Allows data to be shared and reused across applications, enterprises and communities.
- ✓ <header> <footer> <nav> <section> <article> looks more organized and easier.

Basic wireframe using HTML5 Semantic Elements

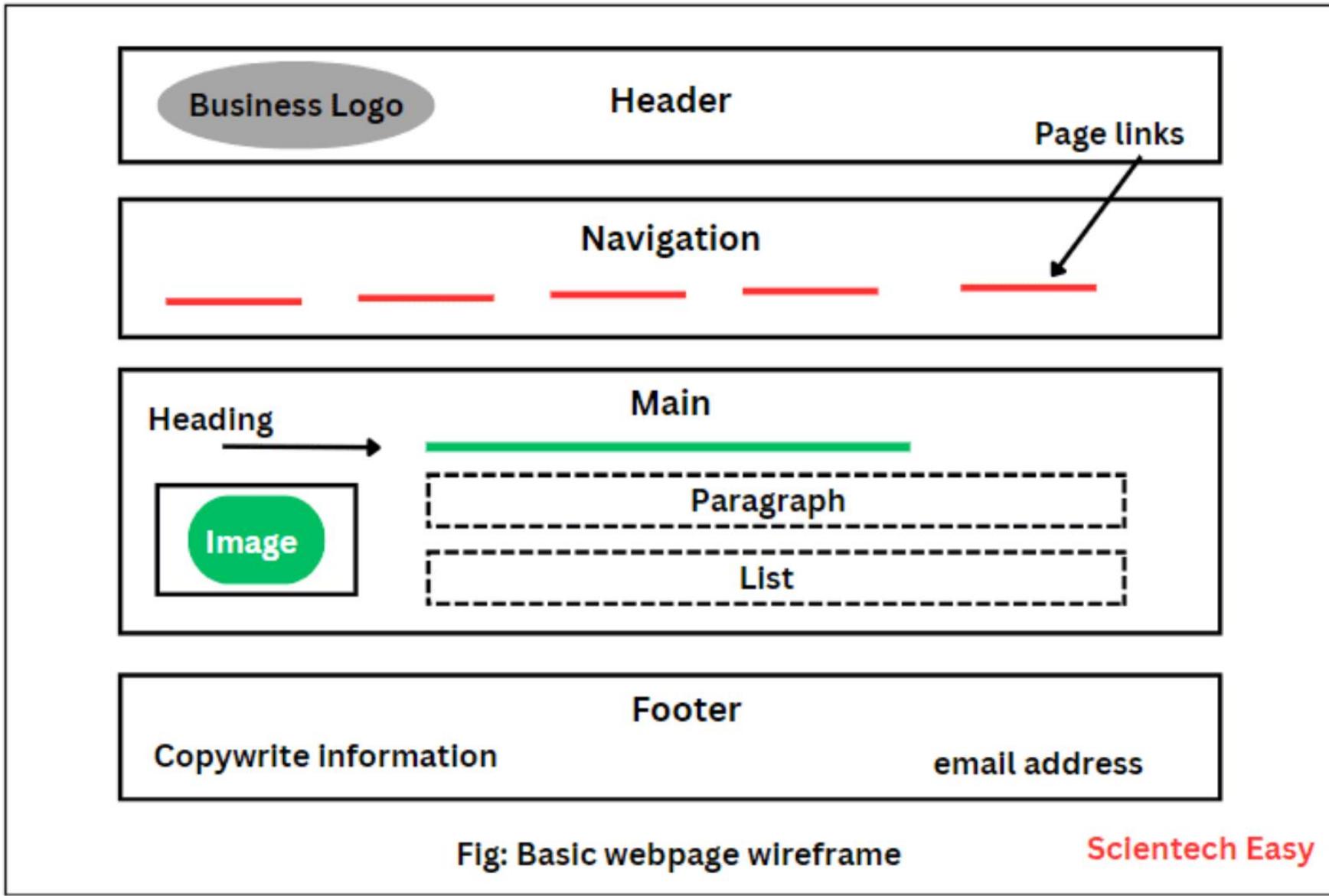


Fig: Basic webpage wireframe

Scientechn Easy

Semantic Elements Example

index.html ●

P ⏪ ⏴ ... ⏵ stackblitzstartersq269pttc-ltw2--80

```
3   <header>
4     <h2>My Website</h2>
5   </header>
6   <nav>
7     <ul>
8       <li><a href="index.html">Home</a></li>
9       <li><a href="about.html">About</a></li>
10      <li><a href="about.html">Services</a></li>
11      <li><a href="#">Contact</a></li>
12    </ul>
13  </nav>
14
15  <article>
16    <p> Lets sale something to ubitians here </p>
17  </article>
18
19
20  <footer>
21    <p>Copyright BSCS 633 2025</p>
22  </footer>
23
24 </section>
25
```

My Website

- [Home](#)
- [About](#)
- [Services](#)
- [Contact](#)

Lets sale something to ubitians here

Copyright BSCS 633 2025

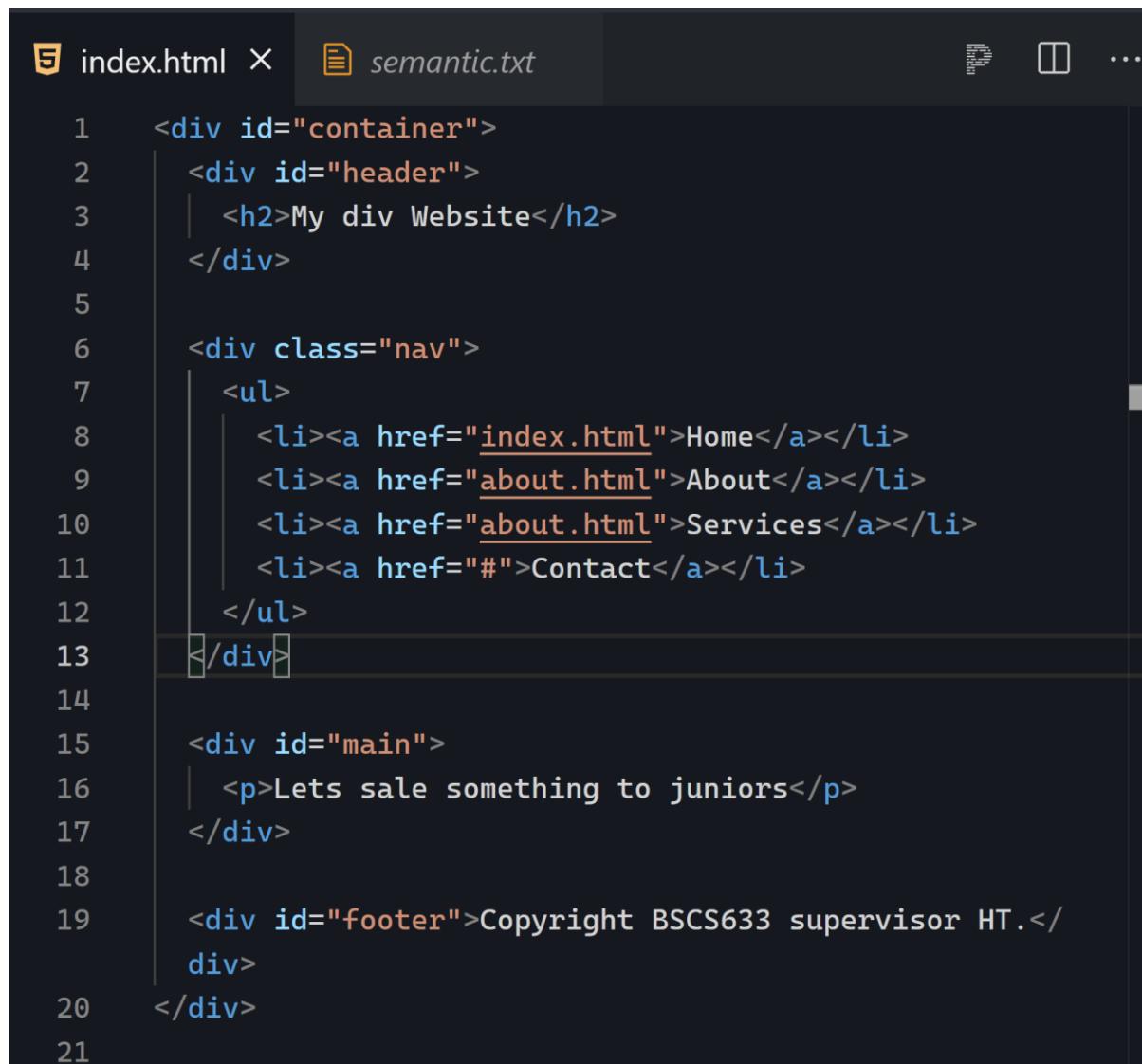
Syntax:

```
<a href="#section1" >section 1</a>
```

Approach

- **Use the Anchor Tag `<a>`:** In HTML, use the `<a>` tag to create links within the same page.
- **Assign Unique IDs:** Assign unique IDs to different sections of the webpage using the `id` attribute.
- **Set the `href` Attribute:** Set the `href` attribute of the anchor tag to `#section1` (replace “`section1`” with the desired ID) to link to a specific section.
- **Avoid Using Class Names in `href`:** Class names are not unique identifiers and should not be used in the `href` attribute for internal linking.

Dividing the webpage into sections using <div>



```
index.html X semantic.txt
1  <div id="container">
2    <div id="header">
3      <h2>My div Website</h2>
4    </div>
5
6    <div class="nav">
7      <ul>
8        <li><a href="index.html">Home</a></li>
9        <li><a href="about.html">About</a></li>
10       <li><a href="about.html">Services</a></li>
11       <li><a href="#">Contact</a></li>
12     </ul>
13   </div>
14
15   <div id="main">
16     <p>Lets sale something to juniors</p>
17   </div>
18
19   <div id="footer">Copyright BSCS633 supervisor HT.</
20     div>
21   </div>
```

My div Website

- [Home](#)
- [About](#)
- [Services](#)
- [Contact](#)

Lets sale something to juniors

Copyright BSCS633 supervisor HT.



UNIVERSITY OF
KARACHI

Quick Overview

VS code

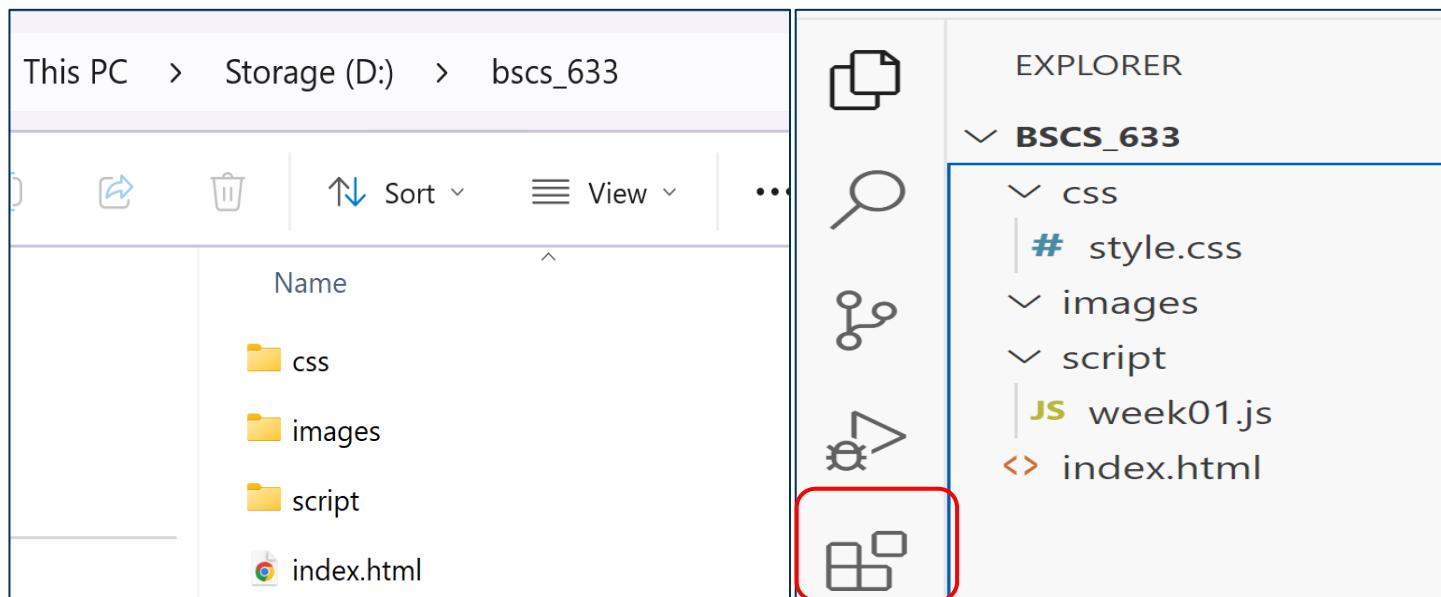
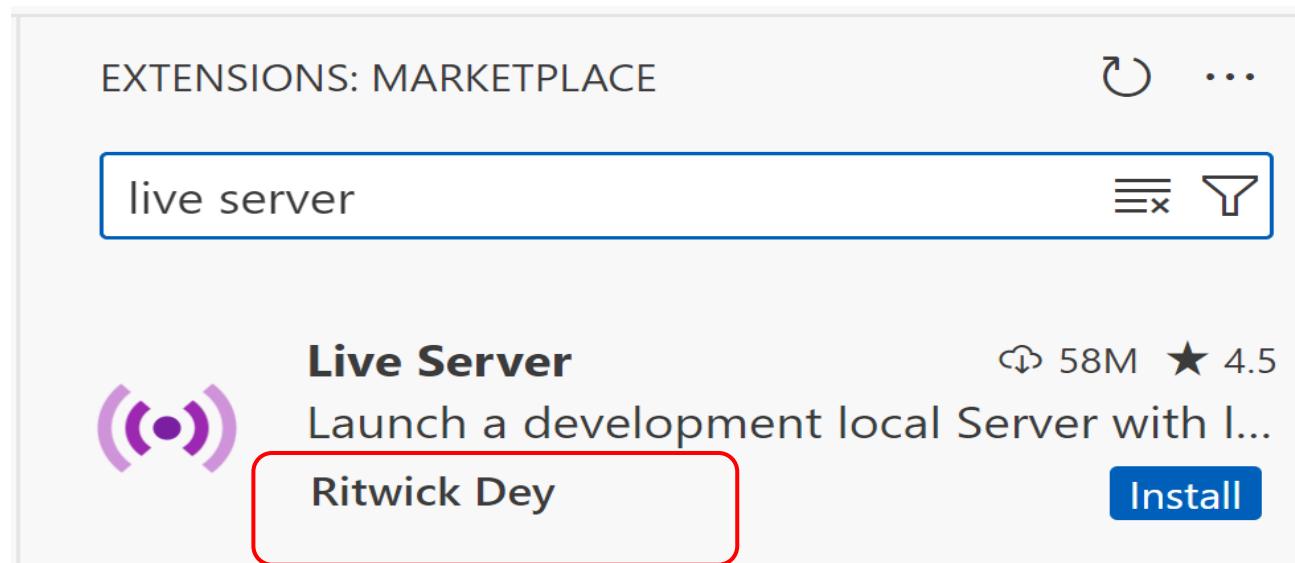
DOM

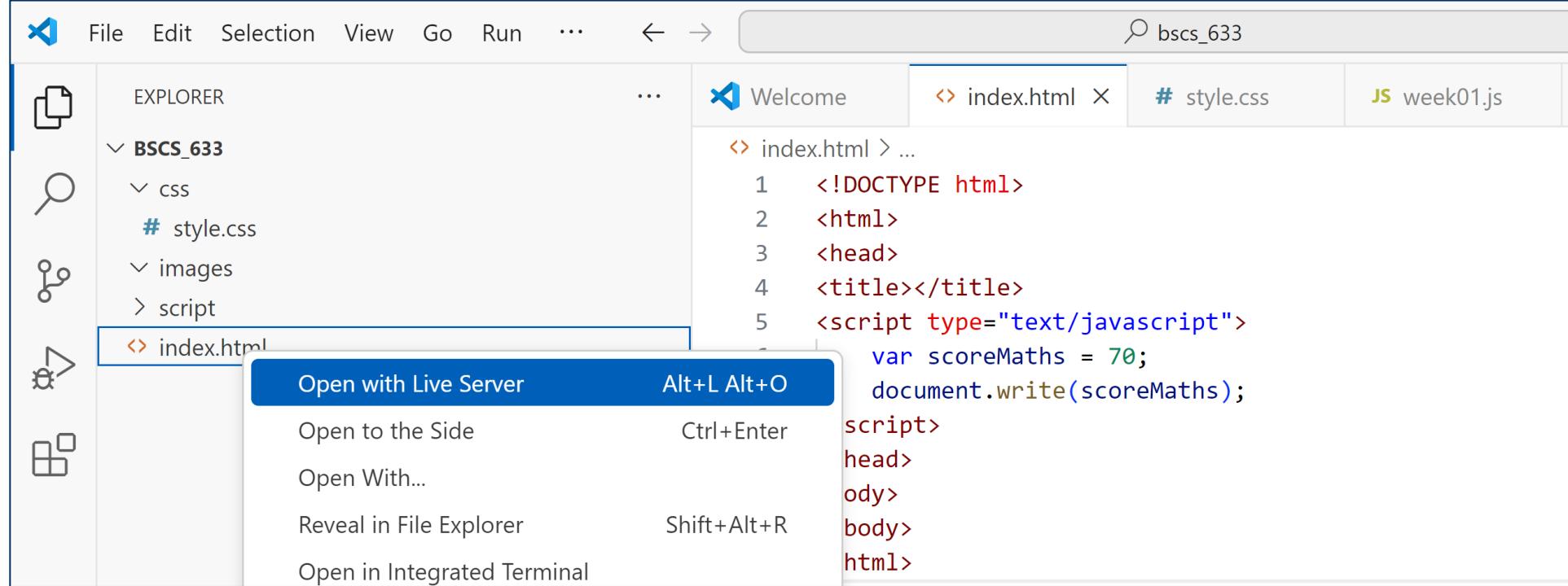
Chrome developer tools

Copyright © 2025, Humera Tariq

Department of Computer Science (DCS/UBIT)
University of Karachi
January 2025

Open the Extension section and search for your extension in the search bar.





File Edit Selection View Go Run ... ← → bscs_633

EXPLORER

BSCS_633

- css
- # style.css
- images
- script

index.html

Open with Live Server Alt+L Alt+O

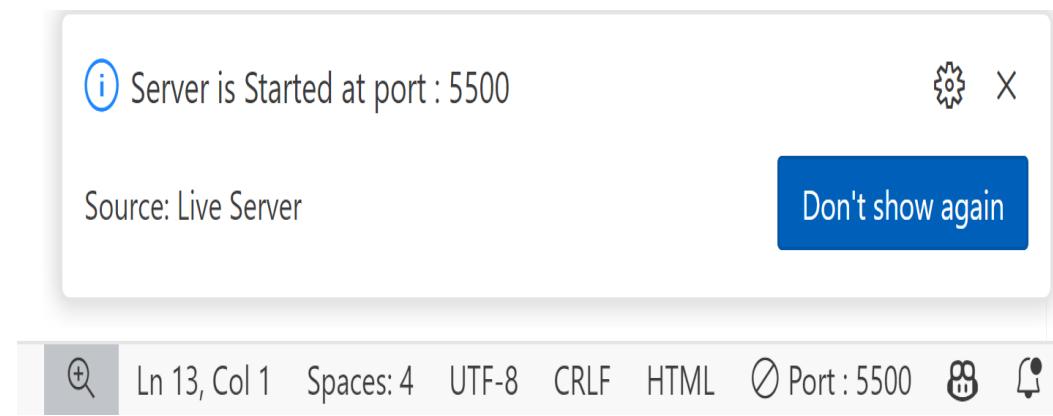
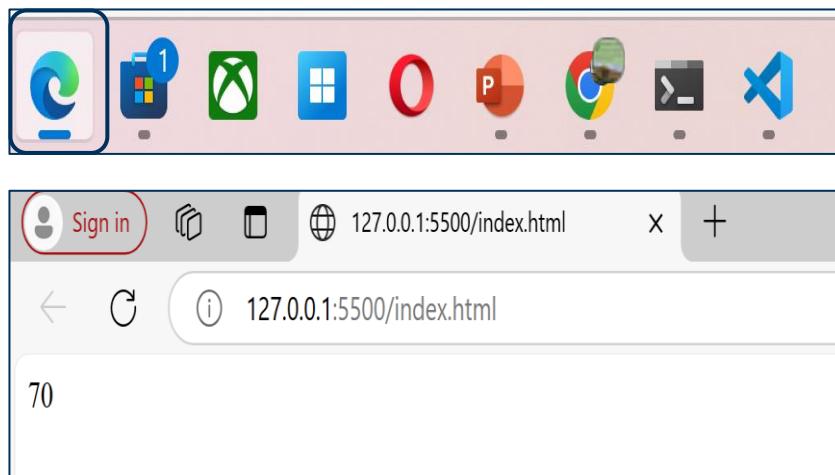
Open to the Side Ctrl+Enter

Open With...

Reveal in File Explorer Shift+Alt+R

Open in Integrated Terminal

```
<!DOCTYPE html>
<html>
<head>
<title></title>
<script type="text/javascript">
var scoreMaths = 70;
document.write(scoreMaths);
</script>
</head>
<body>
</body>
</html>
```



Sign in

BSCS 633 Humera Tariq, World!

127.0.0.1:5500/index.html

Hello
World!

The internet is a series of tubes.

index.html X style.css week01.js

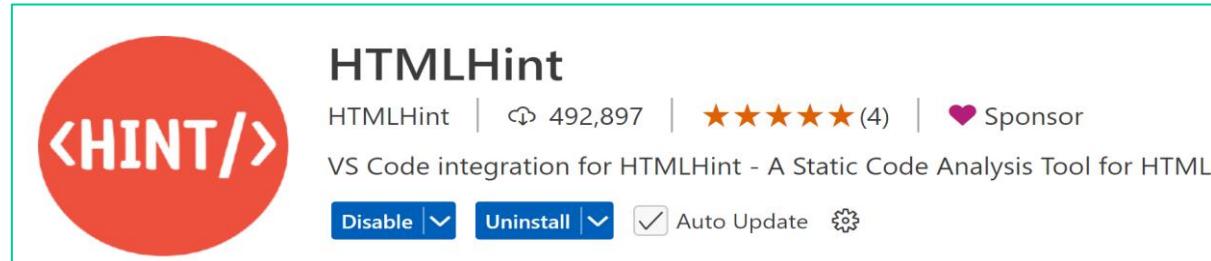
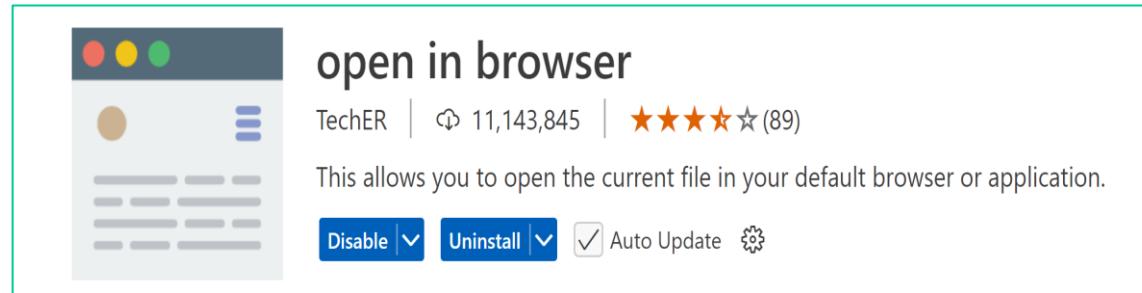
```
1 <html lang="en">
2 <head>
3   <title> BSCS 633 Humera Tariq, World! </title>
4 </head>
5 <body>
6 <div>
7   <span>Hello <div>World!</div></span>
8   <src="hello.jpg" alt="image of hello" img>
9 </div>
10 <p>
11   BSCS 633 is the best class ever! #propagandamachine
12 </p>
13 <blockquote>
14   The internet is a series of tubes.
15 </body>
16 </html>
```



vtime.com ID:34210352 © Stuart Miles

Visual Studio Code
doesn't have HTML
validation by default.
But it allows you to
add extensions and
enable these
features.

To add **HTML validation (linting)**, open Visual Studio Code, and then press **Ctrl + P**. Then paste ext install **HTMLHint** in it, and press Enter. It will install an HTML validator. You may need to reload Visual Studio Code to load the extension.



stackblitz.com/edit/stackblitz-starters-q269pttc?file=index.html

```
5 index.html X P □

1 <html lang="en">
2 <head>
3 | <title> BSCS 633 Humera Tariq, World! </title>
4 </head>
5 <body>
6 <div>
7 | <span>Hello <div>World!</div></span>
8 | <src="hello.jpg" alt="image of hello" img>
9 </div>
10 <p
11 | BSCS 633 is the best class ever! #propagandamachine
12 </p>
13 <blockquote>
14 | The internet is a series of tubes.
15 </body>
16 </html>
```

Terminal

```
█ Serving: /home/projects/stackblitz-starters-q269pttc  
█ Local: http://localhost:8080  
█ Network: http://192.168.1.104:8080
```



The internet is a series of tubes.



index.html 6

- ⚠ Doctype must be declared first. (doctype-first) [Ln 1, Col 1]
- ⚠ Special characters must be escaped : [<]. (spec-char-escape) [Ln 8, Col 2]
- ⚠ Special characters must be escaped : [>]. (spec-char-escape) [Ln 8, Col 43]
- ⚠ Special characters must be escaped : [<]. (spec-char-escape) [Ln 10, Col 1]
- ⚠ Tag must be paired, no start tag: [</p>] (tag-pair) [Ln 12, Col 1]
- ⚠ Tag must be paired, missing: [</blockquote>], start tag match failed [<blockquote>] on line 13. (tag-pair) [Ln 15, Col 1]

index.html 6 X

Extension: open in browser

style.css

JS week01.js

<> index.html > html

```
1  <html lang="en">
2  <head>
3  | <title> BSCS 633 Humera Tariq, World! </title>
4  </head>
5  <body>
6  <div>
7  | <span>Hello <div>World!</div></span>
8  | <src="hello.jpg" alt="image of hello" img>
9  </div>
10 <p
11 | BSCS 633 is the best class ever! #propagandamachine
12 </p>
13 <blockquote>
14 | The internet is a series of tubes.
15 </body>
16 </html>
```

First look at HTML, DOM and JS!

```
<!doctype html>
<html>
  <head>
    <title>Hello, world!</title>
  </head>
  <body>
    <h1>Hello, world!</h1>
    <p>This is a hypertext document on the World Wide Web.</p>
    <script src="/script.js" async></script>
  </body>
</html>
```

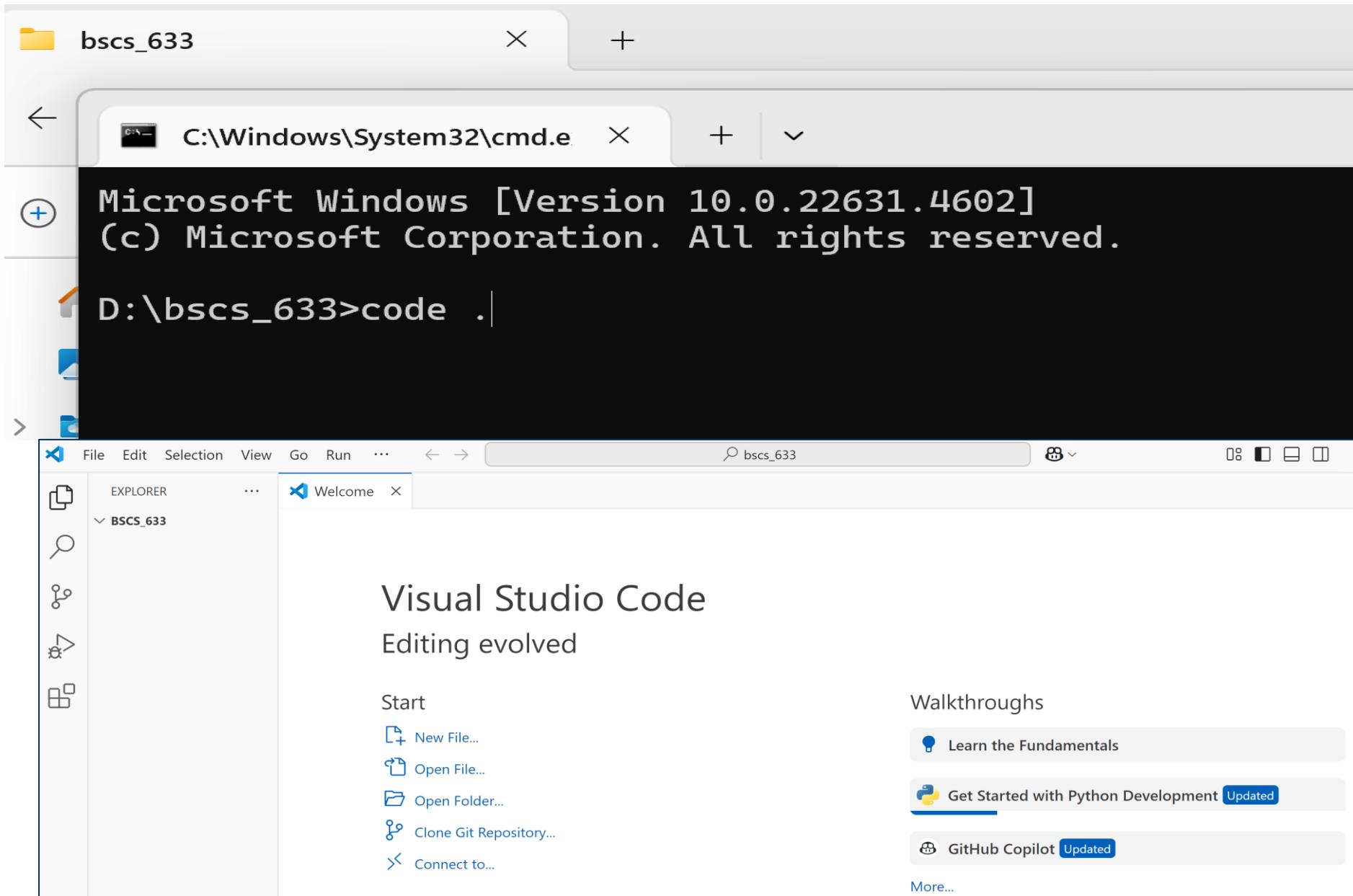
```
html
  head
    title
  body
    h1
    p
    script
```

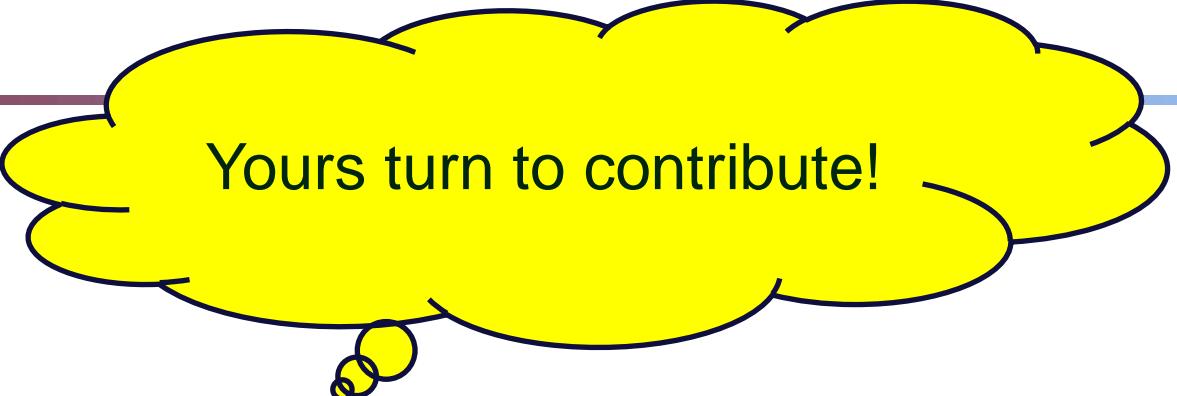
```
const h1 = document.querySelector('h1');
h1.parentElement.removeChild(h1);
const p = document.createElement('p');
p.textContent = 'Wildcard!';
document.body.appendChild(p);
```



UNIVERSITY OF
KARACHI

Create project folder and open in VSCode





Yours turn to contribute!

Class Activity 1: Arrange and Translate the given workflow into a schematic/block diagram showing the sequence with appropriate numbering

- ✓ The script referenced at the bottom of the HTML runs controller code
- ✓ The browser parses the HTML and creates a tree of objects
- ✓ The JS code removes the h1 node and adds another p node to the DOM.
- ✓ When you use a web browser to request a page like <https://example.com> the server returns HTML
- ✓ The page's HTML is now different than its DOM.



Class Activity 2

Yours turn to contribute!

a) Write and draw a new DOM as Tree!

```
html  
  head  
    title  
  body  
    h1  
    p  
  script
```

```
const h1 = document.querySelector('h1');  
h1.parentElement.removeChild(h1);  
const p = document.createElement('p');  
p.textContent = 'Wildcard!';  
document.body.appendChild(p);
```

b) Choose the best word/rewrite given line using correct words:

(HTML,DOM) represents initial page content/state, and the

(HTML,DOM) represents current page content. When (HTML,

JavaScript,DOM) adds, removes, or edits nodes, the (HTML,DOM,JS)

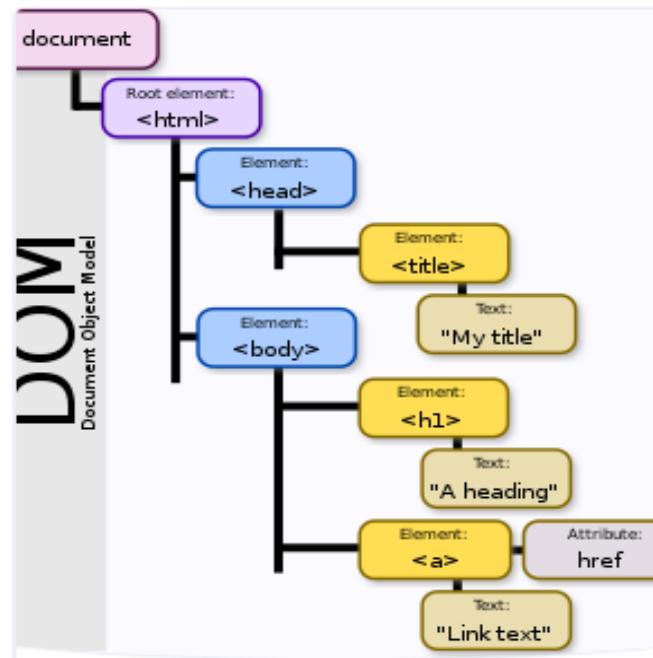
becomes different than the (HTML, DOM,JS).



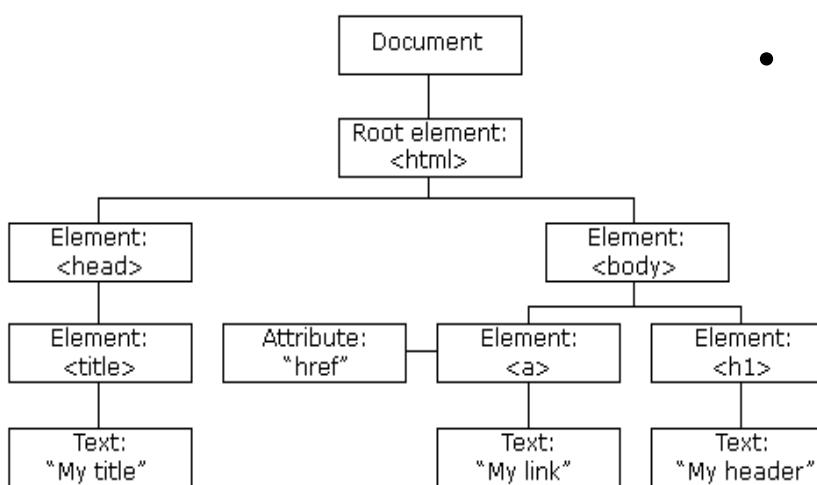
UNIVERSITY OF
KARACHI

What the heck is the DOM?

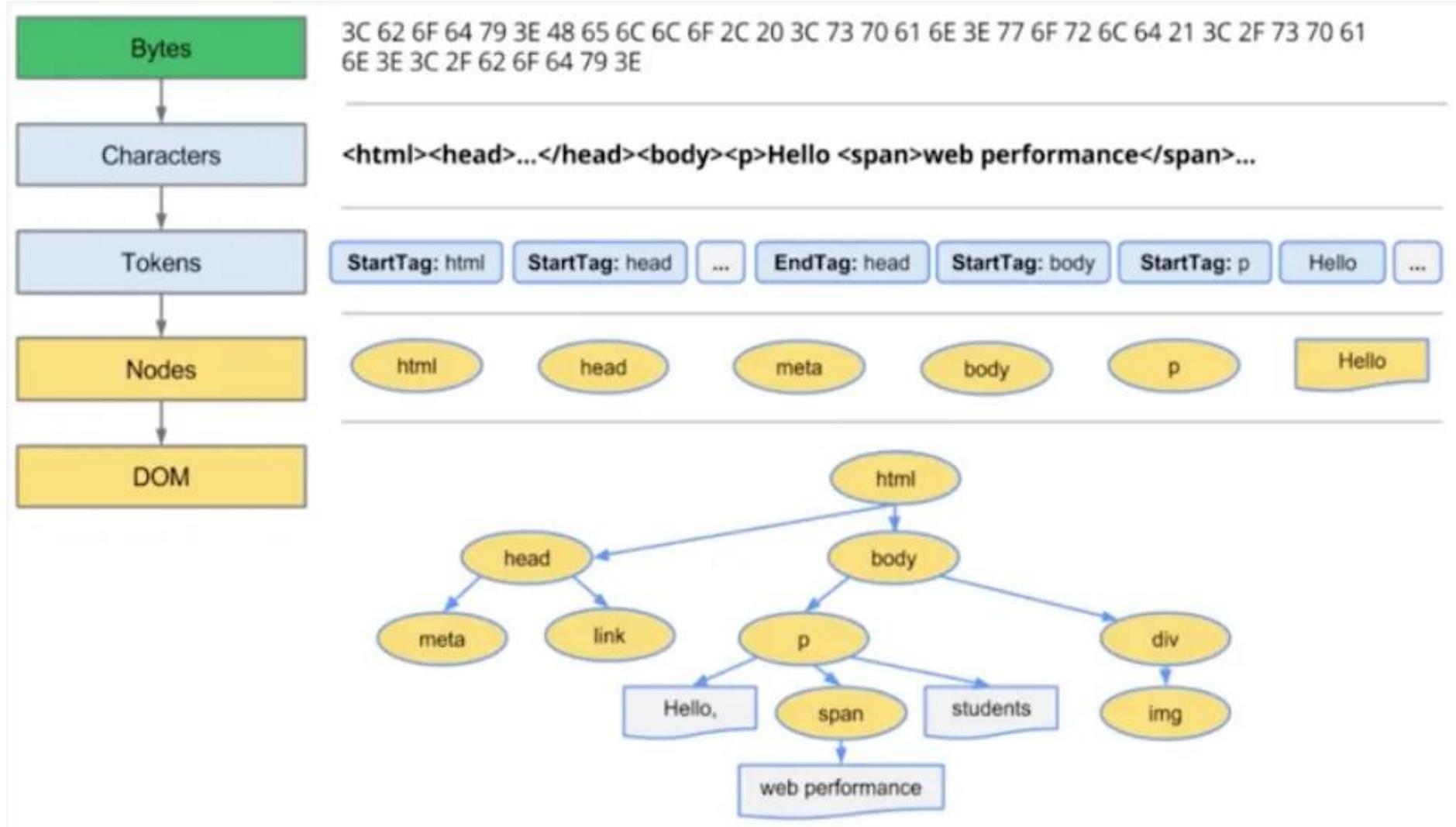
- Your web browser builds a *model* of the web page (the *document*) that includes all the *objects* in the page (tags, text, etc)
- All of the properties, methods, and events available to the web developer for manipulating and creating web pages are organized into objects
- Those objects are accessible via scripting languages in modern web browsers



- 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



HMTL is a Page Description Language, like PDF, or post script



Take home

Internet Application Development

Copyright © 2025, Humera Tariq

Department of Computer Science (DCS/UBIT)
University of Karachi
January 2025

Practice, Practice , Practice.....

<https://developer.chrome.com/docs/devtools>

<https://www.youtube.com/watch?v=wcFnnxfA70g>

<https://stackblitz.com/register>

“StackBlitz is the first online IDE whose compute model makes sense to me.”



Tom Preston-Werner

founder of GitHub & investor in StackBlitz



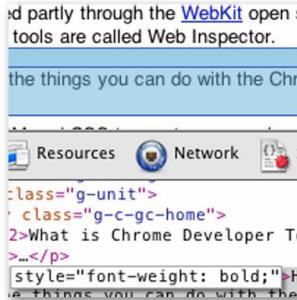
Chrome Developer Tools Cheat Sheet

Open with **Ctrl - Alt - I** or **right click** and **inspect element**. View keyboard shortcuts with **?**

HTML

Edit DOM attributes inline

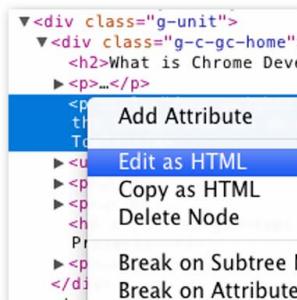
In *Elements* tab or styles drawer, **double click** attribute.



Edit chunks of HTML

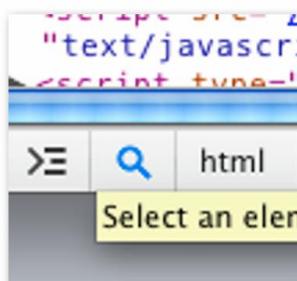
In *Elements* tab, **right click** an element and select "Edit as HTML"

- Copy HTML via menu item
- Delete node via menu item



Visually inspect DOM elements

In *Elements* tab, **click** footer button or **Ctrl - Shift - C** and select desired element in page.

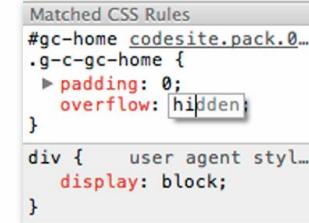


CSS

Inline CSS per-element styling

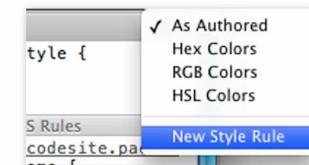
In *Elements* sidebar, **double click** CSS

- Autocompletion of keys and values
- **Tab** between keys and values
- **Double click** after closing bracket to add new property



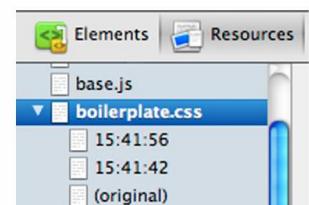
Create new CSS rules

Click on gear icon in *Elements* sidebar



Free form CSS editing

In *Resources* sidebar, **double click** CSS and **Ctrl - S**



Versioned CSS/JS

CSS/JS changes get saved into versions

- **Right click** and revert to a revision.
- **Right click** and save back to a local file.



Chrome Developer Tools Cheat Sheet

Open with **Ctrl - Alt - I** or **right click** and **inspect element**. View keyboard shortcuts with **?**

JavaScript

Console

Invoke with **Esc** or **Ctrl - Alt - J** or *Console*

- Get current DOM element with `$0`

```
> var foo = 'bar';
```

Execution controls: Stop, Run, Break, All.

Console API

- `copy(text)` – copy text to clipboard
- `dir(elt)` – show element properties
- `inspect(elt)` – open in *Elements* tab

```
> var foo = 'bar';
```

Context menu options: XMLHttpRequest logging, Clear Console, Inspect Element.

Edit chunks of JavaScript

In *Scripts* tab, **double click** script to edit. **Ctrl - S** to save

- Go to line with **Ctrl - L**

Script pretty printing

```
true},data:function(a,
```

Pretty print button.

Breakpoints

In *Scripts* tab, **click** in the gutter

- Conditional breakpoints **right click** the gutter
- DOM breakpoints in *Elements* tab, right click
- XHR breakpoints in *Scripts* sidebar **+** to filter by URL
- Event listener breakpoints In *Scripts* sidebar, check off type
- Exception breakpoints In *Scripts* tab, break icon in footer

```
if(!url)
    return url;
// If URL is al
if(/^([a-zA-Z]+\://
    return url;
// Leading / me
if(url[0] == '/'
    return docu
```

Breakpoint icon on line 8.

Breakpoints sidebar:

- ▶ Breakpoints
- ▶ DOM Breakpoints
- ▼ XHR Breakpoints +
- Any XHR
- ▼ Event Listener Breakpoints
- ▶ Keyboard
- ▶ Mouse

```
function theFunction
    listen_for_broadcast_dispatcher[functio
```

Function definition in the Scripts tab.

```
9
10
11
12
13
14
15
16
```

Breakpoint icon on line 9.

Variable inspection

When at a breakpoint, **hover** over variable

by @borismus and @paul_irish



UNIVERSITY OF
KARACHI



Chrome Developer Tools Cheat Sheet

Open with **Ctrl - Alt - I** or right click and **inspect element**. View keyboard shortcuts with **?**

A hands-on walk-through tweaking a web application using the developer tools is Fun!

<https://developer.chrome.com/docs/devtools>

<https://developer.chrome.com/docs/devtools/overview>

- [View and change the DOM](#)
- [View and change CSS](#)
- [Debug JavaScript](#)
- [View messages and run JavaScript in the Console](#)
- [Optimize website speed](#)
- [Inspect network activity](#)



UNIVERSITY OF
KARACHI

HMTL is a Page Description Language, like PDF, or post script

HTML

HTML means Hyper Text Markup Language.

The HTML allow us to define the structure of a document or a website.

HTML is **NOT** a programming language, it's a markup language, which means its purpose is to give structure to the content of the website, not to define an algorithm.

It is a series of nested tags (it is a subset of [XML](#)) that contain all the website information (like texts, images and videos). Here is an example of tags:

```
<title>This is a title</title>
```

The HTML defines the page structure. A website can have several HTMLs to different pages.

```
<html>
  <head>
  </head>
  <body>
    <div>
      <p>Hi</p>
    </div>
  </body>
</html>
```

HTML: some basic rules

- It uses XML syntax (tags with attributes, can contain other tags).
`<tag_name attribute="value"> content </tag_name>`
- It stores all the information that must be shown to the user.
- There are different HTML elements for different types of information and behaviour.
- The information is stored in a tree-like structure (nodes that contain nodes inside) called DOM (Document Object Model).
- It gives the document some semantic structure (pe. this is a title, this is a section, this is a form) which is helpful for computers to understand websites content.
- It must not contain information related to how it should be displayed (that information belongs to the CSS), so no color information, font size, position, etc.

Building the HTML Page

An HTML page is a collection of nodes represented by text elements contained in angle brackets.

HTML is a page description language.

A browser connects to an HTTP server through an Internet connection and specifies a text file containing HTML code that should be returned down the TCP pipe to the requesting browser.

The browser contains an HTML rendering engine which knows how to parse the HTML document object model and render the described content onto the screen of the device.

The original HTML specification prescribed only text and images. [The current HTML specification-HTML 5](#)-allows for the use of plug-ins which can accommodate the rendering of additional rich media types such as sound and video.





UNIVERSITY OF
KARACHI



"Don't be satisfied with stories, how things have gone with others. Unfold your own myth." ~Rumi



UNIVERSITY OF
KARACHI



Department of Compute Science (UBIT Building), Karachi, Pakistan.

1200 Acres (5.2 Km sq.)

53 Departments

19 Institutes

25000 Students

My Homeland Pakistan

