

## HTML Intro: Structure (file & code snippet)

- ② HTML Syntax ③ Common Terminology (Elements, Tags, Content, Attributes (name="value")).

## HTML Elements: ① Text {Align, Color, Decoration} <sup>← CSS</sup>

- ① Container (Div), ② Head (h {1-5}), ③ Paragraph (p) ④ button  
④ Image (img (src & alt)) ⑤ Anchor (a (target (blank) & href = "link/path"))  
⑥ List: ⑥ li (list-style-type) ⑦ ordered list (ol and values (Numeric/Decimal, Alpha, Roman) → (lower upper) case.  
⑧ unordered list (ul and values (shape (Circle, Disc, Square and None))) ⑦ Horizontal Rule (hr) ⑧ Line Break (br).

## HTML Attributes: Class, Style, onclick <sup>⑨ Span (<span> highlights text)</sup>

- ① Class: used to define CSS selectors (id, type, class(·))  
② Style: used to define CSS (property: value) for inline styling.  
③ onclick: used to Integrate webpage / Migrate the code from one page container other page without hyperlinks.

Note: Syntax: onclick="display('sectionId')"

Steps: Since, its (CBP display utility Kit funcn first we need to add the JS code snippet at bottom "before" body tag ends & "After frontend code ends".

- ② while defining ID place the frontend code in a container & name the container id with (id="sectionId")  
③ Use the onclick attribute according to page navigation inside button tag <btn onclick="display('sectionId')"></btn>

Note: HTML (child vs parent) Element, Types of Element (void, Block and Inline)

Inline: occupies a portion of space & starts in same line

Block: occupies a full line space & starts in new line

Bootstrap for frontend: ① Including Bootstrap  
② Flex Box Concept (Container, Direction (row & column), Position (start, center, end), ③ Bootstrap components and utilities: a) Carousels, b) Embedded links

\* flex Box Concept [Container (d-flex), Direction (flex (row, column), Position (justify-content (start/center/end))]

④ Bootstrap Grid System (Container & row & col-byout) BGS (layouts & break points)  $\Rightarrow$  layouts (none, sm, md, lg, xl) and breakpoints (limit combination upto "12" brk-pts) BGS (1 to 12) & (1 to 5) Spacers

⑤ Bootstrap for CSS [button (btn), background (bg), Box Spacing (margin (m) & T, B, L, R; padding (P & T, B, L, R)]

Note: \* layouts for device sizes (sm, md, lg, xl, none (used))

\* Breakpoints for BGS (Limit: 12 (1 to 12))

\* Breakpoints for Spacers (1 to 5) \*

⑥ Bootstrap utilities (spacing, color, sizing, display, flex position) and Bootstrap components (carousel, Modal, Gradient) Note: Syntax: "property-layout-breakpoint"

⑦ Bootstrap Spacing: a) Margin (mb, mt, ml, mr: (1 to 5)) b) padding (pb, pt, pl, pr: (1 to 5)) c) value: Auto.

⑧ Bootstrap Navbar: ① Nav element, ② Nav Item, ③ Nav link

⑨ Bootstrap Icons ⑩ Bootstrap Shadow utility (Shadow value (none, sm, shadow, lg) ⑪ BS (Flex) U [Order-layout-breakpoint (1 to 12)] ⑫ BS (display) U [d-layout / brk-pt (sm, md, lg, xl)] ⑬ BS (Position) U [fixed-top / bottom]



CSS Intro : ① CSS Syntax, file structure, common Terminology [Rule, Rule set, Selectors (property: value)  $\Rightarrow$  class, Id, type] ② CSS properties : Text (Align, decoration, color)

③ Background (size, Image, color) ④ Font (style, size, weight, family) ⑤ Border (style, width, radius, color) ⑥ Box (height, width, padding, Margin) ⑦ Units [pixel (px), viewport (height (vh), width (vw)), percentage (%)] ⑧ style [Inline, Internal and External]

⑨ `<link rel="stylesheet" href="file-path/link"/>`

⑩ CSS Box property value (Margin (top, Bottom, left, right) and padding (top, Bottom, left, right))

⑪ CSS concepts [(selectors: Id, class, type) and (Inheritance: ① Inheritance (Inherited & Non-Inherited elements) ② Specificity ("preference") (① Id  $\rightarrow$  ② class  $\rightarrow$  ③ type) value: ! Important (Applied After p:v) ③ Cascading (order of execution preference in CSS file (last one mostly preferred))]

⑫ color: transparent ⑬ CSS Bootstrap sizing for width, units (%) (w: (1 to 100)%)



## HTML:

## Contents of HTML:

- ① Elements, ② content, ③ Tags  
④ Attributes

conclusions:

Elements  $\Rightarrow$  (Content + Tags)

## Types of HTML :

Tags (Start & End)

Attributes (Name & value)

### Elements (void, Inline, Block)

### Examples of HTML Elements

Tags { Heading (h1), Paragraph (p),  
Button (button), Container (div), Image  
(img {src, alt}), List (ul, ol, li),  
anchor (a {href, target})

- ⑥ CSS concepts (specificity & Cascade)  
CSS styleline (Inline), K.W (! Importance)  
Attribute (style ↑)

a) Bootstrap utilities: Spacing, Color, Sizing, Display, Text, order, Hiding.

Responsive web design: Inherited (Text, list)  
Non-Inherited (Box, Background)

- ① Boot Grid system (containers, row and Breakpoints across devices (Extra small, small, medium, large, very large) and range (1 to 12))
- ② CSS Box Model spacing properties

### Summary:

## \*HTML elements, Attributes, Tags, Link, Syntax, Structure

→  $\text{lags}(h, p, \text{button})$

- \* CSS properties, Ruleset, Rules, selector, styles, structure, syntax

→ Gas properties:-

- ① Position ② Box ③ Background

- ④ Color ⑤ Font (Size) ⑥ View Port

- ⑦ Box - Border properly ⑧ Units (px, %)

To Be continued,

like padding (P, Pb, Pt, Pl, Pr)

and Margin ( $m, mb, mt, ml, mr$ ) wi

range(0,1,2,3,4,5) and find

formula  $\Rightarrow \text{Size} * \text{Spacer (i.e., lpx)}$

$$(0, 1, 2, 3, 4, 5) \Leftrightarrow (0, 0.25, 0.5, 1, 1.5, 3)$$

- ③ Design & Develop color palettes

- ④ Bootstrap navbar designed  
develop navbar components  
like (Navbar element, items, links)

- ⑤ HTML elements (Inline, Block and void) and Hyper Text link (a → href; img → src)

## CSS Building Blocks



# Introduction to HTML and CSS:

① HTML part: Common (elements, Attributes) Terminology, structure, syntax and three tags (h, p, button)

② CSS part: Common Terminology, structure, syntax and properties like (text-align, color, font)

C.T: (Rule set, Rules, selectors, styling types)

## CSS Box Model:

① properties & units: P:  $\rightarrow$  (Background, size & image), Box (height, width)

U:  $\rightarrow$  (viewport (height (vh) & width (vw)))

② Box properties: padding, margin

Border properties: Radius, width, color, style

Bootstrap: Block level (fullspace - New Line)  
Inline (Necessary space - same line)

① part 1: Bootstrap CSS reusable properties & Attributes for (color, text, position)

② part 2: Flex Box properties like flex container (d-flex), flex item direction (flex-row/column) and flex position (justify-content (end, start) and (center))

object: Building static web page using flex properties

# Developing Layouts:

Part 1: HTML (Image Tag, SRC Attribute), CSS (Margin (Top, Bottom, left, Right), Guide When to use HTML tags & CSS properties like (Margin, padding) Image and Two projects a (static Blog page and Tourism web page (static))

Part 2: Guide to HTML Tags and CSS properties depending on use cases, Building a static diwali card;

Part-3: Bootstrap components (Carousels and Embedded links) and Building a view section for tourism web site using Carousels.

Project: ① Blog page ② Diwali card ③ Tourism

## Website Integration:

\* Bio Section page.

\* Tourism page (Home, favourite places, Detail view) sections.

\* Tags (Br, h3, a), list style type

\* HTML Tag (anchor (a)) HTML Attribute (onclick)

\* HTML (list tags) ordered and unordered and list styling (ul, ol, li)

\* HTML Hyperlink tags

HTML basics, Elements,  
CSS basics, Properties

1 Bootstrap basics (Component & utilities)

Types of styling

1. Inline, Internal & External.

2. CSS concepts:

Cascade, Specificity, Inheritance

3. CSS selectors

Type, ID, Class, Inline

RWD, BGS, MCL(ex),

Bootstrap:

Components: Icon, button, carousel, Navbar, modal, etc.

Utilities: Display, Spacing, Color, Text, Size, etc.

HTML:

Text, Box, Border, Background, Element Content, Tags, Attributes

Identity, Mipdset, [False & True] Assumptions, optimist, pessimist, Realistic.

Backend Basics (Internet, client, server (DNS, APP), DB, IP Address, URL Breakdown (Top Level, Domain, Sub); Amazon Services (Route 53 (DNS), cloud front (Server), Storage (S3));

HTML (Elements, Content, Tags, Attributes (Name & value))

HTML(e): { Heading (h1), Paragraph (p), Button, container (div), Image (img) }

HTML: { Elements (Content + Tags (start + end)) => Inline, Block, void }

2. Attributes (Name & value) }

1. Responsive (WD), Bootstrap Grid System, Designing layouts using BGS (various)

2. Column wrapping, Responsive Break points (Combination + Class Name Prefixes)

3. Color palette, BGS Box property (Margin (T, B, R, L)), Bootstrap (U) { Margin (T, B, R, L), Background Color, etc. }

4. Bootstrap (Navbar), HTML (e) { Block, Inline }, CSS Box { margin (P, V), padding (P, V), border (P, V), background-color, etc. }

5. CSS concepts (Inheritance) & CSS selectors (Id, Class, Type, Inline)