#### HTML

* It is a Markup Language which is used to build webpages.
* It is implement through Elements.
* It works with **CSS** (for design) and **JavaScript** (for interactivity).

#### HTML Elements

* An HTML element is everything from the opening tag to the closing tag, including the content in between.

Ex: <p>Hello, World!</p>

#### HTML Attributes –

* HTML Elements can Have Attributes
* Attributes provides additional information about the element. Ex:

<html lang =”en”> Language is English

</html>

#### <!Doctype html>

* Document to be HTML 5.

#### HTML Page Structure:

<!DOCTYPE html>  Tells the version of the HTML

<html>  HTML Root Element

<head>  Used to contain Page HTML

metadata

<title>page title</title>  Title of Html page

</head>

<body>  hold content

<h1>hi</h1>  Holding head content

</body>

</html>

#### HTML 4:

HTML 4 vs HTML 5

**Release Date:** 1997

**Doctype** : Lengthy & Complex

**Vector Graphic** : requires plug-in like Flash **Audio & Video** : requires plug-in like Flash **Semantics :** Limited Semantic Elements **Graphics :** Limited External Plugins **Storage :** Cookies for Client-side Storage **Mobile Support:** Limited

**Form Controls :** Basic Control

**Scripting :** Heavy Reliant on External.

Drag & Drop Features not available

#### HTML 5:

**Release Date:** Oct -14

**Doctype :** <!DOCTYPE html>

**Vector Graphic :** Native Support <svg>

**Audio & Video :** Native Support <audio> & <video>

**Semantics :**Rich Semantic Elements like <article>,<nav>,<footer>

**Graphics :** Canvas API for 2D/3D Graphics

**Storage :** Local Storage, Session Storage

**Mobile Support:** Enhanced with mobile optimization

**Form Control :** New Form Controls like date, time, colour picker

**Scripting :** More Integrated Scripted Capabilities Drag & Drop Features available

## Tags

### Heading Tag:

* Heading Tags are used in HTML to define headings and subheadings on a webpage.
* They range from **`<h1>**` to **`<h6>`**, where **`<h1>`** is the highest (or most important) level heading and **`<h6>`** is the lowest (or least important).

### Default Sizes for Each Heading Tag 1 em = 16 pixels

**`<h1>`:** 2 em (32px)

**`<h2>`:** 1.5 em (24px)

**`<h3>`:** 1.17 em (18.72px)

**`<h4>`:** 1 em (16px)

**`<h5>`:** 0.83 em (13.28px)

**`<h6>`:** 0.67 em (10.72px)

### Paragraph Tag :

* The `<P>` tag in HTML stands for paragraph.
* It is used to define and group a block of text as a paragraph, separating it from other blocks of text on a webpage.

### Formatting Tag :

* Formatting tags are used in HTML to change the appearance of text or content, applying different styles such as bold, italic, underline, and more.

#### List of Formatting Tags:

1. **<b>** - Displays the content in bold format.
2. **<strong>** - Displays the content in bold format, and conveys that the information is important.
3. **<i>** - Displays the content in italic format.
4. **<emp>** - Displays the content in italic format and adds emphasis, an alternative to the `<i>*` tag.*
5. ***<u>*** *- Underlines the content.*
6. ***<ins>*** *- Underlines the content, an alternative to the `<u>` tag, often used to indicate inserted text.*
7. ***<strike>*** *- Strikes off (crosses out) the content.*
8. **<del>** - Strikes off the content, an alternative to the `<del>` tag, often used to indicate deleted text.
9. **<q>** - Provides quotations around the content.
10. **<mark>** - Highlights the content with a yellow background color.
11. **<big>** - Displays the content with a larger font size.
12. **<small>** - Displays the content with a smaller font size.
13. **<sup>** - Displays the content as superscript, raising it above the baseline.
14. **<sub>** - Displays the content as subscript, lowering it below the baseline.
15. **<code>** - Displays the content in the "monospace" font family, often used for code snippets.
16. **<pre>** - Preserves both spaces and line breaks, displaying content exactly as for preformatted written, often used text.

# LIST

* + HTML lists allow you to group a set of related items together in a structured format.
  + Lists in HTML are categorized into three types:

1. Ordered List
2. Unordered List
3. Description List

### Ordered List:

Used to group related items in a sequential, numbered format. It is often referred to as a Number List.

#### Tags:

**<ol> -** Denotes the start of an ordered list.

**<li> -** Indicates each list item within the ordered list.

**type:** Specifies the type of numbering sequence.

#### Values:

* **1 (default) -** Numeric (1, 2, 3, ...)
* **A -** Uppercase letters (A, B, C, ...)
* **a -** Lowercase letters (a, b, c, ...)
* **I -** Uppercase Roman numerals (I, II, III, ...)
* **i -** Lowercase Roman numerals (i, ii, iii, ...)
* **start:** Specifies the starting number of the list.
* **Values:** - 1 (default) - Any other number to start the list from.
* **reversed:** Reverses the order of the list items.

### Unordered List

Used to display a set of related items without any particular order.

Commonly referred to as a Bulleted List.

#### Tags:

**<ul>** - Indicates the start of an unordered list.

**<li>** - Indicates each list item within the unordered list.

#### Attributes:

**Type :** Specifies the type of bullet symbol.

#### Values:

* disc (default) - Solid circle
* circle - Hollow circle
* square - Solid square
* none - No bullet, just plain list items.

## Description List

Used to display a list of terms and their associated descriptions. This is often used for glossaries, or to define terms and their meanings.

Tags:

**<dl>** : Indicates the start of a description list.

**<dt>** : Represents a description term.

**<dd>** : Represents a description definition, providing more information about the term.

# HTML ELEMENT

* An HTML Element is a combination of an opening tag, content, and a closing tag.

#### Basic Structure:

<tagname> Content goes here </tagname>

#### Classification of HTML Elements:

* 1. Inline Level Elements
  2. Block level Elements
  3. Inline-Block Level Elements.

### Inline Level Elements

* + - * Inline elements are displayed in the same line .
      * Here we cannot assign height and width properties directly to inline elements.

**Ex**: **<b>** , **<i>** , **<span>** , etc

### Block Level Elements

* + - * These elements occupy the entire width of their parent container or viewport, starting on a new line.
      * We can assign height and width properties to block-level elements.

**Ex**: **<h1>** , **<p>** , **<div>** , etc

### Inline-Block Level Elements

These elements are displayed in the same line as neighbouring elements, similar to inline elements but here we can assign height and width properties to inline-block elements, giving them block-like characteristics.

**Ex**: **<img>** , **<button>** , **<input>** , **<select>** , **<textarea>** , etc.

### HTML ATTRIBUTES

* + - * They provide additional information or functionality to HTML elements.
      * Attributes are placed inside the opening tag of an element and typically consist of a name-value pair.

#### Syntax of Attributes :

Attributes are added within the opening tag of an element and follow this format:

<element attribute=”value”> Content </element>

# IMAGE TAG

The <img> tag has several important attributes that define the image's behavior and appearance:

#### Attributes:

* **`src` (Source):** Specifies the path to the image file.
* This path can be a relative URL (based on the location of the HTML file) or an absolute URL (complete path).
* **`alt` (Alternate Text): -** Provides alternative text for the image if it cannot be displayed.
* `**height`:** - Defines the height of the image in pixels or as a percentage of its original size.
* **`width`:** - Defines the width of the image in pixels or as a percentage of its original size. –

#### Example:

<img src=”img/virat.png” alt=”virat kohli” width=”200” height=”100” />

DIV

* + It is block-level container which is used to group html elements.
  + It is useful for structuring your page, even before applying any styles.

### What Happens without CSS?

* + Each div starts on a new line by default.
  + All div will start vertically.
  + There is no spacing, colour or layouts styling unless you added CSS later.

### <a href = “”>

* + **a** = anchor tag (Creates a link)
  + **href** = tells the browser where the link should go.
  + “**url**” = the Destination (can be website , a file , or an ID on the Same Page)

### Semantic Tag:

Semantic tags are HTML elements that clearly describe their meaning both to the browser and to developers.

**Purpose:** The primary purpose of semantic tags is to improve the **structure, readability,** and **accessibility of web pages**, making it easier for search engines, screen readers, and developers to understand the layout and content.

* + Section
  + Article
  + Nav
  + Footer
  + Header

### <header> :

**Definition:** Represents the introductory section or a group of navigation links in a webpage.

**Usage:** Typically contains logo, site name, and navigation elements.

### <nav> :

**Definition**: Represents a section of the page that links to other pages or to parts within the same page.

**Usage**: Usually used for navigation bars or menus.

### <article> :

**Definition**: Represents an independent piece of content that could be independently distributed or reused, such as blog posts, news articles, or user comments.

**Usage**: Used for self-contained, reusable content.

### <section>:

**Definition**: Represents a generic section of a document or application. Used to group related content together.

**Usage**: Used for thematic grouping of content.

### <main>:

**Definition:** Represents the main content of the webpage that is unique to the document, excluding headers, footers, and sidebars.

**Usage:** Used for the central content of the webpage.

### <footer>:

**Definition:** Represents the footer of a section or page, typically containing copyright information, links to privacy policies, or contact details.

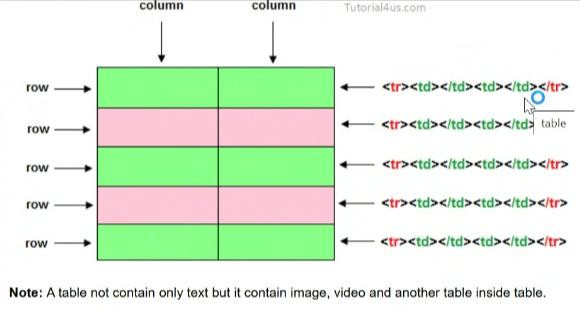
**Usage:** Found at the bottom of the page or section.

## Table

* + HTML table is creating using table element and made up of rows **<tr>**

and cells **<td>** along with header **<th>**.

#### Structure of the Table

****

**Attribute of the Table**

* + **Id :** This attribute are used for provide unique id for table.
  + **border:** Specifies the width of the border around the table and cells. The default value is 0 (no border).
  + **cellpadding:** Specifies the amount of space between the cell content and the cell border.
  + **cellspacing:** Specifies the amount of space between the borders of adjacent cells.
  + **width and height:** Specifies the dimensions of the table. These can be set in pixels or as a percentage of the containing element.

#### Merging Cells

**colspan:** Merges two or more cells within a row.

**rowspan:** Merges two or more rows within a cell.

#### CLASS ATTRIBUTE

* + Class is an attribute which is used to assign one or more labels to an element.
  + Style with css
  + Target with Java Script
  + Group Similar Element

#### What <label> ?

Is used to define a caption or description for an input element

<label for=”username”> User Name:</label>

<input type id=”text” id=”username” name=”username”>

**Form**

* Forms in HTML are used to collect user information.
* Forms are block-level elements, while most of the form elements inside them are inline-level. You have two input attributes.

**Tags Used In Forms:**

1. **<FORM> :**

* It is a semantic element that defines the beginning of the form.
* It contains all the form-related elements.

1. **<Label> :**

* Specifies the purpose of the input field.
* It's linked to the field using the for attribute, which matches the id of the input element.

1. **<input> :**

* Used to collect user data.
* There are various types such as text, email, password, number, etc.

1. **<select> :**

Creates a dropdown list from which users can select options.

1. **<option> :**

Defines the options within a dropdown list.

**6.<Fieldset>**

Groups form elements together and provides a border around them.

**7.<legend>**

Provides a caption for the group**.**

**8.<textarea>**

* Used to collect multi-line text input.
* The rows and cols attributes specify the visible height and width.

**9.<button>**

* Creates a clickable button in the form.
* It can be used to submit or reset the form based on the type attribute.

**Attributes of the <form> tag:**

**1.action:** Specifies the URL where form data will be submitted.

**2.method:** Specifies how the data is sent to the server (GET or POST).

**GET**: Sends data in the URL, not secure.

**POST**: Sends data securely in the HTTP body.

**Attributes of the Tag:**

**1. type:** Specifies the type of input field (e.g., text, email, password, etc.).

**2. id:** Provides a unique identifier for the input element.

**3. name:** Associates a name with the input field, used for form submission.

**4. value:** Defines the initial value of the input field.

**5. required:** Makes the field mandatory to fill before submission.

**6. read only:** Prevents users from modifying the input field value.

**7. disabled:** Disables the input field.

**8. min, max, minlength, maxlength:** Used to define the minimum and maximum values or lengths for input.

**9. placeholder:** Displays a hint text inside the input field.

**10. autofocus:** Automatically focuses on the input field when the page loads.

**11. step:** Specifies the legal number intervals for numeric fields (e.g., step="2" means 2, 4, 6, etc.).

**Values of type Attribute in HTML**

The type attribute of the tag in HTML determines the kind of input field displayed on the webpage. Here are the common values of the type attribute:

**1. text:** Allows users to enter regular text. Default input type.

2. **email**: - Used for email input. It automatically validates the input for an email format.

3. **password**: - Displays input as masked characters (dots or asterisks), hiding the user's input.

4. **number**: - Allows users to enter numerical values. It can have min, max, and step attributes for validation.

5. **tel**: - Allows users to input telephone numbers. It doesn't enforce specific validation but provides a numeric keyboard on mobile devices.

6. **radio**: - Creates a radio button, allowing users to select one option from a group.

7. **checkbox**: - Creates a checkbox, allowing users to select multiple options independently.

8. **button**: - Creates a clickable button. Typically, you can use JavaScript to define what happens when it’s clicked.

9. **submit**: - Creates a submit button that submits the form data to the server.

10. **reset**: - Resets all the form fields to their default values.

11. **search**: - Creates a search input field, often styled with a "clear" button to reset the search query.

12. **date**: - Allows users to select a date using a date picker.

13. **time**: - Allows users to select a time.

14. **week**: - Allows users to select a specific week in a year.

15. **month**: - Allows users to select a specific month and year.

16. **color**: - Allows users to select a color from a color picker.

17. **image**: - Creates a submit button in the form of an image.

18. **file**: - Allows users to upload files from their device.

19. **url**: - Specifies a URL input field. It automatically validates the input for URL format.

20. **range**: - Creates a slider input that allows users to select a numeric value within a specific range. Example:

**Spacing**

**1.&nbsp; 🡪 Non Breaking space**

* Stands for Non-Breaking Space.
* It creates a small space (equal to a normal space character) but prevents line breaks between the words or elements it separates.
* Used when you don’t want the browser to break the line at that space.

**2. &emsp; 🡪 (Em Space)**

* Stands for Em Space.
* It creates a wider space, about four times wider than &nbsp; (equal to the width of the letter "M" in the current font).
* It is mainly used for text formatting, indentation, or spacing large blocks of text.

**HTML Color Format**

* In HTML, **predefined color format** means the different ways we can set or apply colors to text, background, borders, etc.
* HTML supports multiple **color formats** – some are **standard names** (like red, blue) and others use codes (like #ff0000, rgb(255,0,0)).

1. **Color Name (Predefined) :**

HTML already supports **140 color names** which you can directly use.

**Ex:**

<p style="color: red;">This is red text</p>

<p style="background-color: yellow;">Yellow background</p>

**2. Hexadecimal Format :**

**Format:** #RRGGBB

**Ex:**

<p style="color: #ff0000;">Red Text</p>

<p style="color: #00ff00;">Green Text</p>

<p style="color: #0000ff;">Blue Text</p>

3. **RGB Format** (Red, Green, Blue) :

This format uses three numbers from 0 to 255 to represent Red, Green, and Blue.

**Ex:**

<p style="color: rgb(255, 0, 0);">Red</p>

<p style="color: rgb(0, 255, 0);">Green</p>

<p style="color: rgb(0, 0, 255);">Blue</p>

4**. HSL Format (Hue, Saturation, Lightness)**

A color is defined using:

* Hue: angle (0–360°)
* Saturation: % of intensity
* Lightness: % of brightness

**Ex:**

<p style="color: hsl(0, 100%, 50%);">Red</p>

<p style="color: hsl(120, 100%, 50%);">Green</p>

<p style="color: hsl(240, 100%, 50%);">Blue</p>

**Layout**

* HTML layout is a blue print used to arrange webpages in a well defined manner.
* It is used to navigate on the various content on the page footer area in the structured way

**Types of Layout**

1. Flex
2. Grid
3. Float
4. Table
5. Fixed
6. Static
7. Liquid
8. Responsive
9. Adaptive
10. Multi Column

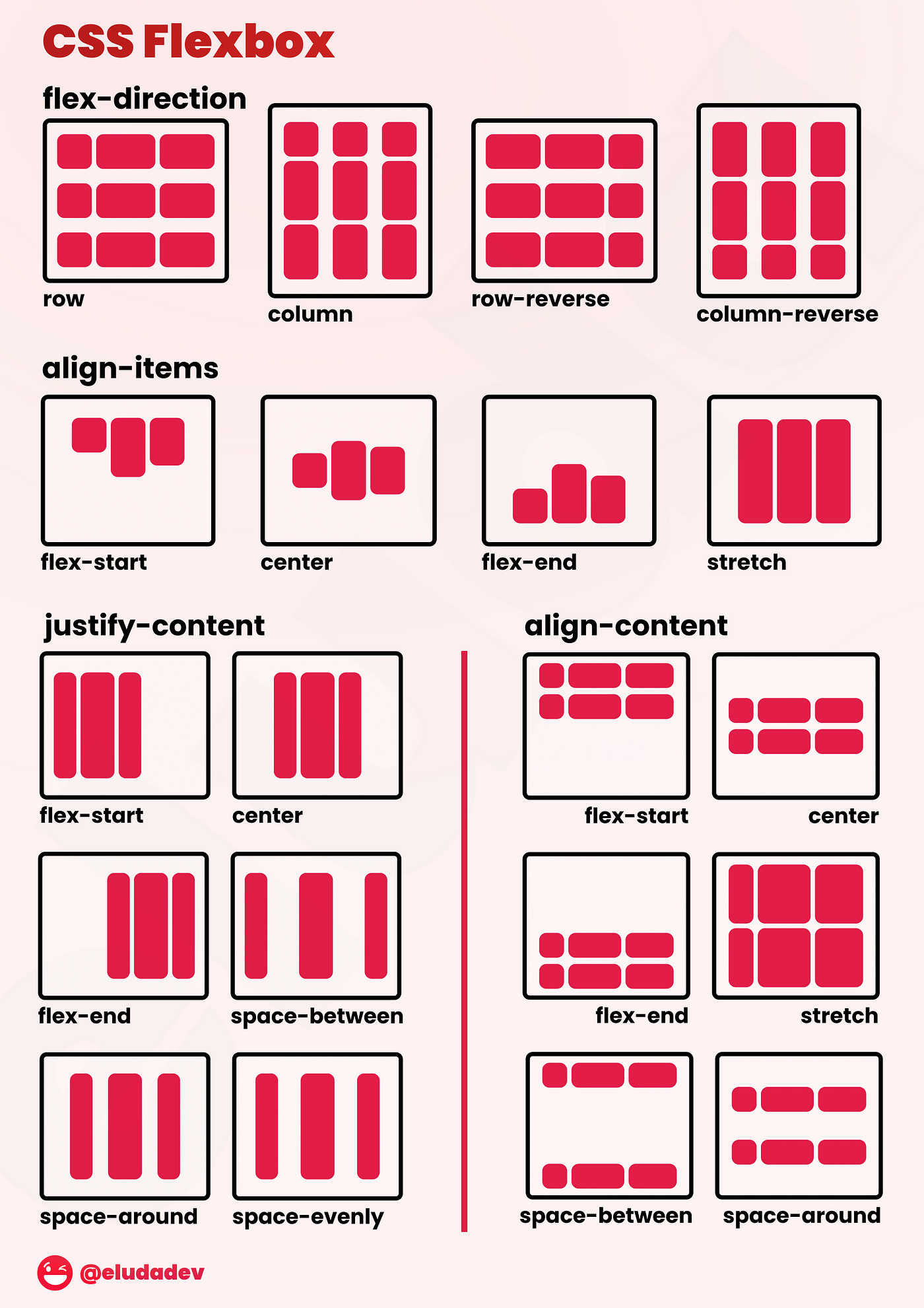
**Responsive Layout**

* Uses Media Query @media to adapt layout based on screen size or device.
* It is most commonly used in mobile based design , system/mobile/laptop/tab.

**Adaptive Layout**

Similar to responsive but uses fixed-width breakpoints with a different fixed-width for layout each.

**1.FLEX Layout**

****