FRONTEND WEB DEVELOPMENT

Unit-1 Notes

UNIT-I:
HTML5: Basics, Lists, Tables, working with links, HTML forms for user Inputs.
Cascading Style sheets: Introduction, Selectors, Fonts and Text Effect, Border and Box Effects, inline, internal and external styling.
Bootstrap: Introduction, Bootstrap components- Helper Classes & Responsive Utilities, Working with Buttons, List, Tables, Forms, Typography, Page Header, Button Groups, navs, modals, grid system.

What is HTML?

- HTML stands for Hypertext Markup Language.

- It is the standard markup language for creating web pages.

- HTML is used to structure content on the web and define the meaning of elements.

Basic Structure of an HTML Document

```
<!DOCTYPE html>
<html>
    <head>
    <meta charset="UTF-8">
    <title>Page Title</title>
    </head>
  <body>
    <h1>Heading</h1>
    <p>Paragraph of text.</p>
  </body>
</html>
```

HTML5 :

HTML5 is the latest version of HTML, introducing new elements and features. It

also emphasize its role in making web content more semantic and accessible.

The structural elements of HTML5:

<header>: Represents the header of a section or a page.

<nav>: Defines a navigation menu.

<section>: Represents a standalone section of content.

<article>: Represents a self-contained piece of content.

<aside>: Defines content that is tangentially related to the content around it.

<footer>: Represents the footer of a section or a page.

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

<head>
    <meta charset="UTF-8">
    <title>HTML5 Structure</title>
</head>

<body>
    <header>
        <h1>My Website</h1>
        <nav>
            <ul>
                <li><a href="/">Home</a></li>
                <li><a href="/about">About</a></li>
            </ul>
        </nav>
    </header>
    <section>
        <h2>About Us</h2>
        <p>Learn about our company and its history.</p>
    </section>
    <article>
        <h2>Latest News</h2>
        <p>Read the latest news and updates.</p>
    </article>
    <aside>
        <h2>Related Links</h2>
        <ul>
            <li><a href="/contact">Contact Us</a></li>
        </ul>
    </aside>
    <footer>
        <p>&copy; 2023 My Website</p>
    </footer>
</body>

</html>
```

127.0.0.1:5500/unit1.html

My Website

- Home
- About

About Us

Learn about our company and its history.

Latest News

Read the latest news and updates.

## Related Links

- Contact Us

© 2023 My Website

HTML4 vs HTML5

HTML4	HTML5
<html>,<body>,<head> tags are mandatory	<html>,<body>,<head> tags can be omitted
Not mobile friendly	Mobile friendly

Doctype declaration is too long
`<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"`
 	
`"http://www.w3.org/TR/html4/strict.dtd">`
	Doctype declaration is simple and easy to use.
	`<!DOCTYPE HTML>`

Cannot handle inaccurate syntax	Capable of handling inaccurate syntax
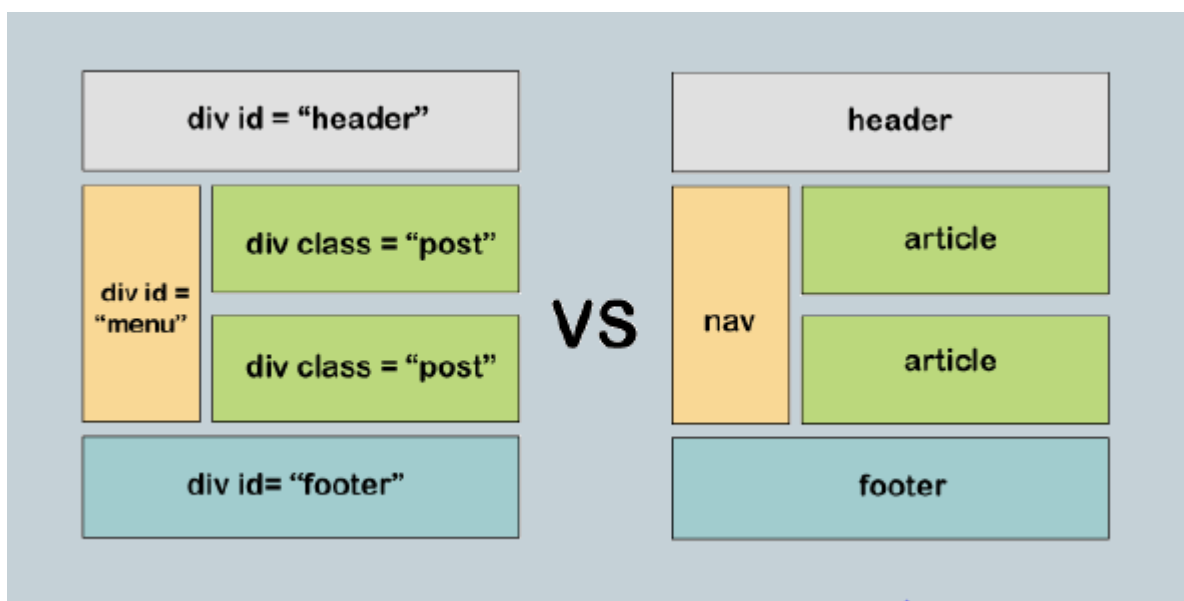
HTML Tags

<html> Tag

The <html> tag is the root element of an HTML document. It encapsulates the entire content of the webpage and indicates that the document is written in HTML.

```
<!DOCTYPE html>
<html>
<!-- Content goes here -->

</html>
```

<head> Tag

The <head> tag contains metadata about the document. This includes information like character encoding, page title, and linked resources (stylesheets, scripts).

Attributes: No common attributes, but often used with <meta>, <title>, and <link> elements.

```
<head>
    <meta charset="UTF-8">
    <title>Page Title</title>
    <link rel="stylesheet" href="styles.css">
</head>
```


`<body>` Tag

The <body> tag contains the visible content of the web page, including text, images, links, and other elements.

Attributes: No common attributes.

```
<body>
    <h1>Welcome to My Website</h1>
    <p>This is a paragraph of text.</p>
</body>
```


Heading Tags: `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`

These tags are used for defining headings of different levels, with <h1> being the highest level and <h6> the lowest.

Attributes: No common attributes.

```
<h1>Main Heading</h1>
<h2>Subheading</h2>
<h3>Sub-subheading</h3>
```
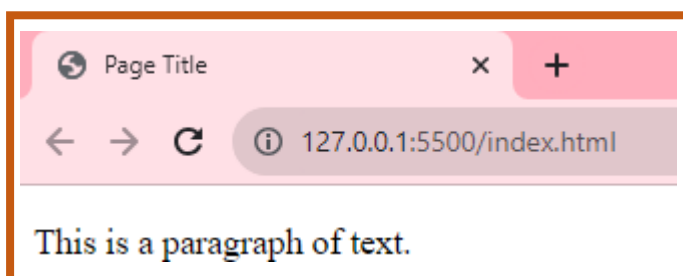

`<p>` Tag

The <p> tag defines a paragraph of text. It is used to separate and format text content into readable blocks.

Attributes: No common attributes.

```
<p>This is a paragraph of text.</p>
```
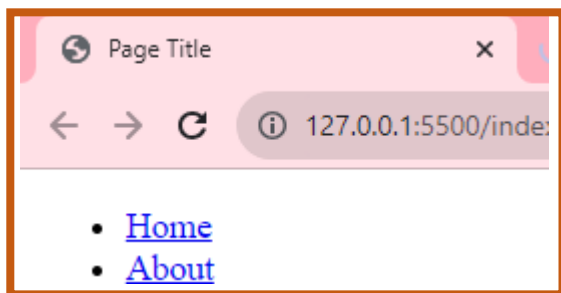

HTML5 Elements

`<nav>` Element

The <nav> element is used to define a section of a webpage that contains navigation links, such as menus or lists of links to other pages.

Attributes: No common attributes, but often used with <ul> and <li> for navigation menus.

```html
<nav>
    <ul>
    <li><a href="/">Home</a></li>
    <li><a href="/about">About</a></li>
    </ul>
</nav>
```
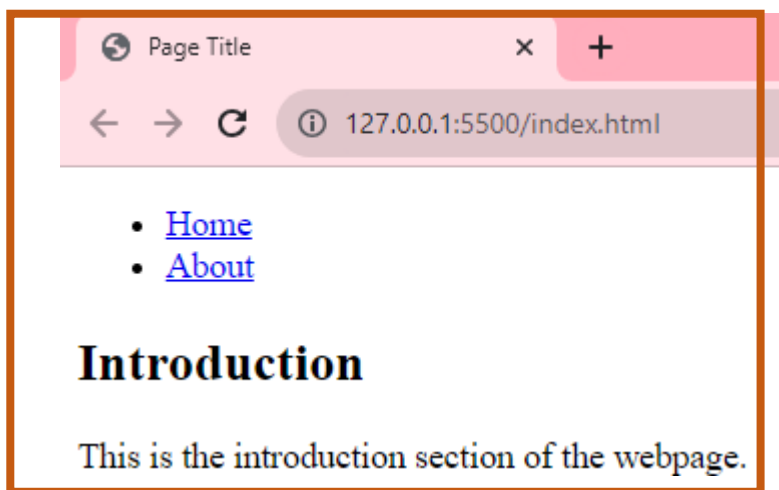
`<section>` Element

The <section> element defines a section of content within an HTML document. It's useful for grouping related content together.

Attributes: No common attributes.

```
<section>
    <h2>Introduction</h2>
    <p>This is the introduction section of the webpage.</p>
</section>
```


`<article>` Element

The <article> element represents a self-contained piece of content, such as a blog post or news article, that can be independently distributed or reused.

Attributes: No common attributes.

```
<article>
    <h3>Blog Post Title</h3>
    <p>Content of the blog post goes here.</p>
</article>
```

`<figure>` and `<figcaption>` Elements

The <figure> element is used to encapsulate media content like images, diagrams, or videos. The <figcaption> element provides a caption or description for the media.

Attributes: No common attributes.

```html
<figure>
    <img src="image.jpg" alt="A beautiful landscape">
    <figcaption>Caption for the image.</figcaption>
</figure>
```

Lists:

To organize and structure content on web pages, there are three types of lists.

1. Unordered lists
2. Ordered lists
3. Description lists

Attributes (common for <ul> and <ol>):

type: Specifies the type of list marker (e.g., disc, circle, decimal, etc.). start: Defines

the starting value for an ordered list.

Unordered lists:

Unordered lists are used when the order of the items is not important. Each item in an unordered list is typically preceded by a bullet point.

Example:

```html
<ul type="square">
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
</ul>
```

- Item 1
- Item 2
- Item 3

Ordered lists:

Ordered lists are used when the order of the items is important. Each item in an ordered list is preceded by a number or another kind of marker.

Example:

```html
<ol>
    <li>First item</li>
    <li>Second item</li>
    <li>Third item</li>
</ol>
```

Output:

1. First item
2. Second item
3. Third item

Description lists:

Description lists are used for term-description pairs. Each term is placed in a <dt> tag and each description is placed in a <dd> tag.

Example:

```
<dl>
    <dt>HTML</dt>
    <dd>HyperText Markup Language</dd>
    <dt>CSS</dt>
    <dd>Cascading Style Sheets</dd>
</dl>
```

Output:

HTML
    HyperText Markup Language
CSS
    Cascading Style Sheets

Tables in HTML:

Tables are used to display data in a structured grid format consisting of rows and columns.

```
They are created using the<table> tag content, and formatting of the table.
To create tables in HTML:element along with several other elements to
define the structure.
```

Structure (<table>): The <table> element is used to define a table.

Table Row (<tr>): Each row in the table is defined using the <tr> element.

Table Header (<th>): Table headers are defined using the <th> element. They typically appear at the beginning of each row or column and are commonly used to represent column headers. By default, text in <th> elements is bold and centered.

Table Data (<td>): Table data cells are defined using the <td> element. They contain the actual data or content of the table.

Here's an example of a simple HTML table with two rows and two columns:

```
<table>
    <tr>
        <th>Heading 1</th>
        <th>Heading 2</th>
    </tr>
    <tr>
        <td>Data 1</td>
        <td>Data 2</td>
    </tr>
</table>
```

Table Attributes:

border: Specifies the width of the border around the table. This attribute is deprecated in HTML5, and it's recommended to use CSS for styling instead.

Example:

```
<table border="1">
```

cellspacing and cellpadding: These attributes control the spacing between cells (cellspacing) and the padding within cells (cellpadding), respectively.

Example:

```
<table cellspacing="5" cellpadding="10">
```

width and height: These attributes set the width and height of the table, respectively. They can be specified in pixels or as a percentage of the available space.

Example:

```
<table width="100%" height="200">
```

align: Specifies the horizontal alignment of the table relative to its containing element. Common values include left, center, and right.

Example:

```
<table align="center">
```

Links in HTML:

Links are used to navigate between different web pages or to specific sections within the same page.

They are created using the <a> (anchor) element in HTML. The basic structure of

a link in HTML is:

```
<a href="URL">Link Text</a>
```

href: Specifies the URL of the page the link goes to. It can be an absolute URL (starting with http:// or https://) or a relative URL (relative to the current page).

Link Text: The text that appears on the page and serves as the clickable link.

Example of a link:

```
<a href="https://www.google.com">Visit Example</a>
```
Additional Link

Attributes:

target: Specifies where to open the linked document. Common values include _blank (opens the linked document in a new tab/window) and _self (opens the linked document in the same frame or tab as the current document).

Example:

```
<a href="https://www. google.com" target="_blank">Visit Example</a>
```

rel: Specifies the relationship between the current document and the linked document. Common values include nofollow, noopener, and noreferrer, which affect search engine behavior and security.

Example:

```
<a href="https://www. google.com" rel="nofollow">Visit Example</a>
```

Forms in HTML:

Forms provide a way to collect user input and send it to a server for processing.

They allow users to input various types of data such as text, numbers, selections, checkboxes, and more.
Here's a basic overview of how to create a form in HTML:

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sample Form</title>
</head>

<body>
    <h2>Sample Form</h2>
    <form action="/submit-form" method="post">
        <!-- Text Input -->
        <label for="name">Name:</label><br>
        <input type="text" id="name" name="name"><br><br>
        <!-- Email Input -->
        <label for="email">Email:</label><br>
        <input type="email" id="email" name="email"><br><br>
        <!-- Password Input -->
        <label for="password">Password:</label><br>
        <input type="password" id="password" name="password"><br><br>
        <!-- Radio Buttons -->
        <label for="gender">Gender:</label><br>
        <input type="radio" id="male" name="gender" value="male">
        <label for="male">Male</label><br>
        <input type="radio" id="female" name="gender" value="female">
        <label for="female">Female</label><br><br>
        <!-- Checkbox -->
        <input type="checkbox" id="subscribe" name="subscribe">
        <label for="subscribe">Subscribe to newsletter</label><br><br>
        <!-- Dropdown Select -->
        <label for="country">Country:</label><br>
        <select id="country" name="country">
            <option value="USA">United States</option>
            <option value="UK">United Kingdom</option>
            <option value="Canada">Canada</option>
        </select><br><br>
```

```
        <!-- Textarea -->
        <label for="message">Message:</label><br>
        <textarea id="message" name="message" rows="4"
cols="50"></textarea><br><br>
        <!-- Submit Button -->
        <input type="submit" value="Submit">
    </form>
</body>

</html>
```

The key components of this form:

<form>: Defines the start and end of the form. It contains all the input elements.
action: Specifies the URL where the form data should be sent upon submission.

method: Specifies the HTTP method to be used when submitting the form (e.g., "post" or "get").
Within the <form> element, various input elements are used to collect different types of data:

<input type="text">: For single-line text input.

<input type="email">: For email input, with built-in validation.

<input type="password">: For password input, with obscured characters.

<input type="radio">: For selecting one option from multiple choices.

<input type="checkbox">: For selecting multiple options from a list.

<select>: Creates a dropdown list from which users can select an option.

<textarea>: For multi-line text input.

<input type="submit">: Creates a submit button to send the form data.

Each input element has attributes like id, name, and value that are used to uniquely identify and process the data.
 Additionally, labels (<label>) are provided for each input element to improve accessibility and user experience.

When the form is submitted, the data is sent to the server specified in the action attribute, using the HTTP method specified in the method attribute (usually "post" for sending data securely). The server then processes the data and responds accordingly.

CSS

CSS stands for Cascading Style Sheets. It's a style sheet language used to describe the presentation of a document written in HTML or XML, including colors, layouts, and fonts.

Importance of CSS:

1. **Separation of Concerns:** CSS separates the structure of a document (HTML) from its presentation, enhancing maintainability and making it easier to update styles across multiple pages.

2. **Consistency:** CSS allows you to apply consistent styling across a website, ensuring uniformity in design and layout.

3. **Flexibility:** CSS provides a high degree of control over the visual appearance of web pages, enabling developers to create custom designs and responsive layouts.

4. **Accessibility:** Properly structured and styled HTML with CSS enhances accessibility by providing clear visual cues and readable content.

CSS Selectors:

CSS selectors are patterns used to select and style elements in an HTML document. They allow you to target specific elements or groups of elements to apply styles. Here are some common CSS selectors:

1. **Element Selector**: Targets HTML elements by their tag name.
   Example:

```
p {
color: blue;
}
```

This example targets all <p> elements and sets their text color to blue.

2. **ID Selector:** Targets elements with a specific ID attribute.

Example:

```
#header {
font-size: 24px;
}
```

This targets an element with the ID "header" and sets its font size to 24 pixels.

3. **Class Selector**: Targets elements with a specific class attribute.
Example:

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

This targets all elements with the class "text-center" and sets their text alignment to center.

4. **Attribute Selector**: Targets elements based on their attribute values

Example:
```
input[type="text"] {
border : 1px solid #ccc;
}
```

This targets all <input> elements with the attribute type="text" and sets their border to a 1-pixel solid color #ccc.

5. **Descendant Selector:** Targets elements that are descendants of a specific parent.
Example:
```
ul li {
    list-style-type: square;
}
```

This targets all <li> elements that are descendants of <ul> elements and sets their list style to square.

6. **Child Selector: Targets elements that are direct children of a specific parent.**
Example:
```
ul>li{

font-weight: bold;

}
```

This targets all <li> elements that are direct children of <ul> elements and sets their font weight to bold.

7. **Universal Selector:** Targets all elements in the document.

Example:
```
* {
 margin:0;
 padding:1px;
}
```

This targets all elements and sets their margin and padding to zero.

8.   **Pseudo-classes and Pseudo-elements:** Targets elements based on their state or position in the document.

Example:

```
a:hover {
    : red;
}

p::first-line {
font-weight: bold;
}
```

This targets hyperlinks (<a>) when hovered over and changes their color to red. It also targets the first line of every <p> element and sets its font weight to bold.

These are just some of the basic CSS selectors. CSS offers many more selectors and features for styling elements, allowing developers to create visually appealing and responsive web pages.

CSS Properties:

Fonts:

1. **font-family**: Specifies the font family for text. It can be a specific font name or a generic font family.

   Example:

   ```
   body{

   font-family: Helvetica, sans-serif;

   }
   ```

2. **font-size**: Sets the size of the font.
   Example:

   ```
   h1{
   Font-size:36px;
   }
   ```

3. **font-weight:** Sets the weight (boldness) of the font.

   Example:

   ```
   p{

   font-weight:bold;

   }
   ```

4. **font-style:** Sets the style (italic or normal) of the font.
   Example:

```
em{
fong-sytle:itealic;
}
```

5. **font-variant:** Controls the usage of small caps for lowercase characters.

   Example:

```
p{

font-variant:small-caps;

}
```

Text Effects:

1. **color:** Sets the color of text.
   Example:

```
.primary-text {
    color:  ;
}
```

2. **text-shadow:** Adds a shadow to text.

Example:

```
h2 {
    : 2px 2px 4px rgba(0, 0, 0, 0.5);
}
```

3. **text-decoration**: Adds decorations like underline, overline, or line-through to text.

Example:

```
underline-text {
    text-decoration: underline;
}

.line-through-text {
        : line-through;
}
```

letter-spacing: Sets the spacing between characters.

```
.spaced-text {
    letter-spacing: 2px;
}
```

4. **word-spacing:** Sets the spacing between words.

Example:

```
.wide-spacing {
    word-spacing: 5px;
}
```

Borders:

1. **border:** Shorthand property for setting the border width, style, and color.

    Example:

```
.bordered-element {
    border: 1px solid #000;
    }
```

2. **border-width:** Sets the width of the border.

Example:

```
.thick-border {
    border-width: 3px;
    }
```

3. **border-style:** Sets the style of the border (e.g., solid, dashed, dotted).

Example:

```
.dashed-border {
    border-style: dashed;
}
```

4. **border-color:** Sets the color of the border.

Example:

```
.red-border {
```

```
    border-color: red;
}
```

5. **border-radius:** Rounds the corners of the border.

Example:
```
.rounded-border {
    border-radius: 10px;
    }
```

These CSS properties provide extensive control over the appearance of fonts, text effects, and borders in your web pages. By leveraging these properties effectively, you can create visually appealing designs and enhance the user experience of your website.

What is Bootstrap?

Bootstrap is a widely-used, open-source front-end framework for building responsive and visually appealing web applications and websites.

Key Features:

Responsive Design:

Bootstrap follows a mobile-first approach, ensuring your web content looks great and functions well on a variety of devices, from mobile phones to desktop computers.

Grid System:

Bootstrap includes a powerful grid system that allows you to create flexible and responsive layouts easily. It's based on a 12-column grid, making it simple to structure your content.

Pre-Designed Components:

Bootstrap provides a rich library of pre-designed UI components such as buttons, forms, navigation bars, and more. These components are customizable and help streamline development.

CSS Reset:

Bootstrap includes a CSS reset, known as "Reboot," which ensures a consistent baseline style across different browsers, helping to avoid cross-browser compatibility issues.

Customization:

You can customize Bootstrap to match your project's branding by modifying its default variables, such as colors, typography, and spacing.

How to Get Started:

Download Bootstrap:

You can download the Bootstrap framework from the official website

(https://getbootstrap.com/docs/5.0/getting-started/introduction/) or include it via a content delivery network (CDN).

Include Bootstrap in Your Project:

Add Bootstrap's CSS and JavaScript files to your HTML documents using <link> and <script> tags.

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.
min.c ss"  rel="stylesheet"  integrity="sha384-
4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+PtmoHDEXuppvnDJzQIu9"
crossorigin="anonymous">
```

Benefits of Using Bootstrap:

Rapid Development: Bootstrap simplifies and accelerates the web development process by providing ready-made components and a responsive grid system.

Consistency: Bootstrap promotes a consistent look and feel across your project, making it easier to maintain a unified design.

Cross-Browser Compatibility: Bootstrap handles many cross-browser issues, reducing the need for extensive testing and adjustments.
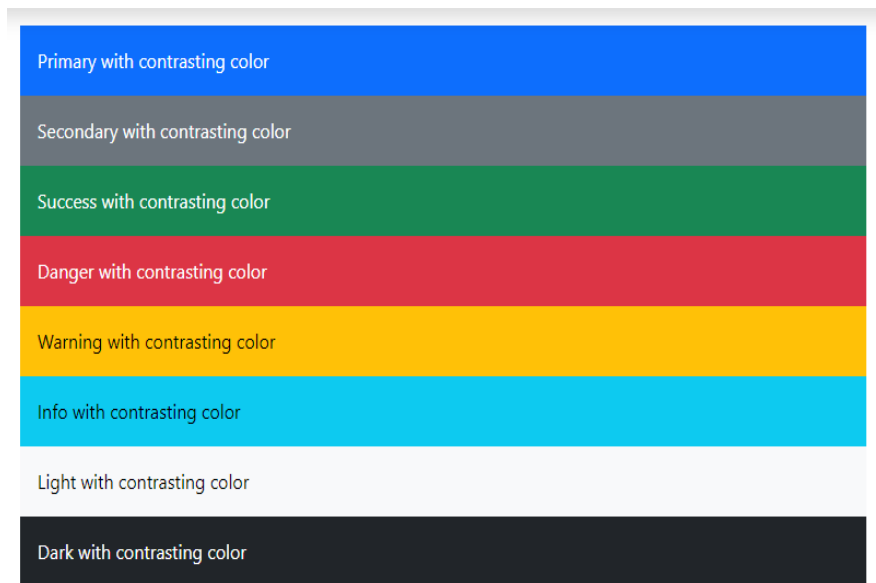
Community and Documentation: Bootstrap has a large and active community, along with comprehensive documentation and numerous online resources for support.

Using Bootstrap Helper classes and responsive utilities

1. **Color and background**

Set a background color with contrasting foreground color. Color and background helpers combine the power of our .text-* utilities and .bg-* utilities in one class.

```
<div class="text-bg-primary p-3">Primary with contrasting color</div>
<div class="text-bg-secondary p-3">Secondary with contrasting
color</div>
<div class="text-bg-success p-3">Success with contrasting color</div>
<div class="text-bg-danger p-3">Danger with contrasting color</div>
<div class="text-bg-warning p-3">Warning with contrasting color</div>
<div class="text-bg-info p-3">Info with contrasting color</div>
<div class="text-bg-light p-3">Light with contrasting color</div>
<div class="text-bg-dark p-3">Dark with contrasting color</div>
```

Primary with contrasting color

Secondary with contrasting color

Success with contrasting color

Danger with contrasting color

Warning with contrasting color

Info with contrasting color

Light with contrasting color

Dark with contrasting color

2. Position

Use these helpers for quickly configuring the position of an element.

```
<div class="fixed-top">...</div>
```

Fixed top

Position an element at the top of the viewport, from edge to edge. Be sure you understand the ramifications of fixed position in your project; you may need to add additional CSS.Fixed bottom Position an element at the bottom of the viewport, from edge to edge. Be sure you understand the ramifications of fixed position in your project; you may need to add additional CSS.

```
<div class="fixed-bottom">...</div>
```

Sticky top

Position an element at the top of the viewport, from edge to edge, but only after you scroll past it.

```
<div class="sticky-top">...</div>
```

Sticky bottom

Position an element at the bottom of the viewport, from edge to edge, but only after you scroll past it.

```
<div class="sticky-bottom">...</div>
```
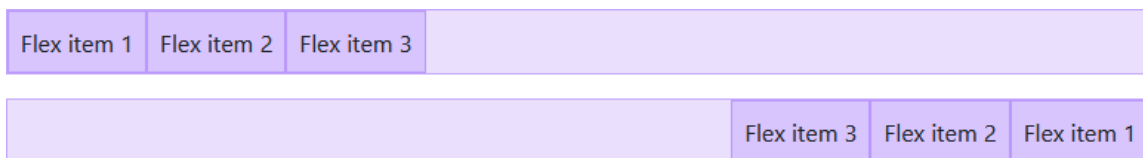
3. Colors

Colorize text with color utilities. If you want to colorize links, you can use the .link-* helper classes which have :hover and :focus states.

.text-primary	`<p class="text-primary">.text-primary</p>`
.text-primary-emphasis	`<p class="text-primary-emphasis">.text-primary-emphasis</p>`
.text-secondary	`<p class="text-secondary">.text-secondary</p>`
.text-secondary-emphasis	`<p class="text-secondary-emphasis">.text-secondary-emphasis</p>`
.text-success	`<p class="text-success">.text-success</p>`
.text-success-emphasis	`<p class="text-success-emphasis">.text-success-emphasis</p>`
.text-danger	`<p class="text-danger">.text-danger</p>`
.text-danger-emphasis	`<p class="text-danger-emphasis">.text-danger-emphasis</p>`
.text-warning	`<p class="text-warning bg-dark">.text-warning</p>`
.text-warning-emphasis	`<p class="text-warning-emphasis">.text-warning-emphasis</p>`
.text-info	`<p class="text-info bg-dark">.text-info</p>`
.text-info-emphasis	`<p class="text-info-emphasis">.text-info-emphasis</p>`
.text-light	`<p class="text-light bg-dark">.text-light</p>`
.text-light-emphasis	`<p class="text-light-emphasis">.text-light-emphasis</p>`
.text-dark	`<p class="text-dark bg-white">.text-dark</p>`
.text-dark-emphasis	`<p class="text-dark-emphasis">.text-dark-emphasis</p>`
.text-body	`<p class="text-body">.text-body</p>`
.text-body-emphasis	`<p class="text-body-emphasis">.text-body-emphasis</p>`
.text-body-secondary	`<p class="text-body-secondary">.text-body-secondary</p>`
.text-body-tertiary	`<p class="text-body-tertiary">.text-body-tertiary</p>`
.text-black	`<p class="text-black bg-white">.text-black</p>`
.text-white	`<p class="text-white bg-dark">.text-white</p>`
.text-black-50	`<p class="text-black-50 bg-white">.text-black-50</p>`
.text-white-50	`<p class="text-white-50 bg-dark">.text-white-50</p>`

4. Flex

To quickly manage the layout, alignment, and sizing of grid columns, navigation, components, and more with a full suite of responsive flexbox utilities. For more complex implementations, custom CSS may be necessary.

```html
<div class="d-flex flex-row mb-3">
    <div class="p-2">Flex item 1</div>
    <div class="p-2">Flex item 2</div>
    <div class="p-2">Flex item 3</div>
</div>
<div class="d-flex flex-row-reverse">
    <div class="p-2">Flex item 1</div>
    <div class="p-2">Flex item 2</div>
    <div class="p-2">Flex item 3</div>
</div>
```

Use `.flex-column` to set a vertical direction, or `.flex-column-reverse` to start the vertical direction from the opposite side.

```html
<div class="d-flex flex-column mb-3">
    <div class="p-2">Flex item 1</div>
    <div class="p-2">Flex item 2</div>
    <div class="p-2">Flex item 3</div>
</div>
<div class="d-flex flex-column-reverse">
    <div class="p-2">Flex item 1</div>
    <div class="p-2">Flex item 2</div>
    <div class="p-2">Flex item 3</div>
</div>
```

5. Navbar

- Navbars require a wrapping .navbar with .navbar-expand{-sm|-md|-lg|-xl|-xxl} for responsive collapsing and color scheme classes.
- Navbars and their contents are fluid by default. Change the container to limit their horizontal width in different ways.
- Use our spacing and flex utility classes for controlling spacing and alignment within navbars.
- Navbars are responsive by default, but you can easily modify them to change that. Responsive behavior depends on our Collapse JavaScript plugin.

| Flex item 1 |
| Flex item 2 |
| Flex item 3 |

| Flex item 3 |
| Flex item 2 |
| Flex item 1 |

Supported content

Navbars come with built-in support for a handful of sub-components. Choose from the following as needed:

- .navbar-brand for your company, product, or project name.
- .navbar-nav for a full-height and lightweight navigation (including support for dropdowns).
- .navbar-toggler for use with our collapse plugin and other navigation toggling behaviors.
- Flex and spacing utilities for any form controls and actions.
- .navbar-text for adding vertically centered strings of text.
- .collapse.navbar-collapse for grouping and hiding navbar contents by a parent breakpoint.
- Add an optional .navbar-scroll to set a max-height and scroll expanded navbar content.

```
<nav class="navbar navbar-expand-lg bg-body-tertiary">
    <div class="container-fluid">
        <a class="navbar-brand" href="#">Navbar</a>
        <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarSupportedContent"
            aria- controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse"
id="navbarSupportedContent">
            <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                <li class="nav-item">
                    <a class="nav-link active" aria-current="page"
href="#">Home</a>
```

```html
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#">Link</a>
                    </li>
                    <li class="nav-item dropdown">
                        <a class="nav-link dropdown-toggle" href="#"
role="button" data- bs-toggle="dropdown"
                            aria-expanded="false">
                            Dropdown
                        </a>
                        <ul class="dropdown-menu">
                            <li><a class="dropdown-item"
href="#">Action</a></li>
                                <li><a class="dropdown-item" href="#">Another
action</a></li>
                            <li>
                                <hr class="dropdown-divider">
                            </li>
                            <li><a class="dropdown-item"
href="#">Something else here</a></li>
                        </ul>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link disabled" aria-
disabled="true">Disabled</a>
                    </li>
                </ul>
            <form class="d-flex" role="search">
                <input class="form-control me-2" type="search"
placeholder="Search" aria-label="Search">
                <button class="btn  btn-outline-success"
type="submit">Search</button>
            </form>
        </div>
    </div>

</nav>
```

Typography

Display headings

Traditional heading elements are designed to work best in the meat of your page content. When you need a heading to stand out, consider using a **display heading**—a larger, slightly more opinionated heading style.

```html
<h1 class="display-1">Display 1</h1>
<h1 class="display-2">Display 2</h1>
<h1 class="display-3">Display 3</h1>
<h1 class="display-4">Display 4</h1>
<h1 class="display-5">Display 5</h1>
<h1 class="display-6">Display 6</h1>
```

Display 1

Display 2

Display 3

Display 4

Display 5

Display 6

Do visit for more details on typography https://getbootstrap.com/docs/5.3/content/typography/
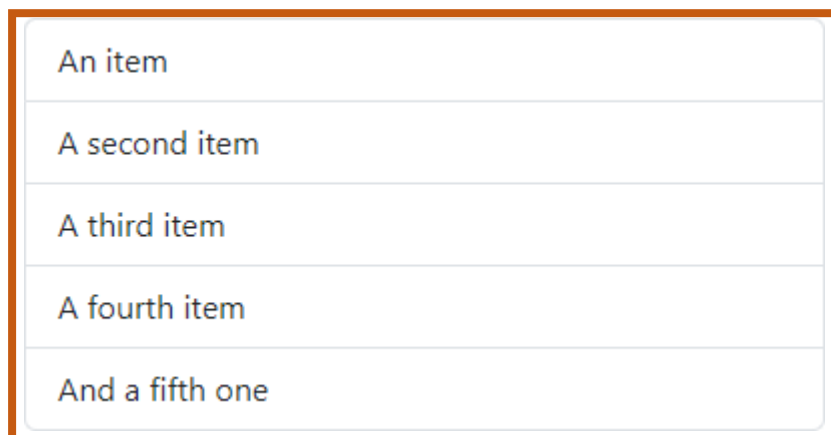
List group

List groups are a flexible and powerful component for displaying a series of content. Modify and extend them to support just about any content within.

Basic example

The most basic list group is an unordered list with list items and the proper classes. Build upon it with the options that follow, or with your own CSS as needed.

```
<ul class="list-group">
    <li class="list-group-item">An item</li>
    <li class="list-group-item">A second item</li>
    <li class="list-group-item">A third item</li>
    <li class="list-group-item">A fourth item</li>
    <li class="list-group-item">And a fifth one</li>
</ul>
```

An item

A second item

A third item

A fourth item

And a fifth one

Do visit for more details on typography https://getbootstrap.com/docs/5.3/components/list-group/

Grid system

Use grid system for powerful mobile-first flexbox grid to build layouts of all shapes and sizes thanks to a twelve column system, six default responsive tiers, mixins, and dozens of predefined classes.

Example

Bootstrap's grid system uses a series of containers, rows, and columns to layout and align content. It's built with flexbox and is fully responsive. Below is an example and an in-depth explanation for how the grid system comes together.

```
<div class="container text-center">
    <div class="row">
        <div class="col"> Column
        </div>
        <div class="col"> Column
        </div>
        <div class="col"> Column
```
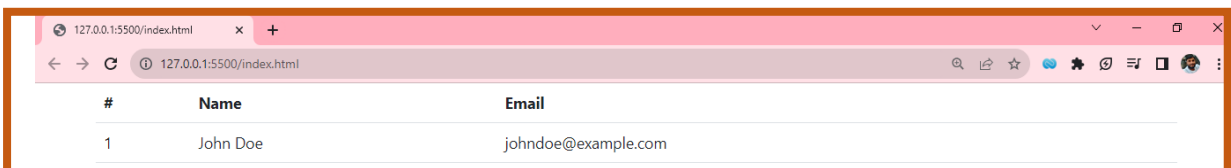
```
        </div>
    </div>
</div>
```


Do visit for more details on typography https://getbootstrap.com/docs/5.3/layout/grid/

Tables

Bootstrap provides classes to style HTML tables, making them more visually appealing.

```html
<table class="table">
    <thead>
        <tr>
            <th>#</th>
            <th>Name</th>
            <th>Email</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>1</td>
            <td>John Doe</td>
            <td>johndoe@example.com</td>
        </tr>
        <!-- More rows -->
    </tbody>
</table>
```

The table class applies Bootstrap styling to the table.

Refer : https://getbootstrap.com/docs/5.3/content/tables/

Forms

Form Controls

Bootstrap 5 offers a wide range of form control styles and components to help streamline the process of creating forms in web development.

Basic Form Controls:

Bootstrap provides styles for various basic form controls such as input fields, checkboxes, radio buttons, textareas, and selects.

```html
<form>
    <div class="mb-3">
        <label for="exampleFormControlInput1" class="form-label">Email
address</label>
        <input type="email" class="form-control"
id="exampleFormControlInput1" placeholder="name@example.com">
    </div>
    <div class="mb-3">
        <label for="exampleFormControlTextarea1" class="form-
label">Example textarea</label>
        <textarea class="form-control"
id="exampleFormControlTextarea1" rows="3"></textarea>
    </div>
    <div class="mb-3">
        <label for="exampleFormControlSelect1" class="form-
label">Example select</label>
        <select class="form-select" id="exampleFormControlSelect1">
            <option>1</option>
            <option>2</option>
            <option>3</option>
            <option>4</option>
            <option>5</option>
        </select>
    </div>
    <div class="mb-3 form-check">
        <input type="checkbox" class="form-check-input"
id="exampleCheck1">
        <label class="form-check-label" for="exampleCheck1">Check me
out</label>
    </div>
    <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Validation States:

Bootstrap provides styles for form controls with different validation states like valid and invalid.

```html
<div class="mb-3">
    <label for="validationExample" class="form-label">Example
select</label>
    <select class="form-select" id="validationExample" required>
        <option value="">Open this select menu</option>
        <option value="1">One</option>
        <option value="2">Two</option>
        <option value="3">Three</option>
    </select>
    <div class="invalid-feedback"> Please select a valid option.
    </div>
</div>
```
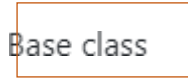
Do visit for more details on form controls https://getbootstrap.com/docs/5.3/forms/form-control/

Buttons

Base class

Bootstrap has a base .btn class that sets up basic styles such as padding and content alignment. By default, .btn controls have a transparent border and background color, and lack any explicit focus and hover styles.

```
<button type="button" class="btn">Base class</button>
```


Variants

Bootstrap includes several button variants, each serving its own semantic purpose, with a few extras thrown in for more control.

```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

<button type="button" class="btn btn-link">Link</button>
```


Do visit for more details on buttons https://getbootstrap.com/docs/5.3/components/buttons/

Button group

Group a series of buttons together on a single line or stack them in a vertical column.

Basic example

Wrap a series of buttons with .btn in .btn-group.

```
<div class="btn-group" role="group" aria-label="Basic mixed styles
example">
    <button type="button" class="btn btn-danger">Left</button>
    <button type="button" class="btn btn-warning">Middle</button>
    <button type="button" class="btn btn-success">Right</button>
</div>
```

Output:

Do visit for more details on buttongroups
https://getbootstrap.com/docs/5.3/components/button- group/

Modal

Use Bootstrap's JavaScript modal plugin to add dialogs to your site for lightboxes, user notifications, or completely custom content.

How it works

Before getting started with Bootstrap's modal component, be sure to read the following as our menu options have recently changed.

- Modals are built with HTML, CSS, and JavaScript. They're positioned over everything else in the document and remove scroll from the **<body>** so that modal content scrolls instead.
- Clicking on the modal "backdrop" will automatically close the modal.
- Bootstrap only supports one modal window at a time. Nested modals aren't supported as we believe them to be poor user experiences.
- Modals use **position: fixed**, which can sometimes be a bit particular about its rendering. Whenever possible, place your modal HTML in a top-level position to avoid potential interference from other elements. You'll likely run into issues when nesting a **.modal** within another fixed element.
- Once again, due to position: fixed, there are some caveats with using modals on mobile devices.
- Due to how HTML5 defines its semantics, the autofocus HTML attribute has no effect in Bootstrap modals. To achieve the same effect, use some custom JavaScript:

```
const myModal = document.getElementById('myModal') const myInput
= document.getElementById('myInput')

myModal.addEventListener('shown.bs.modal', () => {
myInput.focus()

})
```

Below is a *static* modal example (meaning its position and display have been overridden). Included are the modal header, modal body (required for padding), and modal footer (optional). We ask that you include modal headers with dismiss actions whenever possible, or provide another explicit dismiss action.

```html
<div class="modal" tabindex="-1">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Modal title</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
                <p>Modal body text goes here.</p>
            </div>
            <div class="modal-footer"><button type="button" dismiss="modal">Close</button>

                <button type="button" class="btn btn-secondory" btn-primary">Save changes</button>
            </div>
        </div>
    </div>
</div>
```

Do visit for more details on modals https://getbootstrap.com/docs/5.3/components/modal/