



# INTRO TO WEB DEVELOPMENT

SJU ACM STUDENT CHAPTER



# SJU ACM

Student Chapter

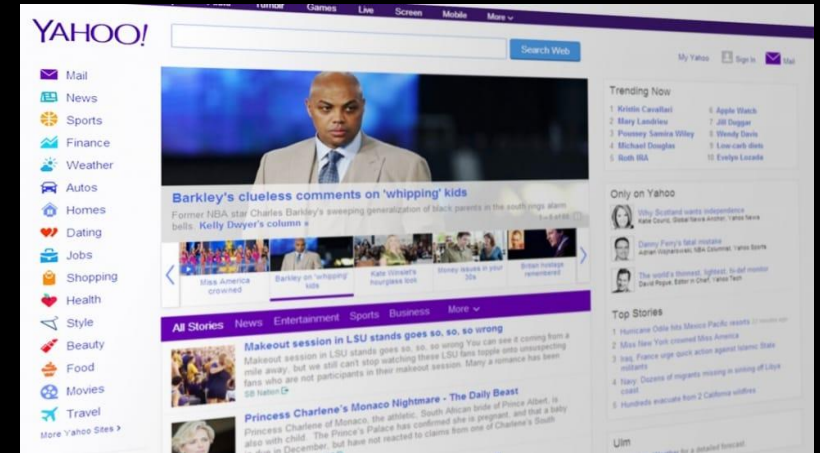




# **BASICS OF WEB DEVELOPMENT**

# ● WHAT IS WEB DEVELOPMENT ?

- **AS DEFINED BY THE WEBSITE GEEKSFORGEEKS.ORG:**
  - **WEB DEV REFERS TO THE CREATING, BUILDING, AND MAINTAINING OF WEBSITES.**
  - **IT INCLUDES ASPECTS SUCH AS WEB DESIGN, WEB PUBLISHING, WEB PROGRAMMING, AND DATABASE MANAGEMENT.**
  - **IT IS THE CREATION OF AN APPLICATION THAT WORKS OVER THE INTERNET (I.E. WEBSITES).**

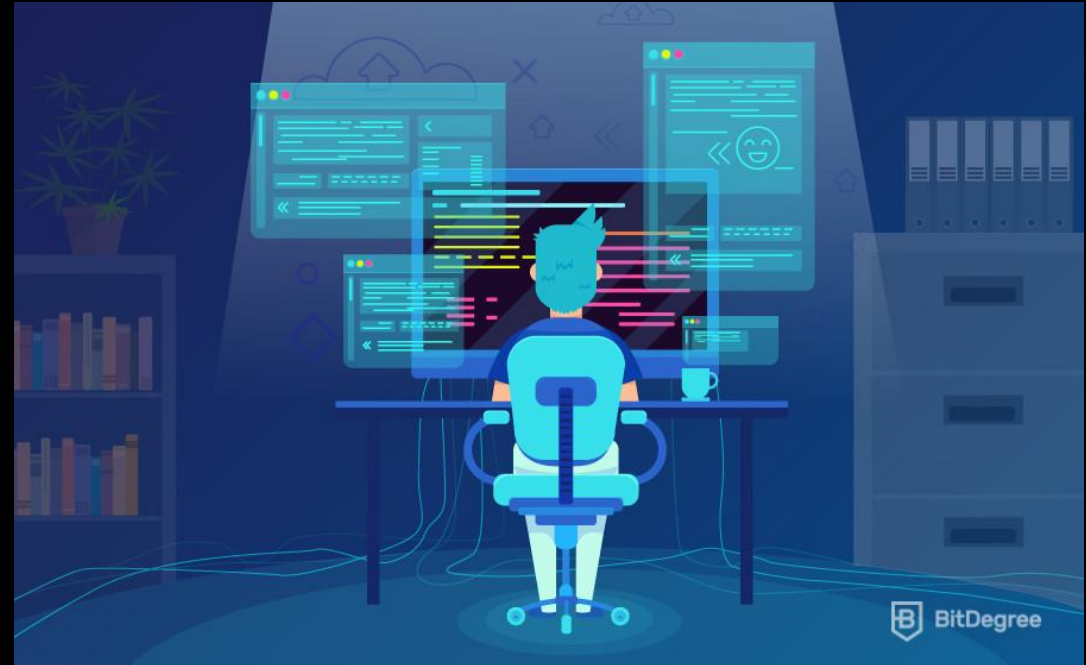


# ● WHAT IS WEB DEVELOPMENT? (CONT.)

- THE DESCRIPTION OF A WEB DEVELOPER AS OUTLINED BY O\*NET

IS AS FOLLOWS:

- “[WEB DEVELOPERS] DEVELOP AND IMPLEMENT WEBSITES, WEB APPLICATIONS, APPLICATION DATABASES, AND INTERACTIVE WEB INTERFACES.
- EVALUATE CODE TO ENSURE THAT IT IS PROPERLY STRUCTURED, MEETS INDUSTRY STANDARDS, AND IS COMPATIBLE WITH BROWSERS AND DEVICES.
- OPTIMIZE WEBSITE PERFORMANCE, SCALABILITY, AND SERVER-SIDE CODE AND PROCESSES...”



# ● WEB DEV EMPLOYMENT TRENDS

- PER O\*NET:

- MEDIAN WAGES:

- US:

- \$37.78/HOUR, \$78,580/YEAR

- NEW YORK

- \$33.87/HOUR, \$70,450/YEAR

- PROJECTED GROWTH (2021-2031)

- 22% IN NY, 30% IN US

- MUCH FASTER THAN AVERAGE

- BRIGHT OUTLOOK FOR THE FUTURE

## New York Employment Trends

15-1254.00 - [Web Developers](#) 🌟 **Bright Outlook**

View trends for state: New York

Go

### In New York:

Employment data for **Web Developers and Digital Interface Designers**.

Employment (2020)	14,410 employees
Projected employment (2030)	17,520 employees
Projected growth (2020-2030)	22%
Projected annual job openings (2020-2030)	1,470

### In the United States:

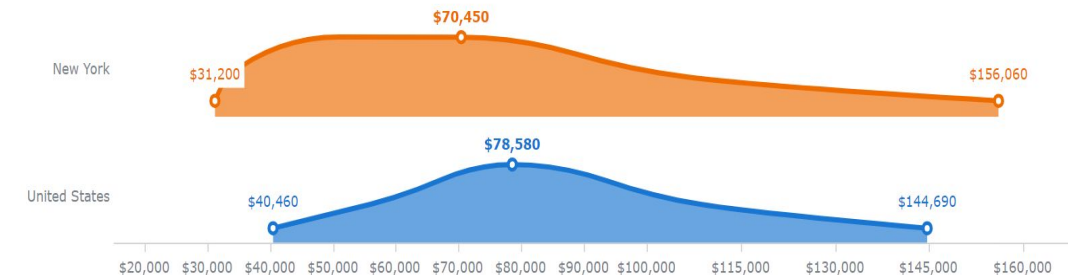
Employment data for **Web Developers**.

Employment (2021)	95,300 employees
Projected employment (2031)	124,100 employees
Projected growth (2021-2031)	■ ■ ■ ■ 30% Much faster than average
Projected annual job openings (2021-2031)	11,000

New York source: Projections Central [2020-2030 long-term projections](#). United States source: Bureau of Labor Statistics [2021-2031 employment projections](#). "Projected growth" represents the estimated change in total employment over the projections period. "Projected annual job openings" represent openings due to growth and replacement.

Annual Wages

Hourly Wages





# **TYPES OF WEB DEV**

# ● FRONT-END WEB DEV

- INVOLVES THE DEVELOPMENT OF THE GRAPHICAL USER INTERFACE (UI) OF A WEBSITE APPLICATION, SO THAT USER CAN VIEW AND INTERACT WITH THAT WEBSITE.

- “CLIENT-SIDE” OF THE APP

- NECESSARY SKILLS & TOOLS:

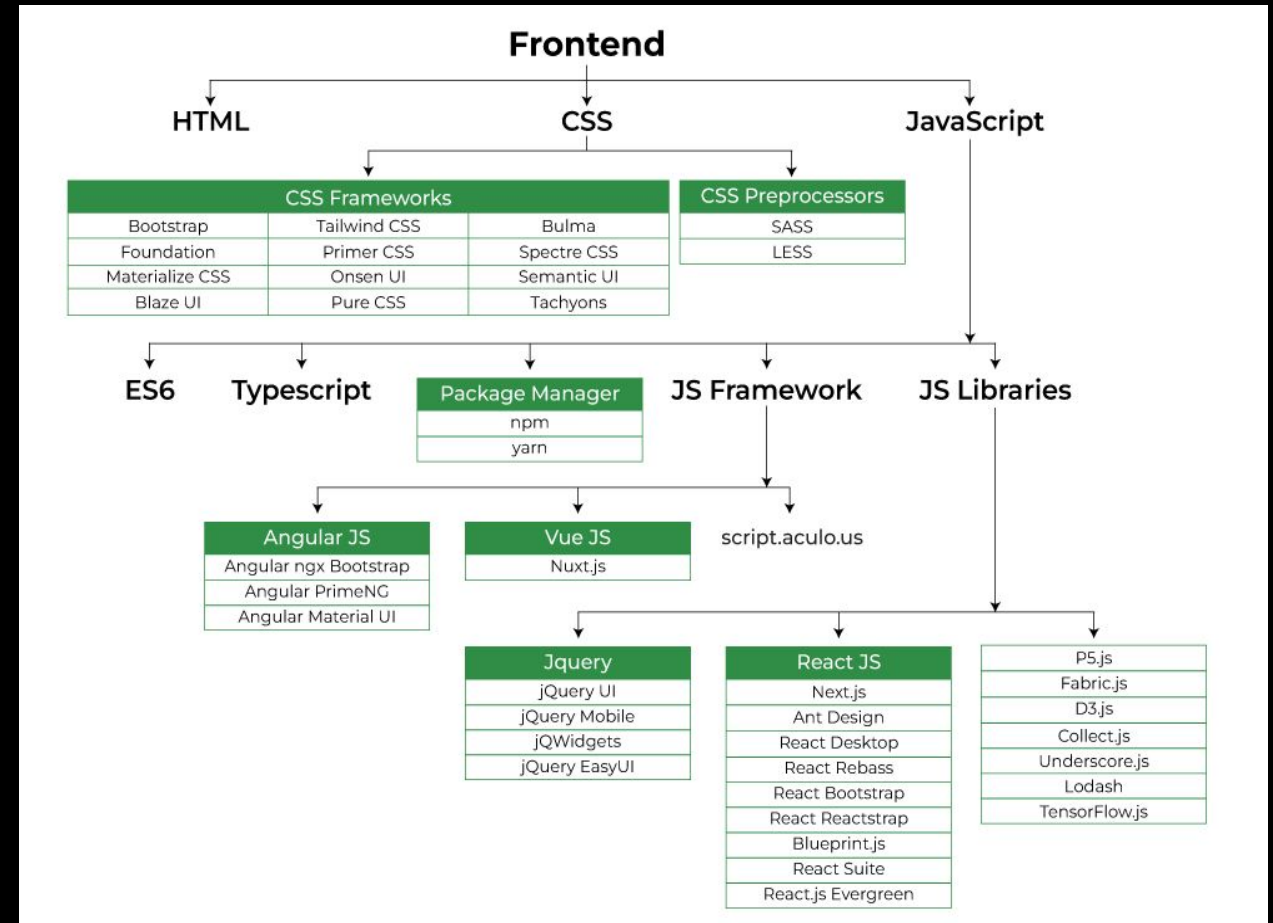
- LANGUAGES:

- HTML, CSS, JAVASCRIPT

- FRAMEWORKS/LIBRARIES:

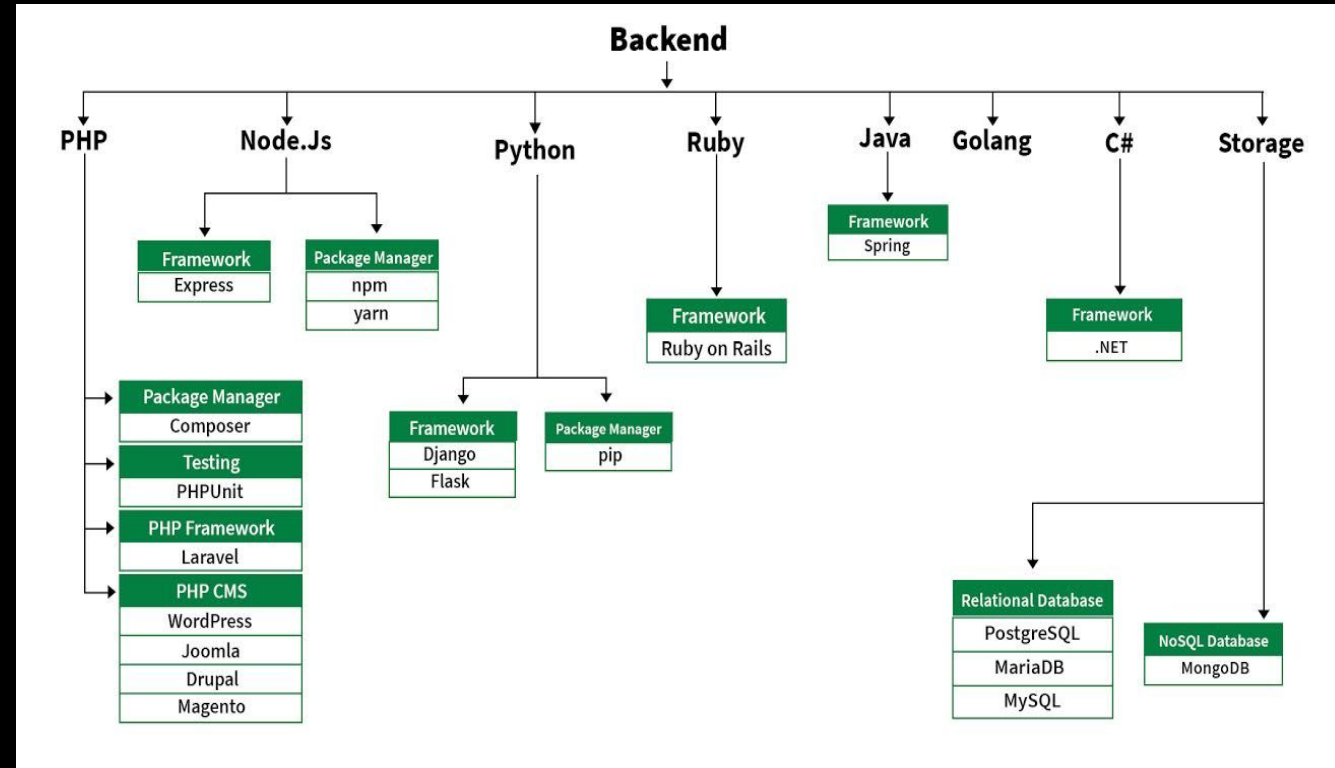
- BOOTSTRAP, TAILWIND CSS

- ES6, TYPESCRIPT, ANGULARJS, REACT



# ● BACK-END WEB DEV

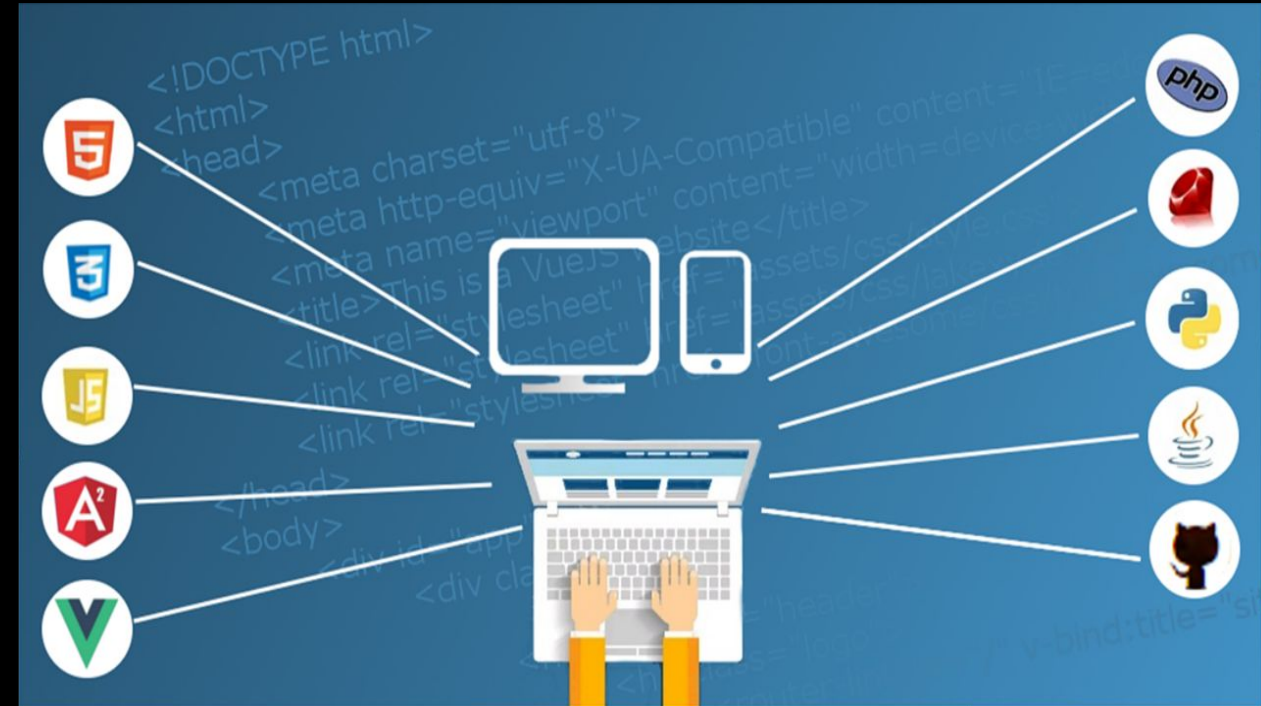
- INVOLVES THE DEVELOPMENT OF THE SERVER-SIDE OF THE WEBSITE.
- IT IS THE PART OF THE WEBSITE THAT IS INVISIBLE, NON-INTERACTIVE, DOES NOT COME INTO DIRECT CONTACT WITH USERS.
- NECESSARY SKILLS & TOOLS:
  - LANGUAGES:
    - PHP, JAVA, PYTHON, RUBY, C#
  - FRAMEWORKS/LIBRARIES:
    - NODE.JS, DJANGO, FLASK
    - MONGODB, MYSQL, .NET





# ● FULL-STACK WEB DEV

- INVOLVES DEVELOPING BOTH THE CLIENT-SIDE AND THE SERVER-SIDE OF THE WEBSITE.
  - FULL-STACK = FRONT-END + BACK-END
- FULL-STACK WEB DEVS TYPICALLY USE STACKS TO COMPLETE THEIR PROJECTS.
- POPULAR STACKS:
  - LAMP
    - LINUX, APACHE, MYSQL, PHP
  - MEAN/MERN
    - MONGODB, EXPRESS.JS, ANGULAR/REACT, NODE.JS

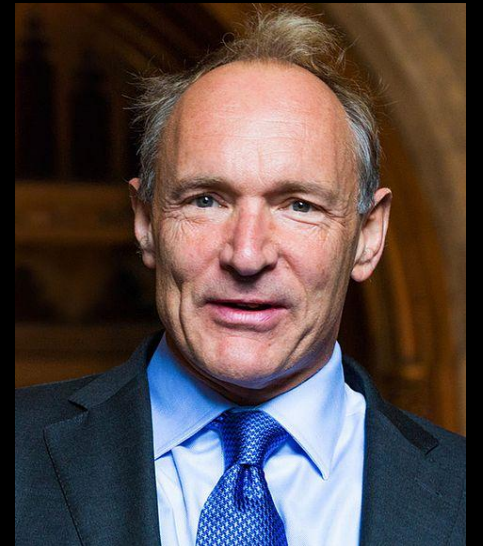




# INTRO TO HTML

# ● WHAT IS HTML ?

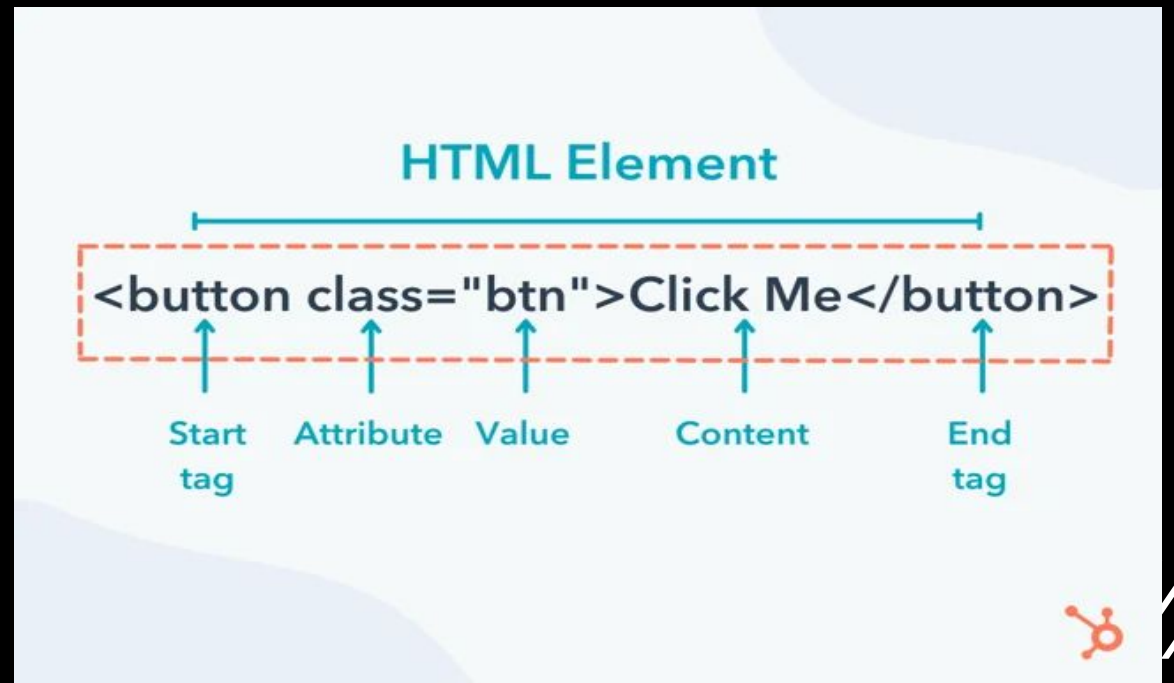
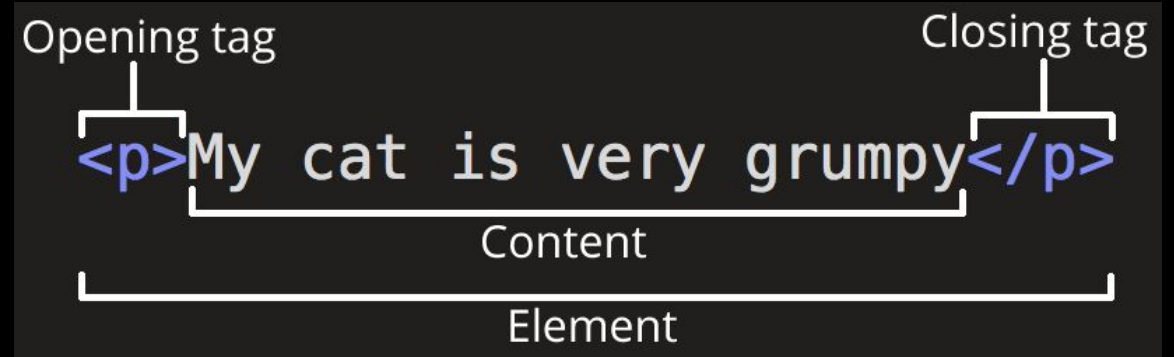
- **HTML (AKA HYPERTEXT MARKUP LANGUAGE) IS THE STANDARD MARKUP LANGUAGE FOR DOCUMENTS DESIGNED TO BE DISPLAYED IN A WEB BROWSER.**
  - **DEFINES THE MEANING AND STRUCTURE OF WEB CONTENT**
- **FAST FACTS:**
  - **DEVELOPMENT START: 1989**
  - **INITIAL RELEASE: 1993**
  - **LATEST VERSION: HTML5, RELEASED IN 2008**
  - **CREATORS: TIM BERNERS-LEE, ROBERT CAILLIAU, OTHERS**



# ● BASICS OF HTML - ELEMENTS

- **ELEMENTS:**

- **BASIC COMPONENTS OF AN HTML DOCUMENT THAT TELLS THE WEB BROWSER HOW TO STRUCTURE & INTERPRET PARTS OF IT.**
- **TYPICALLY MADE UP OF AN OPENING/CLOSING TAG, ATTRIBUTES, AND CONTENT.**
  - **TAGS: MARK UP THE START OF AN ELEMENT, USUALLY ENCLOSED IN ANGLE BRACKETS.**
  - **ATTRIBUTES: CONTAIN ADDITIONAL PIECES OF INFORMATION.**
  - **CONTENT: STUFF INSIDE OF THE ELEMENT.**



# ● BASICS OF HTML - ELEMENTS (CONT.)

- FUN FACTS:

- THERE ARE ABOUT 100+ HTML TAGS THAT EXIST AND THAT ARE USEABLE.

- HTMLREFERENCE.IO LISTS 113 HTML TAGS (VARIES FROM SITE TO SITE)

- KEY THINGS TO REMEMBER:

- THE VAST MAJORITY OF TAGS MUST BE OPENED AND CLOSED WITH CONTENT RESTING IN BETWEEN THE TAGS.
- WHEN USING MULTIPLE TAGS, THEY MUST BE CLOSED IN THE ORDER IN WHICH THEY WERE OPENED.

Search 113 HTML elements				
<input checked="" type="checkbox"/> experimental	<input checked="" type="checkbox"/> meta	<input checked="" type="checkbox"/> self-closing	<input checked="" type="checkbox"/> inline	<input checked="" type="checkbox"/> block
a	meta	self-closing	inline	block
abbr	meta	self-closing	inline	block
address	meta	self-closing	inline	block
area	meta	self-closing	inline	block
article	meta	self-closing	inline	block
aside	meta	self-closing	inline	block
audio	meta	self-closing	inline	block
b	meta	self-closing	inline	block
base	meta	self-closing	inline	block
bdi	meta	self-closing	inline	block
bdo	meta	self-closing	inline	block
blockquote	meta	self-closing	inline	block
body	meta	self-closing	inline	block
br	meta	self-closing	inline	block



# ● BASICS OF HTML - PAGE CONSTRUCTION

- **<!DOCTYPE HTML>**
  - THIS IS A DOCUMENT TYPE DECLARATION, WHICH SPECIFIES THE LANGUAGE YOU WILL BE WRITING IN.
  - SHOULD ALWAYS GO ON THE VERY FIRST LINE OF CODE.
- **<HTML>**
  - DEFINES THE ROOT ELEMENT OF HTML DOCUMENT.
  - ALL OTHER ELEMENTS MUST BE CONTAINED WITHIN THIS ELEMENT.



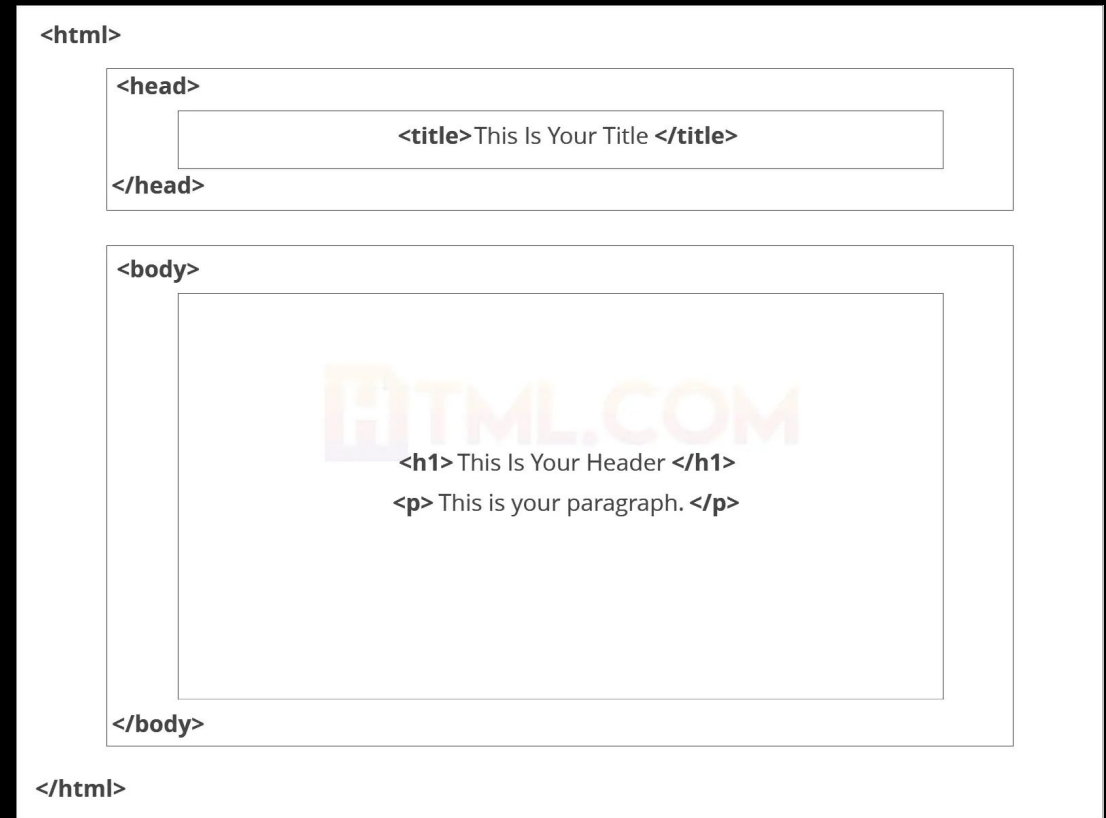
# ● BASICS OF HTML - PAGE CONSTRUCTION (CONT.)

- **<HEAD>**

- **DEFINES A CONTAINER FOR A WEB PAGE'S METADATA.**
- **IN OTHER WORDS, THIS IS WHERE ALL THE METADATA GOES, SO SEARCH ENGINES & OTHER PROGRAMS CAN USE THEM.**

- **<BODY>**

- **THE CONTAINER FOR THE WEB PAGE'S CONTENT.**
- **WHERE ELEMENTS LIKE <P> , <H1>, AND <TITLE> RESIDE.**



# ● BASICS OF HTML - OTHER IMPORTANT ELEMENTS

- HEADINGS (<H1> -<H6>)

- USED TO CREATE HEADINGS, DEPENDING ON LEVEL OF IMPORTANCE. <H1> = MOST IMPORTANT, <H6> = LEAST IMPORTANT

- ANCHOR (<A>)

- CREATES A LINK TO A URL
  - ABLE TO TARGET WEB PAGES, SECTIONS WITHIN A PAGE,, EMAIL ADDRESSES, ETC.

- IMAGE (<IMG>)

- USED TO INSERT IMAGES INTO A WEB PAGE.

- METADATA (<META>)

- DEFINES THE METADATA ATTACHED TO A WEB PAGE,

Element	Meaning	Purpose
<b>	Bold	Highlight important information
<strong>	Strong	Similarly to bold, to highlight key text
<i>	Italic	To denote text
<em>	Emphasised Text	Usually used as image captions
<mark>	Marked Text	Highlight the background of the text
<small>	Small Text	To shrink the text
<strike>	Striked Out Text	To place a horizontal line across the text
<u>	Underlined Text	Used for links or text highlights
<ins>	Inserted Text	Displayed with an underline to show an inserted text
<sub>	Subscript Text	Typographical stylistic choice
<sup>	Superscript Text	Another typographical presentation style







# INTRO TO CSS

# ● LIVE VISUAL: HTML WITH/WITHOUT CSS

## WEBSITE



with HTML  
& CSS

with  
HTML  
only



HTML

CSS

# ● WHAT IS CSS?

- CSS (AKA CASCADING STYLE SHEETS) IS THE STYLE SHEET LANGUAGE USED TO DESCRIBE THE PRESENTATION OF WEB PAGES.
- FAST FACTS:
  - PROPOSED: 1994
  - RELEASED: DECEMBER 17TH, 1996
  - LATEST RELEASE: CSS3, RELEASED IN 1999
  - CREATORS: HAKON WILM LIE, BERT BOS, WORLD WIDE WEB CONSORTIUM (W3C)



# ● BASICS OF CSS - WHAT DOES IT DO?

- CSS IS USED TO DEFINE STYLES FOR WEB PAGES:
  - CONTROLS THE LAYOUT, TYPOGRAPHY, COLORS, AND OTHER VISUAL ASPECTS
  - CREATES RESPONSIBLE AND SCALABLE LAYOUTS THAT LOOK GOOD ON ANY DEVICE
  - ADDS VISUAL EFFECTS SUCH AS ANIMATIONS, TRANSITIONS, AND OTHER VISUAL EFFECTS
- CSS SAVES TIME AND EFFORT:
  - CAN CONTROL THE LAYOUT OF MULTIPLE WEB PAGES ALL AT ONCE
- CSS SOLVED BIG PROBLEMS:
  - INCONSISTENCY, SCALABILITY, ACCESSIBILITY, ETC.



# ● BASICS OF CSS - SYNTAX

- **SELECTOR:**

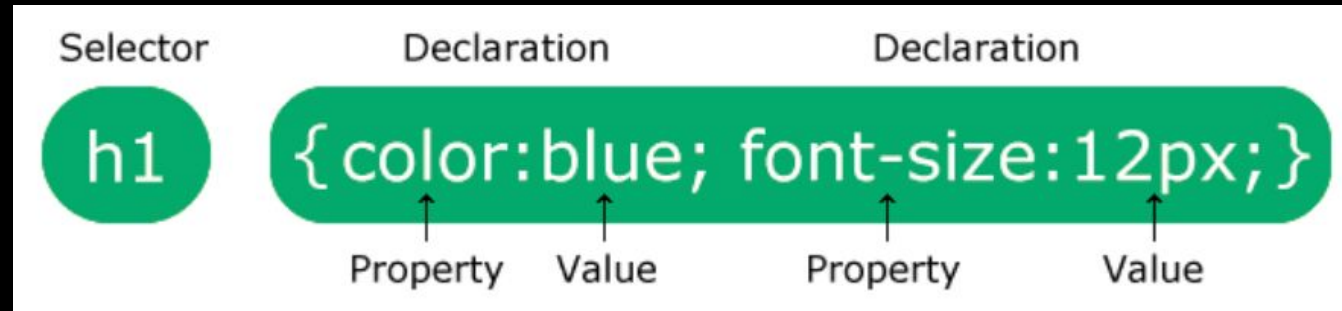
- POINTS TO THE HTML ELEMENT THAT YOU ARE TRYING TO STYLE

- **PROPERTY:**

- CHARACTERISTIC OF AN HTML ELEMENT THAT CAN BE STYLED USING CSS. COLOR IS AN EXAMPLE OF A PROPERTY.

- **PROPERTY VALUE**

- A VALUE THAT IS ASSIGNED TO A CSS PROPERTY. FOR INSTANCE, IF YOU ASSIGN “BLUE” TO THE COLOR PROPERTY, THE TEXT IN A GIVEN ELEMENT WILL TURN BLUE.



```
body {  
  background-color: ■ #c8f7ea;  
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
}
```





# ● BASICS OF CSS - HOW TO ADD TO HTML

- THERE ARE THREE WAYS TO INSERT CSS INTO AN HTML DOC:

- EXTERNAL CSS

- TYPICALLY IN A SEPARATE FILE, CAN BE USED TO CONTROL THE STYLE OF ENTIRE WEBSITES
    - EACH HTML PAGE MUST INCLUDE A REFERENCE TO THE EXTERNAL STYLE SHEET FILE INSIDE THE <LINK> ELEMENT, INSIDE THE HEAD SECTION.

- INTERNAL CSS

CAN BE USED TO CONTROL INDIVIDUAL HTML PAGES IF THEY HAVE A UNIQUE STYLE

- THE INTERNAL STYLE IS DEFINED INSIDE THE <STYLE> ELEMENT, INSIDE THE HEAD SECTION.

- INLINE CSS

- CAN BE USED TO CONTROL THE STYLE OF INDIVIDUAL ELEMENTS .
    - ADD THE STYLE ATTRIBUTE TO THE RELEVANT ELEMENT. THE STYLE ATTRIBUTE CAN CONTAIN ANY CSS PROPERTY.



# ● BASICS OF CSS - HOW TO ADD TO HTML CONT.

- EXTERNAL CSS EXAMPLE:

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

- INTERNAL CSS EXAMPLE:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: linen;
}

h1 {
    color: maroon;
    margin-left: 40px;
}
```



# ● BASICS OF CSS - HOW TO ADD TO HTML CONT.

- **INLINE CSS EXAMPLE:**

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```





# ● BASICS OF CSS - SELECTORS

- CSS SELECTORS ARE USED TO FIND/SELECT THE HTML ELEMENTS

YOU WANT TO STYLE.

- FIVE CATEGORIES OF SELECTORS:

- SIMPLE SELECTORS

- SELECT BASED ON NAME, ID , CLASS

- COMBINATOR SELECTORS

- SELECT BASED ON SPECIFIC RELATIONSHIPS

- PSEUDO-CLASS SELECTORS

- SELECT BASED ON A CERTAIN STATE

- PSEUDO-ELEMENTS SELECTORS

- SELECT & STYLE PART OF AN ELEMENT

- ATTRIBUTE SELECTORS

- SELECT BASED ON ATTRIBUTE & VALUE

We can divide CSS selectors into five categories:

- Simple selectors (select elements based on name, id, class)
- Combinator selectors (select elements based on a specific relationship between them)
- Pseudo-class selectors (select elements based on a certain state)
- Pseudo-elements selectors (select and style a part of an element)
- Attribute selectors (select elements based on an attribute or attribute value)



# ● BASICS OF CSS - SELECTOR EXAMPLES

- SIMPLE SELECTORS:

```
p {  
  text-align: center;  
  color: red;  
}
```

```
.center {  
  text-align: center;  
  color: red;  
}
```

```
#para1 {  
  text-align: center;  
  color: red;  
}
```

- COMBINATOR SELECTORS:

```
div p {  
  background-color: yellow;  
}
```

```
div > p {  
  background-color: yellow;  
}
```

```
div + p {  
  background-color: yellow;  
}
```

- PSEUDO-CLASS SELECTORS:

```
selector:pseudo-class {  
  property: value;  
}
```

```
a.highlight:hover {  
  color: #ff0000;  
}
```

```
div:hover {  
  background-color: blue;  
}
```



# ● BASICS OF CSS - SELECTOR EXAMPLES CONT.

- PSEUDO-ELEMENTS SELECTORS:

```
selector::pseudo-element {  
    property: value;  
}
```

```
p::first-line {  
    color: #ff0000;  
    font-variant: small-caps;  
}
```

```
p::first-letter {  
    color: #ff0000;  
    font-size: xx-large;  
}
```

- ATTRIBUTE SELECTORS:

```
a[target] {  
    background-color: yellow;  
}
```

```
a[target="_blank"] {  
    background-color: yellow;  
}
```

```
[title~="flower"] {  
    border: 5px solid yellow;  
}
```



# ● BASICS OF CSS - IMPORTANT PROPERTIES

HERE ARE SOME IMPORTANT PROPERTIES AND THE POTENTIAL VALUES

THAT CAN BE ASSIGNED TO THEM:

Property	Property Value
color	red, blue, green, #ffffff, rgb(255, 0, 0), etc.
font-family	Arial, Helvetica, Times New Roman, etc.
font-size	12px, 16px, 20px, etc.
width	100px, 50%, auto, etc.
height	100px, 50%, auto, etc.
margin	10px, 20px, auto, etc.
padding	10px, 20px, auto, etc.
background-color	red, blue, green, #ffffff, rgb(255, 0, 0), etc.
background-image	none, url(image.png), linear-gradient(red, blue), etc.



# ● BASICS OF CSS - COLOR PROPERTY

- COLORS ARE SPECIFIED IN TWO DIFFERENT WAYS:

- PREDEFINED COLOR NAMES

- HTML/CSS BOTH SUPPORT 140 COLOR NAMES  
(TOMATO, ORANGE, DODGERBLUE, ETC.)

- RGB/RGBA, HEX, AND HSL/HSLA VALUES

- RGB: (RED, GREEN, BLUE)
- RGBA: (RED, GREEN, BLUE, ALPHA)
- HEX: HEXADECIMAL VALUE (#FFFFFF)
- HSL: (HUE, SATURATION, LIGHTNESS)
- HSLA: (HUE, SATURATION, LIGHTNESS, ALPHA)

<code>rgb(255, 0, 0)</code>	<code>rgb(0, 0, 255)</code>
<code>rgb(60, 179, 113)</code>	<code>rgb(238, 130, 238)</code>
<code>rgb(255, 165, 0)</code>	<code>rgb(106, 90, 205)</code>

<code>#ff0000</code>	<code>#0000ff</code>
<code>#3cb371</code>	<code>#ee82ee</code>
<code>#ffa500</code>	<code>#6a5acd</code>

<code>hsl(0, 100%, 50%)</code>	<code>hsl(240, 100%, 50%)</code>
<code>hsl(147, 50%, 47%)</code>	<code>hsl(300, 76%, 72%)</code>
<code>hsl(39, 100%, 50%)</code>	<code>hsl(248, 53%, 58%)</code>



# ● BASICS OF CSS - HEIGHT/WIDTH/MARGIN/PADDING

- THESE PROPERTIES ARE TYPICALLY USED TO CHANGE THE ENVIRONMENT AROUND AN HTML ELEMENT.
  - HEIGHT/WIDTH:
    - SET HEIGHT/WIDTH OF AN ELEMENT.
  - MARGIN:
    - CREATES SPACE AROUND ELEMENTS, OUTSIDE OF ANY DEFINED BORDERS.
  - PADDING:
    - CREATES SPACE AROUND AN ELEMENT'S CONTENT, INSIDE OF ANY DEFINED BORDERS.

```
div {  
  height: 200px;  
  width: 50%;  
  background-color: powderblue;  
}
```

```
p {  
  margin-top: 100px;  
  margin-bottom: 100px;  
  margin-right: 150px;  
  margin-left: 80px;  
}
```

```
div {  
  padding-top: 50px;  
  padding-right: 30px;  
  padding-bottom: 50px;  
  padding-left: 80px;  
}
```





# **BRIEF OVERVIEW OF JS**



# ● WHAT IS JS?

- JAVASCRIPT (AKA JS) IS A HIGH-LEVEL PROGRAMMING LANGUAGE THAT IS ONE OF THE CORE TECHNOLOGIES OF THE WORLD WIDE WEB, ALONG WITH HTML/CSS.
- FAST FACTS:
  - RELEASED: DECEMBER 4TH, 1995
  - CREATOR: BRENDAN EICH
  - CONFORMS TO ECMASCRIPT STANDARD.
  - LATEST RELEASE: ECMASCRIPT 2023, JUNE 2023
  - JAVASCRIPT != JAVA





# ● BASICS OF JS - WHAT DOES IT DO?

- JS, WITH HTML/CSS, CAN:
  - CHANGE HTML CONTENT
  - CHANGE HTML ATTRIBUTE VALUES
  - CHANGE CSS STYLES
  - SHOW/HIDE HTML ELEMENTS
- JS CAN ALSO:
  - ADD INTERACTIVITY TO WEBSITES
  - CREATE ANIMATIONS AND VISUAL EFFECTS
  - VALIDATE FORMS
  - COMMUNICATE WITH SERVERS



# ● BASICS OF JS - HOW TO ADD TO HTML/CSS?

- TO INSERT JS INTO HTML:
  - USE THE `<SCRIPT>` TAG
  - USE JS HTML METHODS SUCH AS `GETELEMENTID()`
- JS SCRIPTS CAN BE INSERTED INTO EITHER THE `<HEAD>` OR `<BODY>` ELEMENTS, OR BOTH.
  - JS FUNCTIONS CAN ALSO BE INSERTED IN HERE
  - HOWEVER, IT MIGHT BE MORE ADVANTAGEOUS TO PLACE SCRIPTS IN EXTERNAL FILES

```
<script>
document.getElementById("demo").innerHTML = "My First JavaScript";
</script>
```

```
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
```

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
</head>
<body>

<h2>Demo JavaScript in Head</h2>

<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>

</body>
</html>
```



# ● BASICS OF JS - SYNTAX

- JS DEFINES TWO TYPES OF VALUES:
  - FIXED/LITERAL & VARIABLES
- FIXED/LITERAL:
  - NUMBERS = WRITTEN WITH/WITHOUT DECIMALS
  - STRINGS = TEXT, WRITTEN IN SINGLE/DOUBLE QUOTES
- VARIABLES:
  - USED TO STORE DATA VALUES
  - KEYWORDS: VAR, LET, & CONST (ALL LOWERCASE)
- JS IS ALSO CASE-SENSITIVE, BECAUSE ALL IDENTIFIERS ARE CASE SENSITIVE
- JS COMMENTS = JAVA COMMENTS
- DON'T FORGET YOUR SEMICOLON!

```
// How to create variables:  
var x;  
let y;  
  
// How to use variables:  
x = 5;  
y = 6;  
let z = x + y;
```

```
const PI = 3.141592653589793;  
PI = 3.14;           // This will give an error  
PI = PI + 10;        // This will also give an error
```



# ● BASICS OF JS - OUTPUT/DISPLAY

- JS CAN DISPLAY DATA IN FOUR WAYS:

- WRITING INTO HTML ELEMENT

- WRITING INTO HTML OUTPUT

- WRITING INTO AN ALERT BOX

- WRITING INTO CONSOLE

JavaScript can "display" data in different ways:

- Writing into an HTML element, using `innerHTML` .
- Writing into the HTML output using `document.write()` .
- Writing into an alert box, using `window.alert()` .
- Writing into the browser console, using `console.log()` .



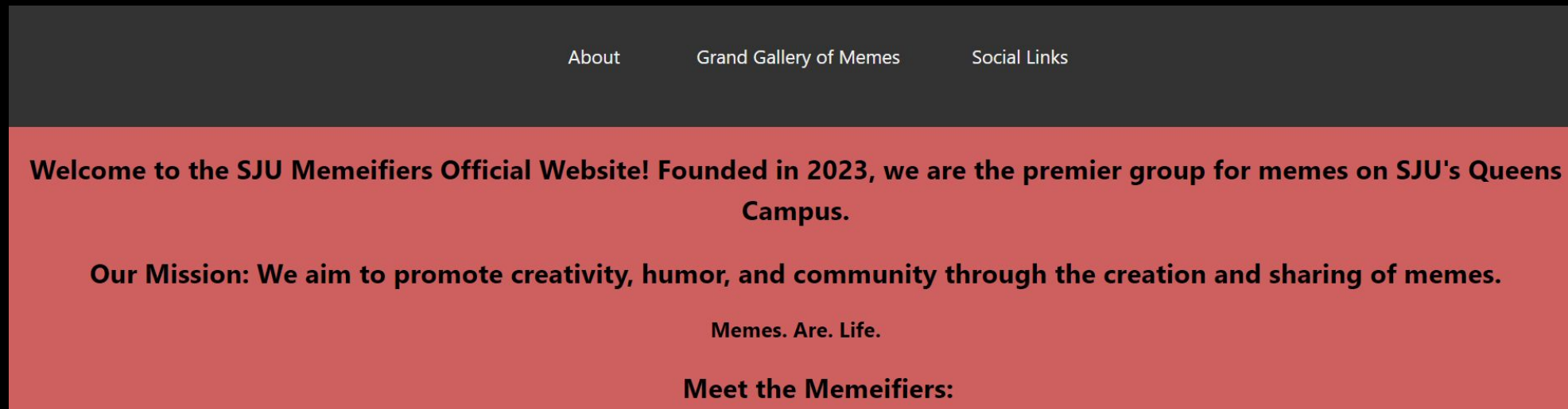


**DEMO TIME!**



# DEMO OVERVIEW

- WE WILL BE LOOKING AT A DEMO WEBSITE THAT I CREATED  
USING HTML/CSS:
- 45.33.85.242





# RESOURCES

- **WEB DEV RESOURCES:**

- **O\*NET:**

- [HTTPS://WWW.ONETONLINE.ORG/LINK/SUMMARY/15-1254.00](https://www.onetonline.org/link/summary/15-1254.00)
    - [HTTPS://WWW.ONETONLINE.ORG/LINK/LOCALTRENDS/15-1254.00?ST=NY](https://www.onetonline.org/link/localtrends/15-1254.00?ST=NY)

- **GEEKS FOR GEEKS:**

- [HTTPS://WWW.GEEKSFORGEEKS.ORG/WEB-DEVELOPMENT/](https://www.geeksforgeeks.org/web-development/)

- **UDACITY:**

- [HTTPS://WWW.UDACITY.COM/BLOG/2020/12/Front-End-vs-Back-End-vs-Full-Stack-Web-Developers.html](https://www.udacity.com/blog/2020/12/front-end-vs-back-end-vs-full-stack-web-developers.html)





# RESOURCES

- **HTML:**

- [HTML.COM](https://www.html.com)
- [HTTPS://DEVELOPER.MOZILLA.ORG/EN-US/DOCS/LEARN/HTML/INTRODUCTION\\_TO\\_HTML/GETTING\\_STARTED](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started)
- [HTTPS://WWW.DIGITALOCEAN.COM/COMMUNITY/TUTORIALS/WHAT-IS-AN-HTML-ELEMENT](https://www.digitalocean.com/community/tutorials/what-is-an-html-element)
- [HTTPS://HTMLREFERENCE.IO/](https://htmlreference.io/)
- [HTTPS://DEVPRACTICAL.COM/HOW-MANY-HTML-TAGS/](https://devpractical.com/how-many-html-tags/)
- [HTTPS://WWW.W3SCHOOLS.COM/HTML/DEFAULT.ASP](https://www.w3schools.com/html/default.asp)
- [HTTPS://HTML.SPEC.WHATWG.ORG/MULTIPAGE/](https://html.spec.whatwg.org/multipage/)
- [HTTPS://BOOKS.GOALKICKER.COM/HTML5BOOK/](https://books.goalkicker.com/html5book/)
- [HTTPS://WWW.FREECODECAMP.ORG/LEARN/2022/RESPONSIVE-WEB-DESIGN/](https://www.freecodecamp.org/learn/2022/responsive-web-design/)

- **CSS:**

- [HTTPS://WWW.W3SCHOOLS.COM/CSS/DEFAULT.ASP](https://www.w3schools.com/css/default.asp)
- [HTTPS://DEVELOPER.MOZILLA.ORG/EN-US/DOCS/WEB/CSS](https://developer.mozilla.org/en-US/docs/Web/CSS)
- [HTTPS://WWW.DOFACTORY.COM/CSS/PROPERTIES](https://www.dofactory.com/css/properties)
- [HTTPS://WWW.W3.ORG/STYLE/CSS/SPECS.EN.HTML](https://www.w3.org/style/css/specs.en.html)
- [HTTPS://BOOKS.GOALKICKER.COM/CSSBOOK/](https://books.goalkicker.com/cssbook/)







# RESOURCES

- **JAVASCRIPT:**

- [HTTPS://BOOKS.GOALKICKER.COM/JAVASCRIPTBOOK/](https://books.goalkicker.com/javascriptbook/)
- [HTTPS://WWW.W3SCHOOLS.COM/JS/DEFAULT.ASP](https://www.w3schools.com/js/default.asp)
- [HTTPS://WWW.FREECODECAMP.ORG/LEARN/JAVASCRIPT-ALGORITHMS-AND-DATA-STRUCTURES/](https://www.freecodecamp.org/learn/javascript-algorithms-and-data-structures/)
- [HTTPS://DEVELOPER.MOZILLA.ORG/EN-US/DOCS/WEB/JAVASCRIPT](https://developer.mozilla.org/en-US/docs/Web/JavaScript)
- [HTTPS://WWW.FREECODECAMP.ORG/NEWS/23-FREE-WEBSITES-TO-LEARN-JAVASCRIPT/](https://www.freecodecamp.org/news/23-free-websites-to-learn-javascript/)





**THANK YOU!**

