



University of Global Village (UGV),Barishal

Web Page Design And Development (Level-1)

Amartya Kundu Durjoy

Lecturer

Department of CSE

University Of Global Village, Barishal

Course Code

Name of Course Title	Web page design and development (Level-1)
Course Type	Skill Course
Level	1 st Semester
Academic Session	Winter 2025
Name(s) of Academic Course teacher(s)	Amartya Kundu Durjoy, Lecturer,CSE. Mobile: 0176638973 E-mail: amartyakundudurjoy@gmail.com
Practice Hour:	80 hours

**Web page design and development Student
(UGV) Format**

Course Code:--

Credits: 1

Exam Hours:--

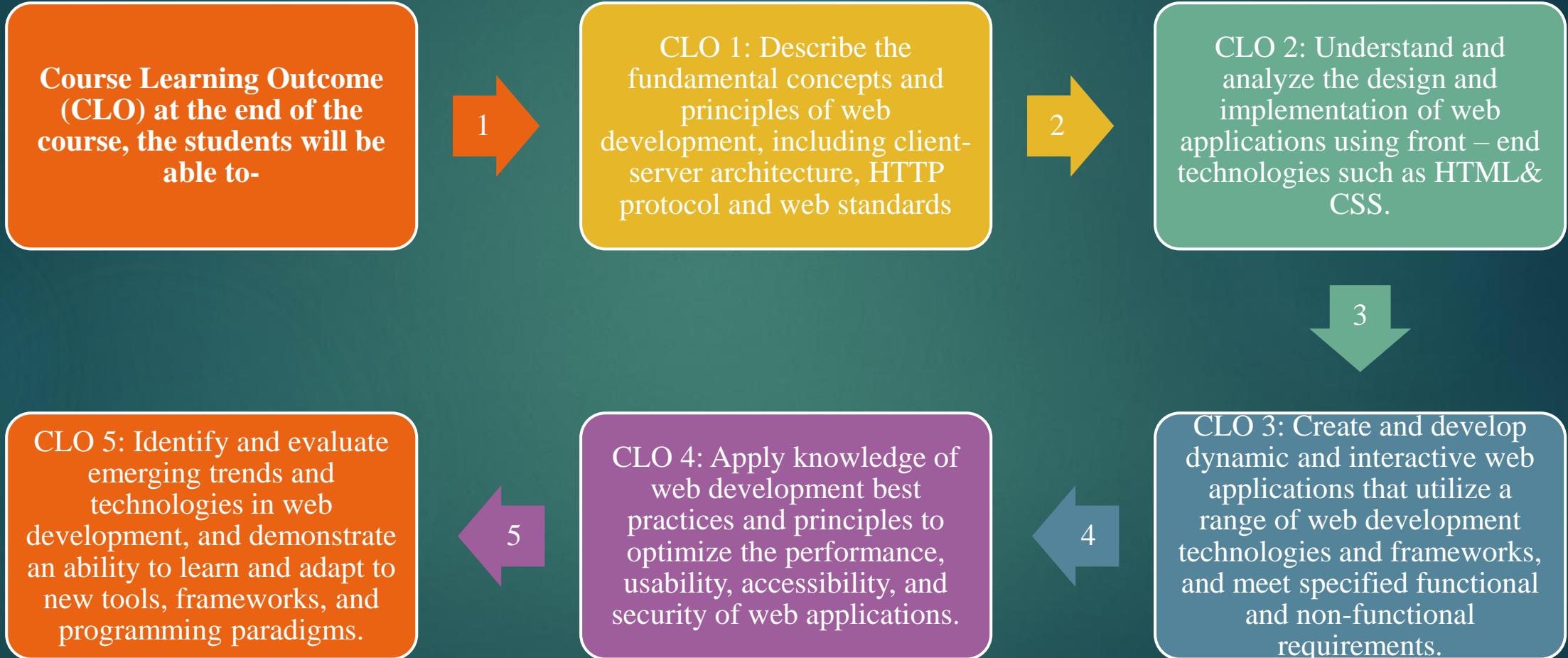
CIEMarks:30

Course for 1stSemester,

SEEMarks:20

**Bachelor of Science in Computer Science
Engineering (CSE)**

CLO



Topics

Week	Topics	Teaching-Learning Strategy(s)	Class Hour	Practice Hour	Assessment Strategy(s)	Mapping with CLO
01	Introduction to Web Development: Understanding the process of web development, from planning, designing, coding, testing, to deployment.	Lecture, Demonstration, Interactive Q&A	5h	2h	Participation, Short Quiz	CLO 1
02,03& 04	HTML Basic Structure, Elements & attributes: Understanding how to design the front end in web design.	Lecture, Hands-on Exercises with HTML	15h	18h	Lab Performance, Practical Exercise	CLO 2
05	HTML Lists and Tables: Understanding how to build nested lists and tables	Lecture, Hands-on Exercises with HTML	5h	4h	Lab Participation, Practical Assignment	CLO 2
06	Introduction to CSS: Understanding css syntax, selectors, properties, and values.	Hands-on sessions with , Coding tutorials	5h	4h	Lab Performance, Coding Test	CLO 2 & CLO 4
07 & 08	CSS box model and Different types of layout: Understanding static, relative, absolute, and fixed positioning, box model, grid view, flexbox	Lecture, Hands-on Exercises with coding tutorials	10h	7h	Hands-on Assignment, Short Quiz	CLO 2 & CLO 4

Topics

Week	Topic Description	Teaching Methodology	Lecture Hours	Practical Hours	Evaluation Method	CLO
09, 10 & 11	CSS advanced: Understanding about the advanced technologies of css such as image filtering, object position, masking, shadows etc.	Lecture, Hands-on Exercises with HTML and CSS	15h	10h	Lab Performance, Coding Test	CLO 4
12 & 13	Bootstrap CSS framework: Understanding how to interactive the web page using css advance technology	Lecture, Hands-on Exercises with HTML and CSS	10h	8h	Lab Performance	CLO 3
14 & 15	Tailwind CSS Mastery: Understanding how to use css framework	Lecture, Hands-on sessions with , Coding tutorials	10h	8h	Lab Performance, Coding Test	CLO 3, CLO 5
16	Project Presentation: Ability to effectively communicate web design project designs and results.	Presentation preparation and peer review	5h	1h	Cloud Integration Exercise, Test on Cloud Tools	CLO 5
17	Final Assessment: Evaluation of knowledge and practical skills in Web development.	Written test, Practical assessment covering all components	5h		Written Exam, Lab Performance Assessment	CLO1 – CLO 5

Week 1

Lecture 1

Introduction to Web Development



WWW

- ▶ The **World Wide Web (WWW)**, commonly known as the Web, is an interconnected system of public webpages accessible via the Internet. It is a service that operates on the Internet, allowing users to access and share information globally using a web browser.

Key Components of the WWW

- **Web Pages:** Documents written in HTML (HyperText Markup Language) and accessed via HTTP (HyperText Transfer Protocol).
- **Web Browsers:** Software applications like Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge used to view web pages.
- **URLs (Uniform Resource Locators):** Addresses used to locate resources (like web pages) on the web.
- **Web Servers:** Computers hosting web pages and delivering them to users upon request.

Working Principle of WWW

- **Client-Server Model:** When you type a URL into a web browser, a request is sent to a web server.
- **HTTP/HTTPS Protocols:** The request and response between the browser and server use HTTP or HTTPS.
- **Rendering Pages:** The server sends the requested web page to your browser, which renders it for display.

Client-Server architecture

CLIENTS COMMUNICATE WITH A SINGLE



Clients communicate with a single server.



Client initiates communication.



Client receives the response and decide whether to continue the communication.



Server awaits connections.



Server receives a request and sends a response.



Example: e-mail, dns, http

A photograph of a wooden pier extending from the bottom center towards a small, blue-painted wooden boathouse with a gabled roof and two windows, situated on stilts in the middle of a body of water. The sky is overcast with a vertical red bar in the top right corner.

Peer to peer Architecture

P2P Networks

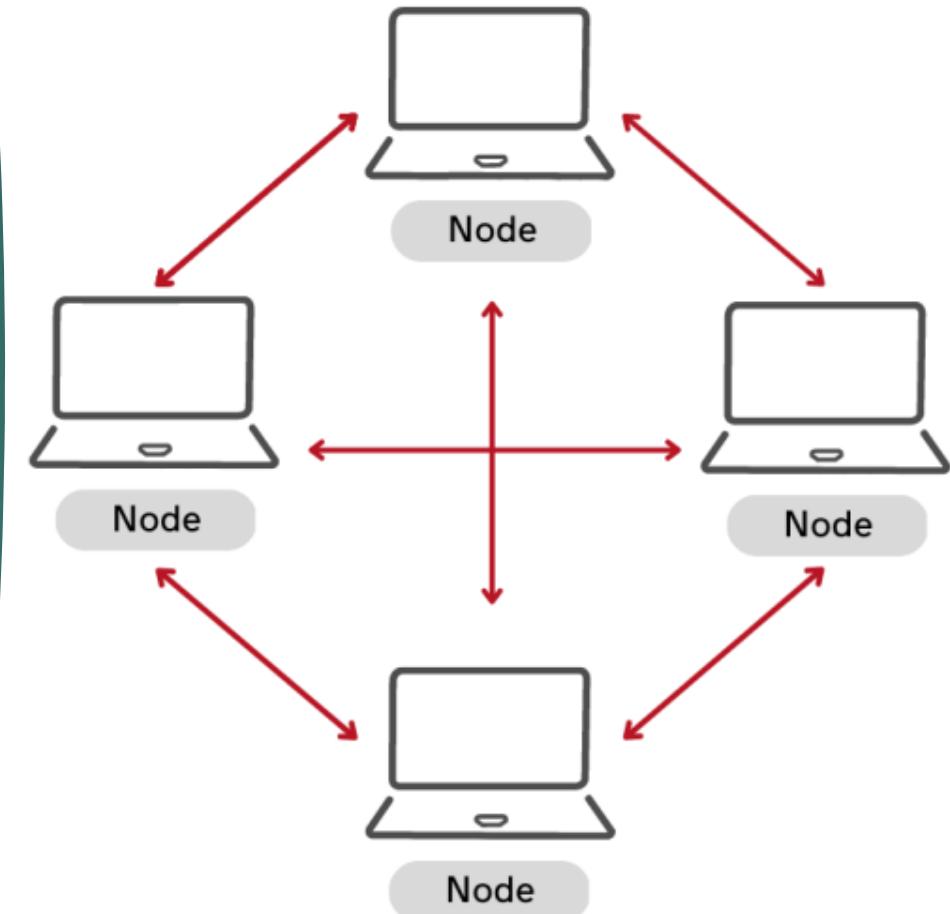
No server in the network

Clients are communicating between them.

No intermediary entities.

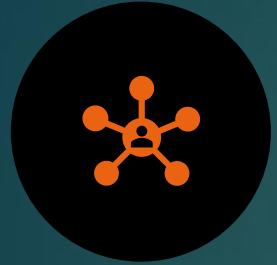
Used primarily in resource sharing.

Example: napster, kazaa, direct connect
hitTorrent.

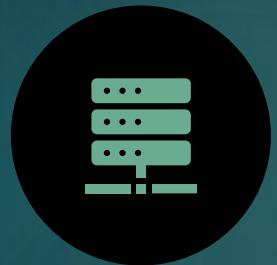


Comparison: Client-Server vs. Peer-to-Peer (P2P) Networks

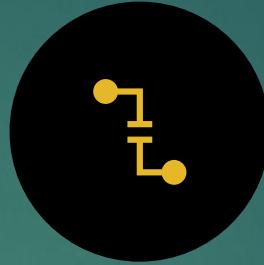




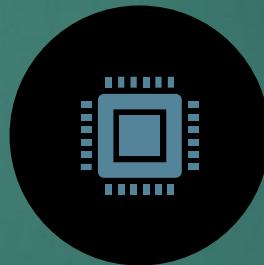
Architecture: Client-server networks rely on a central server, whereas P2P networks are decentralized, with each node acting as both client and server.



Resilience: Client-server networks are vulnerable to server failures, while P2P networks are more resilient since no single node failure disrupts the network.



Scalability: Client-server networks face bottlenecks with increased users, whereas P2P networks scale easily as more nodes join.



Performance: Client-server networks typically offer faster performance due to optimized servers, whereas P2P networks can be slower as resources are shared across nodes.

Student's Task

Explain

Explain the working principle of the World Wide Web (WWW) in your own words.

Create

Create a diagram of the client-server architecture showing how a browser interacts with a web server.

Research and list

Research and list 5 commonly used web browsers with their features.



Thank you

Week 2, 3 & 4

Lecture 2, 3 & 4

HTML Basics - Structure and Elements



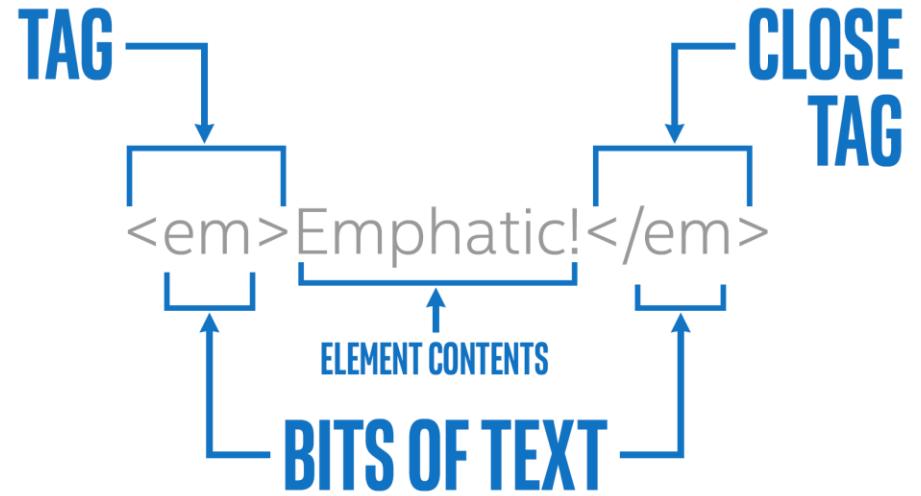
Introduction

HTML (Hypertext Markup Language) is the foundation of web development. It is a markup language used to structure and present content on the web.

Definition: HTML is a language used to describe the structure of web pages using elements and tags.

Markup Language: It uses "tags" to annotate text, images, and other content to be displayed in web browsers.

Hypertext: Refers to text that contains links to other documents or web resources.



Purpose of HTML

Organize

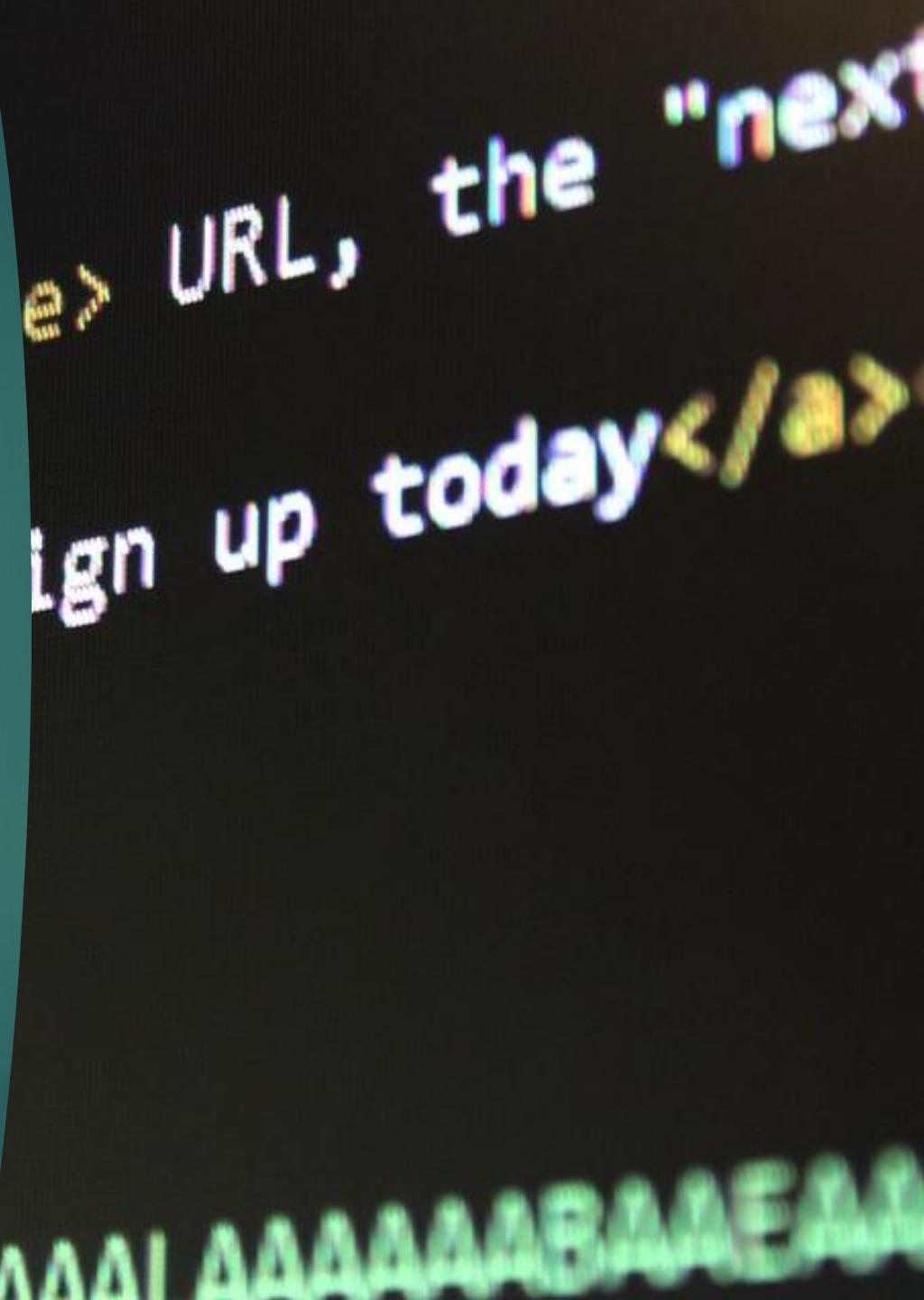
- Organize content into a logical structure.

Define

- Define elements such as headings, paragraphs, images, links, lists, and forms.

Provide

- Provide semantic meaning to content, making it easier for browsers and search engines to understand.



A Simple HTML Document

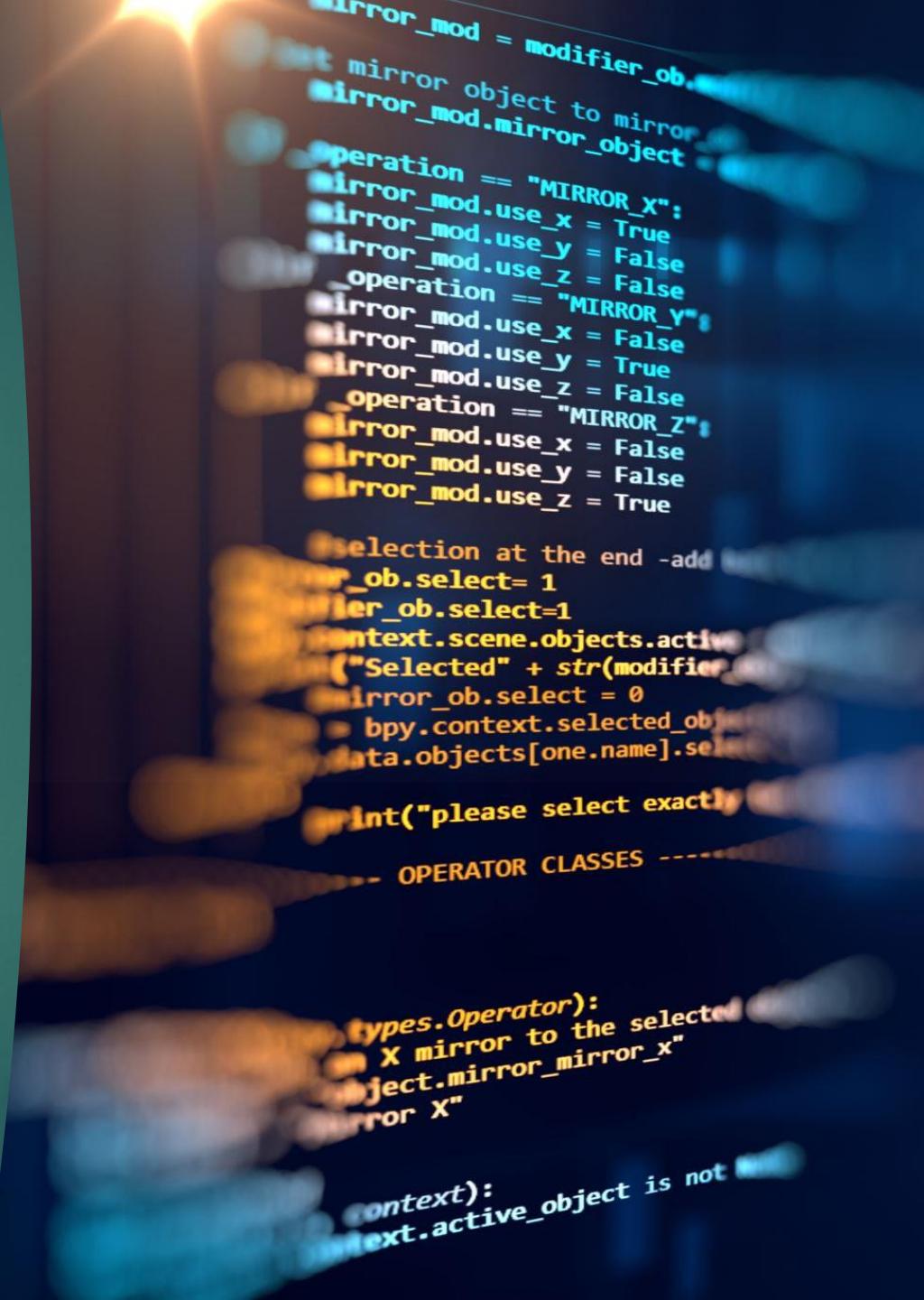
```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Output

My First Heading
My first paragraph.

A photograph of a person's hand pointing towards a computer monitor. The monitor displays a dark-themed code editor with multiple lines of Python-like code. The code includes various conditional statements and function definitions, likely related to 3D modeling or rendering. The background is dark, and the overall lighting is focused on the screen and the person's hand.

```
mirror_mod = modifier_ob
# mirror object to mirror
mirror_mod.mirror_object
operation == "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False
operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

selection at the end -add
    ob.select= 1
    ob.select=1
    context.scene.objects.active
        ("Selected" + str(modifier))
    mirror_ob.select = 0
    bpy.context.selected_objects
    data.objects[one.name].sele
    print("please select exactly one object")
    - OPERATOR CLASSES -
types.Operator):
    X mirror to the selected object.mirror_mirror_x"
    or X"
context):
    context.active_object is not
```

Example Explained

The `<!DOCTYPE html>` declaration defines that this document is an HTML5 document

The `<html>` element is the root element of an HTML page

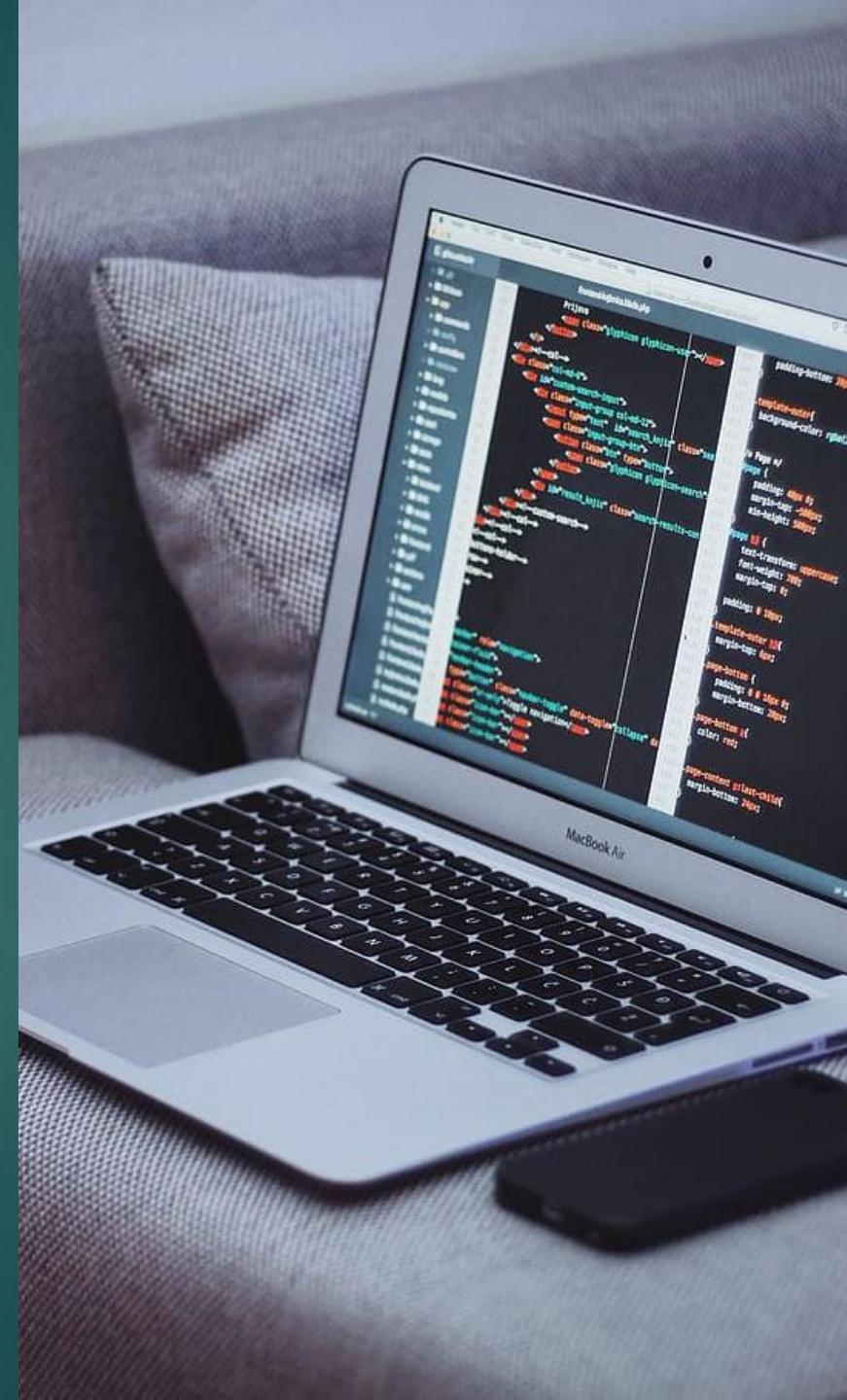
The `<head>` element contains meta information about the HTML page

The `<title>` element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)

The `<body>` element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

The `<h1>` element defines a large heading

The `<p>` element defines a paragraph



What is an HTML Element?

- An HTML element is defined by a start tag, some content, and an end tag:

```
<tagname> Content goes  
here... </tagname>
```

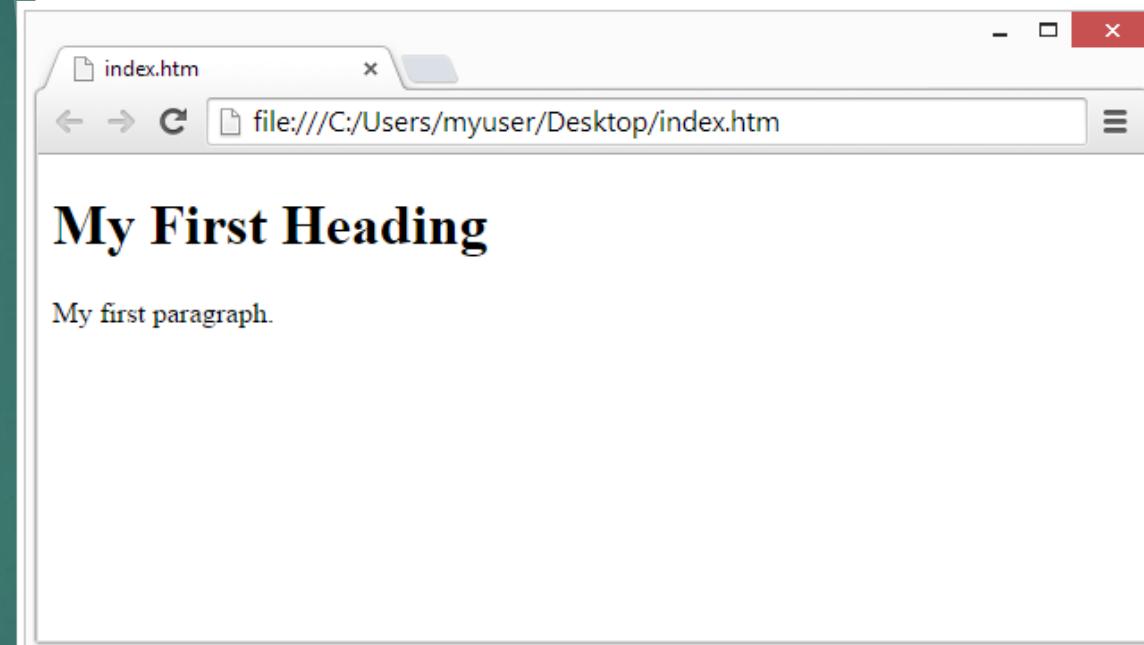
- The HTML **element** is everything from the start tag to the end tag:

```
<h1>My First Heading</h1>  
<p>My first paragraph.</p>
```

Start tag	Element content	End tag
<h1>	My First Heading	</h1>
<p>	My first paragraph.	</p>
 	<i>none</i>	<i>none</i>

Web Browsers

- The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them correctly.
- A browser does not display the HTML tags, but uses them to determine how to display the document:



HTML Page Structure

```
<html>
```

```
  <head>
```

```
    <title>Page title</title>
```

```
  </head>
```

```
  <body>
```

```
    <h1>This is a heading</h1>
```

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

```
    <p>This is another paragraph.</p>
```

```
  </body>
```

```
</html>
```

The <!DOCTYPE> Declaration



The <!DOCTYPE> declaration represents the document type, and helps browsers to display web pages correctly.



It must only appear once, at the top of the page (before any HTML tags).



The <!DOCTYPE> declaration is not case sensitive.



HTML Headings

- ▶ HTML headings are defined with the `<h1>` to `<h6>` tags.
- ▶ `<h1>` defines the most important heading. `<h6>` defines the least important heading:
- ▶ `<h1>This is heading 1</h1>`
`<h2>This is heading 2</h2>`
`<h3>This is heading 3</h3>`
`<h4>This is heading 4</h4>`
`<h5>This is heading 5</h5>`
`<h6>This is heading 6</h6>`

•

OUTPUT

This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

This is heading 6

HTML Paragraphs

- ▶ HTML paragraphs are defined with the <p> tag

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

```
<p>This is another paragraph.</p>
```

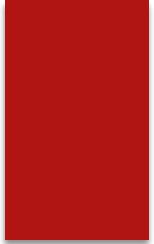
OUTPUT

This is a paragraph.

This is another paragraph.

HTML Attributes

- ▶ HTML attributes provide additional information about HTML elements.
- ▶ HTML Attributes
- All HTML elements can have **attributes**
- Attributes provide **additional information** about elements
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like **name="value"**



```
<HTML>
<html>
<title>HTML</title>
<body>
This is HTML!
</body>
</html>
```

HTML Links

- ▶ HTML links are defined with the `<a>` tag:

```
<a href="https://www.w3schools.com">This is a  
link</a>
```

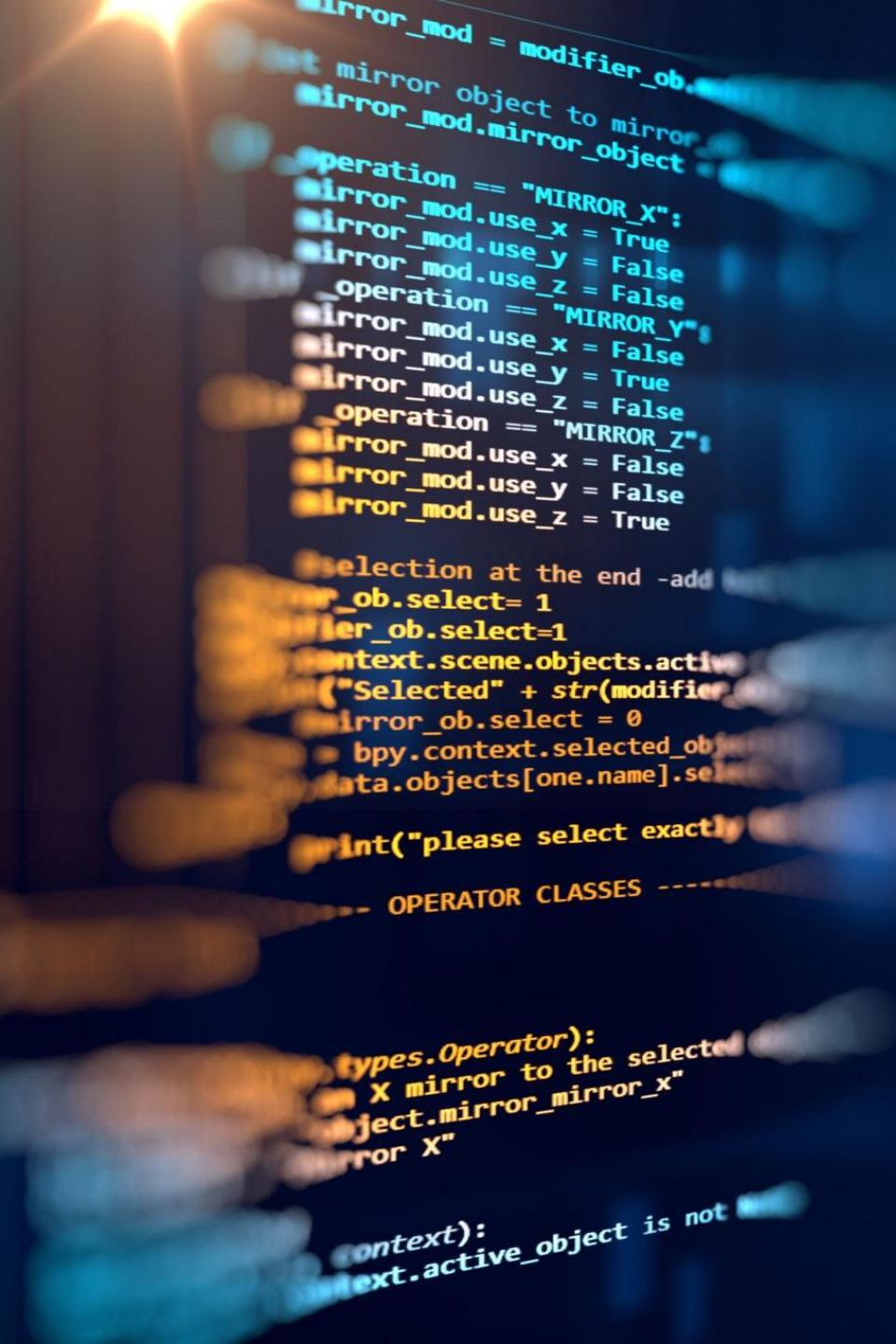
- ▶ The link's destination is specified in the `href` attribute

OUTPUT

HTML Links

HTML links are defined with a tag:

This is a link



The src Attribute

- The `` tag is used to embed an image in an HTML page. The `src` attribute specifies the path to the image to be displayed:

```

```

OUTPUT



The width, height & alt Attributes

- ▶ The `` tag should also contain the width and height attributes, which specify the width and height of the image (in pixels):

```

```

- ▶ The required alt attribute for the `` tag specifies an alternate text for an image, if the image for some reason cannot be displayed. This can be due to a slow connection, or an error in the src attribute, or if the user uses a screen reader.

```

```

OUTPUT

HTML Image



The HTML Style Attribute

- ▶ Setting the style of an HTML element, can be done with the style attribute.
- *<tagname style="property:value;">*

Background Color:

```
<body style="background-color:powderblue;">
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
```

OUTPUT

This is a heading

This is a paragraph.

Set background color for two different elements:

```
<body>  
  
<h1 style="background-color:powderblue;">This is a  
heading</h1>  
<p style="background-color:tomato;">This is a  
paragraph.</p>
```

```
</body>
```

Text Color:

- ▶

```
<h1 style="color:blue;">This is a heading</h1>  
<p style="color:red;">This is a paragraph.</p>
```

OUTPUT

This is a heading

This is a paragraph.

OUTPUT

This is a heading

This is a paragraph.

- The CSS font-family property defines the font to be used for an HTML element:

```
<h1 style="font-family:verdana;">This is a  
heading</h1>  
<p style="font-family:courier;">This is a  
paragraph.</p>
```

OUTPUT

This is a heading

This is a paragraph.

Text size & alignment

- ▶ <h1 style="font-size:300%;">This is a heading</h1>
<p style="font-size:160%;">This is a paragraph.</p>
- ▶ <h1 style="text-align:center;">Centered Heading</h1>
<p style="text-align:center;">Centered paragraph.</p>

OUTPUT

This is a heading

This is a paragraph.

OUTPUT

Centered Heading

Centered paragraph.

HTML Text Formatting

- ▶ This text is bold
- ▶ This text is important!
- ▶ <i>This text is italic</i>
- ▶ This text is emphasized
- ▶ <small>This is some smaller text.</small>
- ▶ <p>Do not forget to buy <mark>milk</mark> today.</p>
- ▶ <p>My favorite color is blue <ins>red</ins>. </p>

OUTPUT

This text is bold.

OUTPUT

This text is important!

OUTPUT

This text is italic

OUTPUT

This text is emphasized.

OUTPUT

This is some smaller text.

OUTPUT

milk today.

OUTPUT

My favorite color is ~~blue~~ red.

HTML Comments

- ▶ HTML comments are not displayed in the browser, but they can help document your HTML source code:

```
<!DOCTYPE html>

<html>
  <body>
    <!-- This is a comment -->
    <p>This is a paragraph.</p>
    <!-- Comments are not displayed in the browser -->
  </body>
</html>
```

Border Color

```
<!DOCTYPE html>

<html>
<body>
<h1 style="border: 2px solid Tomato;">Hello World</h1>
<h1 style="border: 2px solid DodgerBlue;">Hello World</h1>
<h1 style="border: 2px solid Violet;">Hello World</h1>
</body>
</html>
```

Border Color

You can set the color of borders:

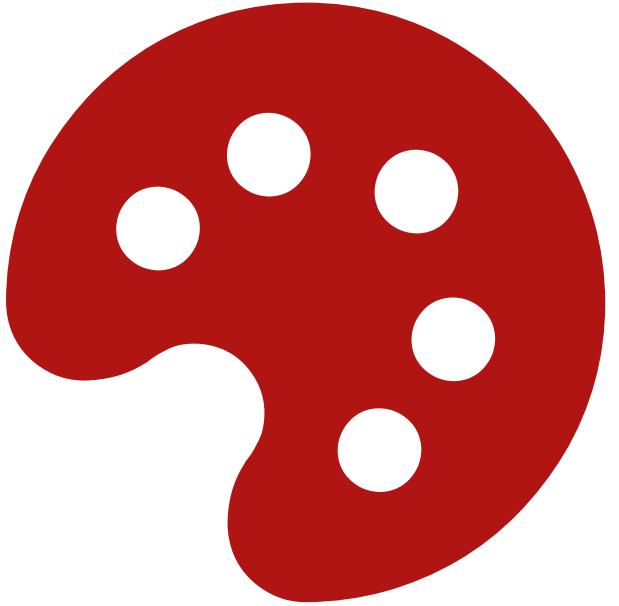
Hello World

Hello World

Hello World

Html Colors

- ▶ https://www.w3schools.com/html/html_colors.asp



Link to an Email Address

- Use mailto: inside the href attribute to create a link that opens the user's email program

```
<body>  
<h2>Link to an Email Address</h2>  
<p>To create a link that opens in the user's email program (to let them  
send a new email), use mailto: inside the href attribute:</p>  
<p><a href="mailto:someone@example.com">Send email</a></p>
```

```
</body>
```

► OUTPUT

Link to an Email Address

To create a link that opens in the user's email program (to let them send a new email), use mailto: inside the href attribute:

[Send email](mailto:someone@example.com)



Button as a Link

- ▶ To use an HTML button as a link, you have to add some JavaScript code.
- ▶ JavaScript allows you to specify what happens at certain events, such as a click of a button:

```
<body>  
<h2>Button as a Links</h2>  
<p>Click the button to go to the HTML tutorial.</p>  
<button onclick="document.location='default.asp'">HTML  
Tutorial</button>  
</body>
```

OUTPUT

Button as a Links

Click the button to go to the HTML tutorial.

[HTML Tutorial](#)

Html Link

- Use a full URL to link to a web page:

```
<a href="https://www.w3schools.com/html/default.asp">HTML  
tutorial</a>
```

- Link to a page located in the html folder on the current web site:

```
<a href="/html/default.asp">HTML tutorial</a>
```

- Link to a page located in the same folder as the current page:

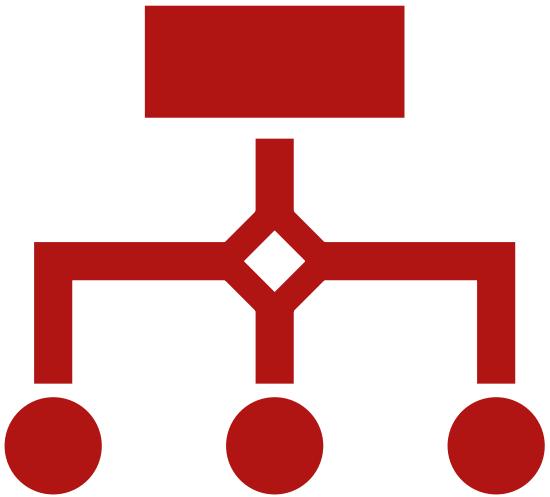
```
<a href="default.asp">HTML tutorial</a>
```

Output:

[HTML tutorial](#)

References:

https://www.w3schools.com/html/html_links.asp



HTML Image Maps

- The HTML <map> tag defines an image map. An image map is an image with clickable areas. The areas are defined with one or more <area> tags.

```


<map name="workmap">
  <area shape="rect" coords="34,44,270,350" alt="Computer" href
="computer.htm">
  <area shape="rect" coords="290,172,333,250" alt="Phone" href
="phone.htm">
  <area shape="circle" coords="337,300,44" alt="Coffee" href="c
offee.htm">
</map>
```

Reference:

https://www.w3schools.com/html/html_images_imagemap.asp

OUTPUT

Image Maps

Click on the computer, the phone, or the cup of coffee to go to a new page and read more about the topic:



HTML Background Images

► <style>

```
body {  
background-image:  
url('img_girl.jpg');  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
  <h2>Background Image</h2>
```

```
  <p> </p>
```

```
</body>
```

Reference:

https://www.w3schools.com/html/html_images_background.asp

Background Image



HTML Forms

- An HTML form is used to collect user input. The user input is most often sent to a server for processing.

First name:

Last name:

The <input> Element

The HTML `<input>` element is the most used form element.

An `<input>` element can be displayed in many ways, depending on the `type` attribute.

Here are some examples:

Type	Description
<code><input type="text"></code>	Displays a single-line text input field
<code><input type="radio"></code>	Displays a radio button (for selecting one of many choices)
<code><input type="checkbox"></code>	Displays a checkbox (for selecting zero or more of many choices)
<code><input type="submit"></code>	Displays a submit button (for submitting the form)
<code><input type="button"></code>	Displays a clickable button

All the different input types are covered in this chapter: [HTML Input Types](#).

The <input> Element

FORM creation

```
<h2>Text input fields</h2>
```

```
<form>  
<label for="fname">First name:</label><br>  
<input type="text" id="fname" name="fname" value="John"><br>  
<label for="lname">Last name:</label><br>  
<input type="text" id="lname" name="lname" value="Doe">  
</form>
```

Reference:

https://www.w3schools.com/html/html_forms.asp

OUTPUT

Text input fields

First name:

John

Last name:

Doe

Radio Buttons

- ▶ The `<input type="radio">` defines a radio button.
- ▶ Radio buttons let a user select ONE of a limited number of choice.

```
<h2>Radio Buttons</h2>  
<p>Choose your favorite Web language:</p>  
<form>  
  <input type="radio" id="html" name="fav_language"  
        value="HTML">  
  <label for="html">HTML</label><br>  
  <input type="radio" id="css" name="fav_language"  
        value="CSS">  
  <label for="css">CSS</label><br>  
  <input type="radio" id="javascript" name="fav_language"  
        value="JavaScript">  
  <label for="javascript">JavaScript</label>  
</form>
```

OUTPUT

Radio Buttons

Choose your favorite Web language:

- HTML
- CSS
- JavaScript

The Submit Button

- ▶ The `<input type="submit">` defines a button for submitting the form data to a form-handler.
- ▶ The form-handler is typically a file on the server with a script for processing input data.

```
<form action="/action_page.php">  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" name="fname" value="John"><br>  
  <label for="lname">Last name:</label><br>  
  <input type="text" id="lname" name="lname" value="Doe"><br><br>  
  <input type="submit" value="Submit">  
</form>
```

OUTPUT

HTML Forms

First name:

John

Last name:

Doe

Submit

Reference for more tags & attributes

- ▶ <https://www.w3schools.com/html/default.asp>
- ▶ <https://www.tutorialspoint.com/html/index.htm>
- ▶ <https://www.youtube.com/watch?v=dX8396ZmSPk>

Student's Task

Create

Create an HTML page with:

- A title and heading.
- Two paragraphs with different text styles.
- A link to your favorite website.

Add

Add an image to the page with appropriate alt, width, and height attributes.

Design

Design an HTML form that collects a user's first name, last name, and email, with a submit button.



Thank you

Week 5

Lecture 5

HTML LIST & TABLE CREATION

What are Lists?

Compound tags

3 different types

UL: Unordered lists

OL: Ordered lists

DL: Definition lists

Unordered HTML List

- Order doesn't matter

```
<ul>  
<li>Coffee</li>  
<li>Tea</li>  
<li>Milk</li>  
</ul>
```

OUTPUT

An unordered HTML list

- Coffee
- Tea
- Milk

Ordered HTML List

```
<ol>  
<li>Coffee</li>  
<li>Tea</li>  
<li>Milk</li>  
</ol>
```

OUTPUT

An ordered HTML list

1. Coffee
2. Tea
3. Milk

HTML Description Lists

- ▶ A description list is a list of terms, with a description of each term.
- ▶ The <dl> tag defines the description list, the <dt> tag defines the term (name), and the <dd> tag describes each term

```
<dl>  
  <dt>Coffee</dt>  
  <dd>- black hot drink</dd>  
  <dt>Milk</dt>  
  <dd>- white cold drink</dd>  
</dl>
```

OUTPUT

A Description List

Coffee

- black hot drink

Milk

- white cold drink

Reference for lists

- ▶ https://www.w3schools.com/html/html_lists.asp
- ▶ https://www.slideshare.net/slideshow/unit-2-hypertext-markup-language-cascading-style-sheets/271220114?from_search=0



HTML Tables

- ▶ HTML tables allow web developers to arrange data into rows and columns.
- ▶ Each table cell is defined by a `<td>` and a `</td>` tag.

`<h2>TD elements define table cells</h2>`

```
<table style="width:100%">  
  
  <tr>  
    <td>Emil</td>  
    <td>Tobias</td>  
    <td>Linus</td>  
  </tr>  
  
</table>
```

OUTPUT

TD elements define table cells

Emil	Tobias	Linus
------	--------	-------

Table Rows

- Each table row starts with a `<tr>` and ends with a `</tr>` tag.

```
<style>  
table, th, td {  
    border:1px solid black;  
}  
</style>
```

```
<body>  
<tr>  
    <td>Emil</td>  
    <td>Tobias</td>  
    <td>Linus</td>  
</tr>
```

```
<tr>  
    <td>16</td>  
    <td>14</td>  
    <td>10</td>
```

```
</tr>
```

```
</table>
```

```
</body>
```

OUTPUT

Emil	Tobias	Linus
16	14	10

HTML Table Borders

To add a border, use the CSS border property on table, th, and td elements

To avoid having double borders like in the example above, set the CSS border-collapse property to collapse.

With the border-radius property, the borders get rounded corners

The diagram illustrates a table structure with three columns labeled 'header1', 'header2', and 'header3'. The first column contains two rows of data, and the second column contains four rows of data. The table uses a combination of blue and black borders. Red borders are used to highlight the vertical borders of the first column and the horizontal borders of the second column, demonstrating how border collapse prevents double borders from appearing at the intersections. The bottom-right corner of the entire table area is rounded, demonstrating the use of border-radius.

header1	header2	header3
		header3

How To Add a Border

```
table, th, td {  
    border: 1px solid black;  
}
```

```
table, th, td {  
    border: 4px solid black;  
}
```

OUTPUT

Table With Border

Use the CSS border property to add a border to the table.

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

OUTPUT

Table With Border

Use the CSS border property to add a border to the table.

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

Collapsed Table Borders

```
table, th, td {  
    border: 1px solid black;  
    border-collapse: collapse;  
}
```

OUTPUT

Collapsed Borders

If you want the borders to collapse into one border, add the CSS border-collapse property.

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

Round Table Borders

```
table, th, td {  
    border: 1px solid black;  
    border-radius: 10px;  
}
```

OUTPUT

Table With Rounded Borders

Use the CSS border-radius property to add rounded corners to the borders.

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

Border Color

```
th, td {  
    border-color: #96D4D4;  
}
```

OUTPUT

Table With Border Color

Use the CSS border-color property to set the color of the borders.

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

HTML Table Width

```
<table style="width:100%">
<tr>
  <th>Firstname</th>
  <th>Lastname</th>
  <th>Age</th>
</tr>
<tr>
  <td>Jill</td>
  <td>Smith</td>
  <td>50</td>
</tr>
<tr>
  <td>Eve</td>
  <td>Jackson</td>
  <td>94</td>
</tr>
</table>
```

OUTPUT

100% wide HTML Table

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

HTML Table Row Height

```
<table style="width:100%">
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Age</th>
</tr>
<tr style="height:200px">
<td>Jill</td>
<td>Smith</td>
<td>50</td>
</tr>
<tr>
<td>Eve</td>
<td>Jackson</td>
<td>94</td>
</tr>
</table>
```

OUTPUT

Set the height of the second row to 100 pixels

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

HTML Table - Colspan

```
<table>
  <tr>
    <th colspan="2">Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>43</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>57</td>
  </tr>
</table>
```

OUTPUT

Cell that spans two columns

To make a cell span more than one column, use the `colspan` attribute.

Name	Age
Jill	Smith
Eve	Jackson

HTML Table - Rowspan

```
<table>
  <tr>
    <th>Name</th>
    <td>Jill</td>
  </tr>
  <tr>
    <th rowspan="2">Phone</th>
    <td>555-1234</td>
  </tr>
  <tr>
    <td>555-8745</td>
  </tr>
</table>
```

OUTPUT

Cell that spans two rows

To make a cell span more than one row, use the rowspan attribute.

Name	Jill
Phone	555-1234
	555-8745

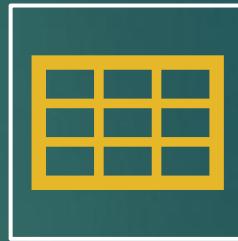
Student's Task



Create an unordered list of your daily activities.



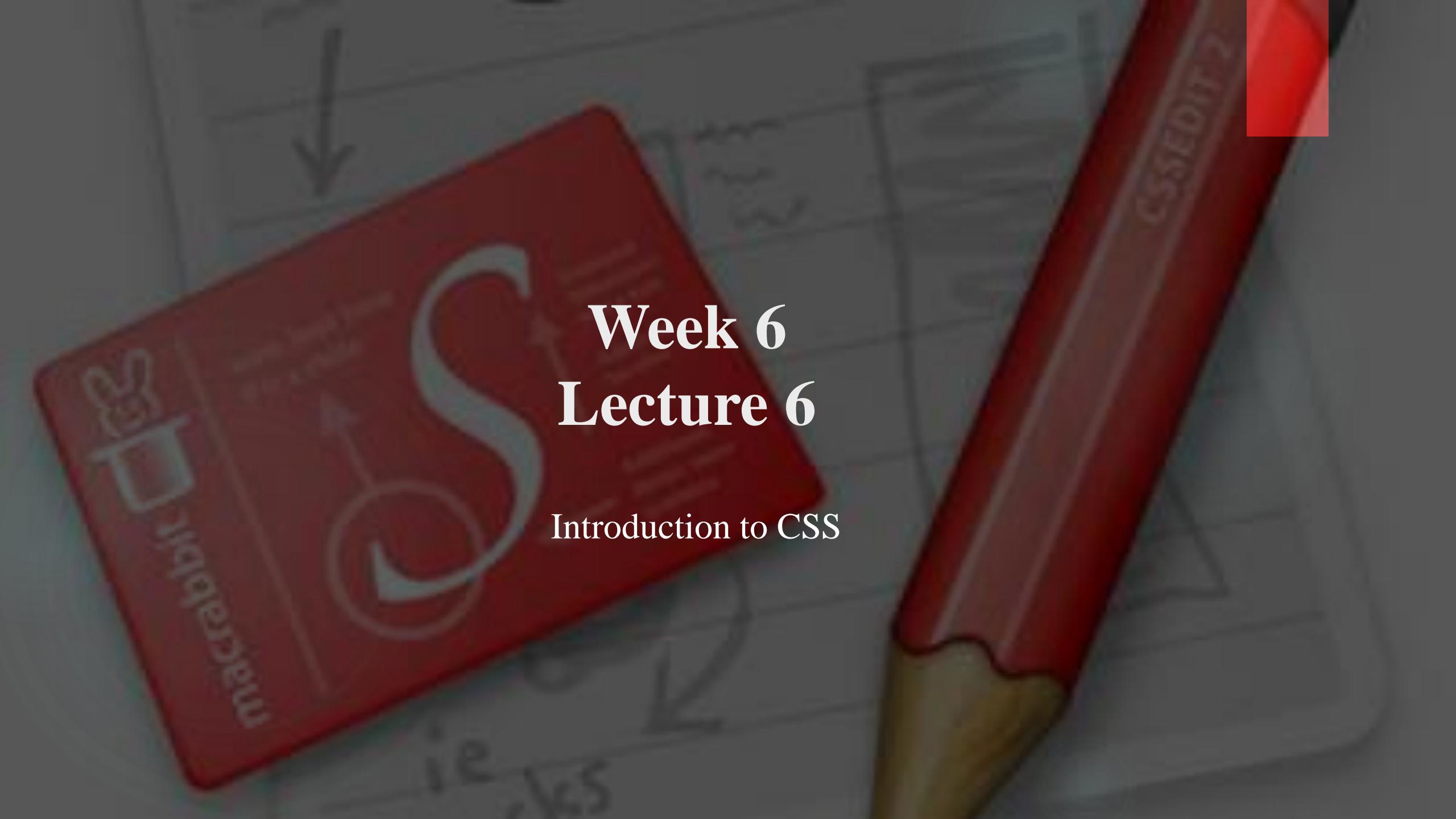
Create an ordered list of your top 5 favorite books or movies.



Design a table displaying a schedule (days vs. times) with at least 2 rows and 2 columns, using different border styles.



Thank you

The background of the slide features a photograph of a red book lying on its side. The word "Success" is printed in large, white, serif capital letters on the front cover. A yellow pencil lies diagonally across the book. The book is resting on a light-colored surface, possibly a desk or table.

Week 6

Lecture 6

Introduction to CSS

Introduction

CSS stands for
Cascading Style
Sheets

CSS describes how
HTML elements are
to be displayed on
screen, paper, or in
other media

External stylesheets are
stored in CSS files

Example

```
body {  
    background-color: lightblue;  
}
```

```
h1 {  
    color: white;  
    text-align: center;  
}
```

```
p {  
    font-family: verdana;  
    font-size: 20px;  
}
```

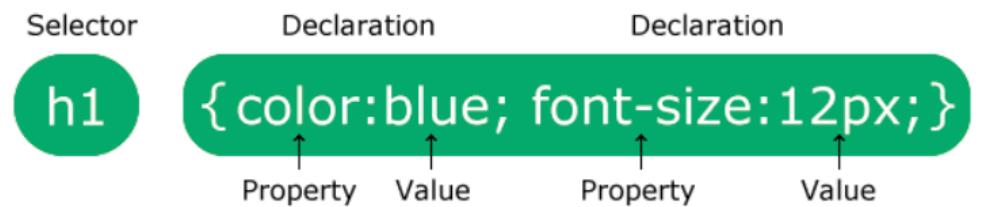
OUTPUT

My First CSS Example

This is a paragraph.

CSS Syntax

- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.



CSS Selectors

- The element selector selects HTML elements based on the element name.

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

OUTPUT

Every paragraph will be affected by the style.

Me too!

And me!

The CSS id Selector

- ▶ The id selector uses the id attribute of an HTML element to select a specific element.
- ▶ To select an element with a specific id, write a hash (#) character, followed by the id of the element.

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

OUTPUT

Hello World!

This paragraph is not affected by the style.

The CSS class Selector

- To select elements with a specific class, write a period (.) character, followed by the class name.

```
<head>
<style>
.center {
    text-align: center;
    color: red;
}
</style>
</head>
<body>
<h1 class="center">Red and center-aligned heading</h1>
<p class="center">Red and center-aligned paragraph.</p>
</body>
```

OUTPUT

Red and center-aligned heading

Red and center-aligned paragraph.

CSS Text Color

```
<h3 style="color:Tomato;">Hello World</h3>  
<p style="color:DodgerBlue;">Lorem ipsum dolor  
sit amet, consectetuer adipiscing elit, sed diam  
nonummy nibh euismod tincidunt ut laoreet dolore  
magna aliquam erat volutpat.</p>  
<p style="color:MediumSeaGreen;">Ut wisi enim ad  
minim veniam, quis nostrud exerci tation  
ullamcorper suscipit lobortis nisl ut aliquip ex ea  
commodo consequat.</p>
```

References:

https://www.w3schools.com/css/css_colors.asp

OUTPUT

Hello World

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

CSS Margins

- The CSS margin properties are used to create space around elements, outside of any defined borders

CSS has properties for specifying the margin for each side of an element:

```
•margin-top  
•margin-right  
•margin-bottom  
•margin-left  
<style>  
div {  
    border: 1px solid black;  
    margin-top: 100px;  
    margin-bottom: 100px;  
    margin-right: 150px;  
    margin-left: 80px;  
    background-color: lightblue;  
}  
</style>  
</head>  
<body>
```

OUTPUT

Using individual margin properties

This div element has a top margin of 100px, a right margin of 150px, a bottom margin of 100px, and a left margin of 80px.

CSS Margins

```
<style>
div {
    border: 1px solid black;
    margin: 25px 50px 75px 100px;
    background-color: lightblue;
}
</style>
```

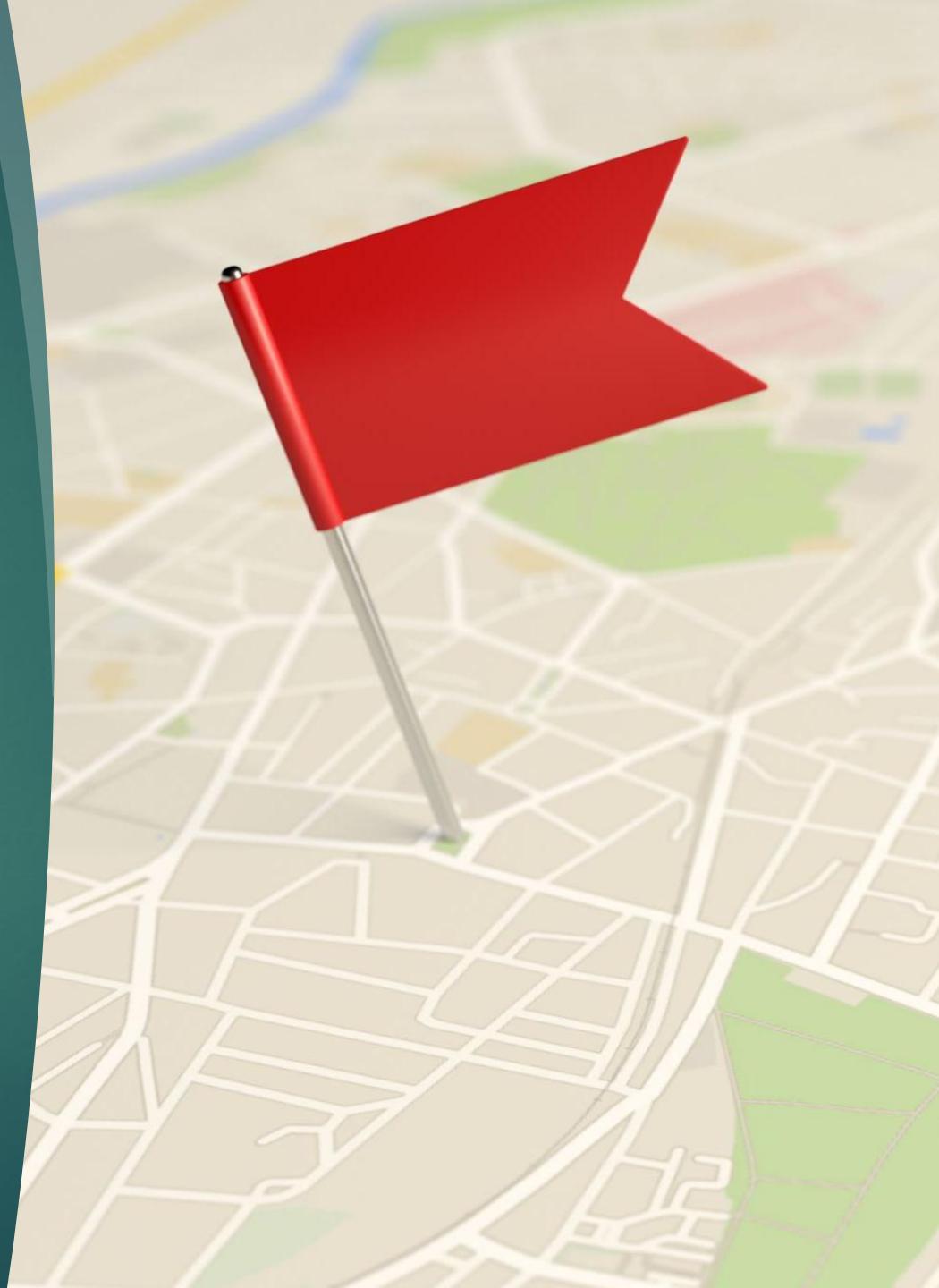
OUTPUT

The margin shorthand property - 4 values

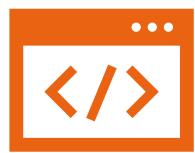
This div element has a top margin of 25px, a right margin of 50px, a bottom margin of 75px, and a left margin of 100px.

Margin & Border References

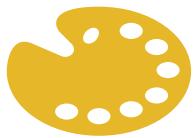
- ▶ https://www.w3schools.com/css/css_border.asp
- ▶ https://www.w3schools.com/css/css_margin.asp
- ▶ <https://www.geeksforgeeks.org/css-borders/>
- ▶ <https://www.programiz.com/css/borders>
- ▶ <https://www.geeksforgeeks.org/css-margins-padding/>



Student's Task



Style the HTML page from Week 2 using inline CSS to:



Change the background color of the page.



Set the font style and size of headings and paragraphs.



Experiment with CSS selectors to change the text color for specific elements.



Thank you

CSS Box Model

Week 7 & 8 Lecture 7 & 8

CSS Box Model and Layouts

Margin

Border

Padding

Margin - The empty area around the border. The margin is completely transparent, no background color.

Border - A border that goes around the padding and content. The border is affected by the background color of the box

Padding - Clears an area around the content. The padding is affected by the background color of the box

CSS Padding

If
the padding property
has four values:

**padding: 25px 50px
75px 100px;**

top padding is 25px

right padding is
50px

bottom padding is
75px

left padding is 100px

CSS Padding

```
<style>  
div {  
    border: 1px solid black;  
    padding: 15px 20px 50px 50px;  
    background-color: lightblue;  
}  
</style>
```

OUTPUT

The padding shorthand property - 4 values

This div element has a top padding of 25px, a right padding of 50px, a bottom padding of 75px, and a left padding of 100px.

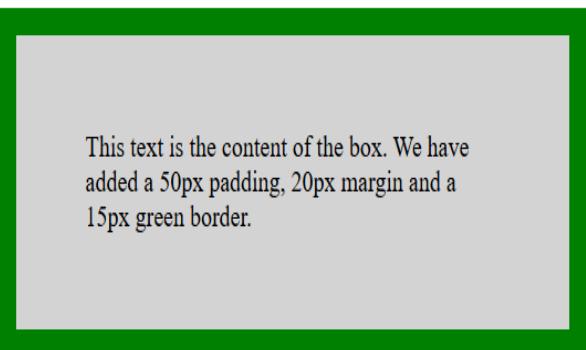
CSS Box Model

```
<head>
<style>
div {
    background-color: lightgrey;
    width: 300px;
    border: 15px solid green;
    padding: 50px;
    margin: 20px;
}
</style>
</head>
<body>
<h2>Demonstrating the Box Model</h2>
<p>The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.</p>
<div>This text is the content of the box. We have added a 50px
```

OUTPUT

Demonstrating the Box Model

The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.



This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border.

CSS Outline



CSS has the following outline properties:



outline-style



outline-color



outline-width



outline-offset



outline

CSS Outline Style

```
p.dotted {outline-style: dotted;}  
p.dashed {outline-style: dashed;}  
p.solid {outline-style: solid;}  
p.double {outline-style: double;}  
p.groove {outline-style: groove;}  
p.ridge {outline-style: ridge;}  
p.inset {outline-style: inset;}  
p.outset {outline-style: outset;}
```

Reference:

https://www.w3schools.com/css/css_outline.asp

OUTPUT

The outline-style Property

A dotted outline

A dashed outline

A solid outline

A double outline

A groove outline. The effect depends on the outline-color value.

A ridge outline. The effect depends on the outline-color value.

An inset outline. The effect depends on the outline-color value.

An outset outline. The effect depends on the outline-color value.

Text Color and Background Color

```
body {  
    background-color: lightgrey;  
    color: blue;  
}  
  
h1 {  
    background-color: black;  
    color: white;  
}  
  
div {  
    background-color: blue;  
    color: white;  
}
```

Reference:

https://www.w3schools.com/css/css_text.asp

OUTPUT

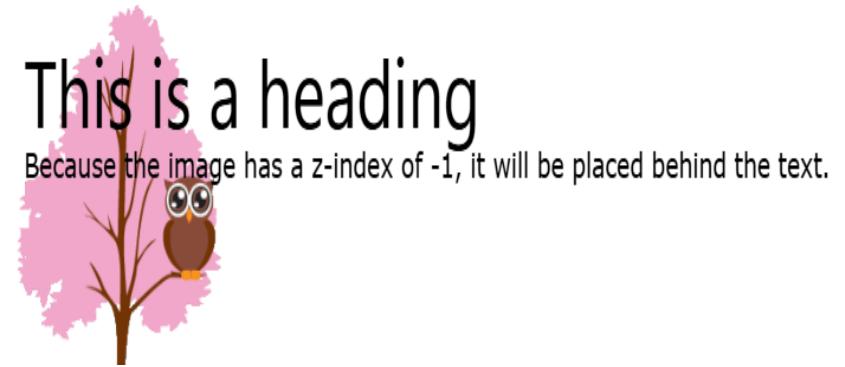
This is a Heading

This page has a grey background color and a blue text.

This is a div.

The z-index Property

- ▶ The z-index property specifies the stack order of an element (which element should be placed in front of, or behind, the others).
- ▶ An element can have a positive or negative stack order:



Example of Z index

```
img {  
    position: absolute;  
    left: 0px;  
    top: 0px;  
    z-index: -1;  
}
```

```
img {  
    position: absolute;  
    left: 0px;  
    top: 0px;  
    z-index: 1;  
}
```

OUTPUT

This is a heading

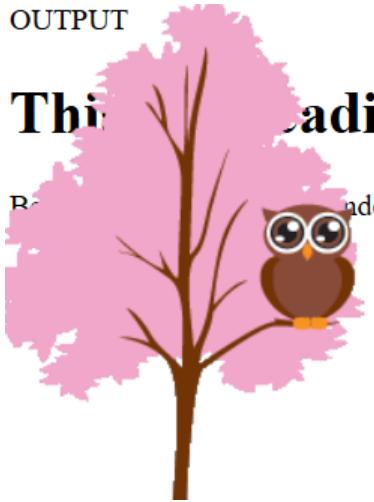
Because the image has a z-index of -1, it will be placed behind the text.



OUTPUT

This is a heading

Because the image has a z-index of 1, it will be placed in front of the text.



CSS Opacity

- The opacity property can take a value from 0.0 - 1.0. The lower the value, the more transparent

```
img {  
    opacity: 0.5;  
}
```

```
img {  
    opacity: 1;  
}
```

Reference:

https://www.w3schools.com/css/css_image_transparency.asp

OUTPUT

Image Transparency

The opacity property specifies the transparency of an element. The lower the value, the more transparent:

Image with 50% opacity:



OUTPUT

Image Transparency

The opacity property specifies the transparency of an element. The lower the value, the more transparent:

Image with 100% opacity:



CSS Forms

► Padded Inputs

```
input[type=text] {  
    width: 100%;  
    padding: 12px 20px;  
    margin: 8px 0;  
    box-sizing: border-box;  
}
```

OUTPUT

Padded input fields

First Name

Last Name

CSS Forms

- ▶ Input with icon/image

```
input[type=text] {  
    background-color: white;  
    background-image: url('searchicon.png');  
    background-position: 10px 10px;  
    background-repeat: no-repeat;  
    padding-left: 40px;  
}
```

OUTPUT

Input field with an icon inside

A search input field with a magnifying glass icon and placeholder text "Search..".

CSS Form

Reference:

https://www.w3schools.com/css/css_form.asp



A photograph of a person's hand holding a small, detailed model house. The house is white with a brown roof, a balcony with a black railing, and several windows with brown frames. It is positioned in the lower-left corner of the slide.

The position Property

- ▶ There are five different position values:
 - static
 - relative
 - fixed
 - absolute
 - sticky

Position

- ▶

```
div.static {  
    position: static;  
    border: 3px solid #73AD21;  
}
```
- ▶

```
div.relative {  
    position: relative;  
    left: 30px;  
    border: 3px solid #73AD21;  
}
```
- ▶

```
div.fixed {  
    position: fixed;  
    bottom: 0;  
    right: 0;  
    width: 300px;  
    border: 3px solid #73AD21;  
}
```

OUTPUT

position: static;

An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:

This div element has position: static;

OUTPUT

position: relative;

An element with position: relative; is positioned relative to its normal position:

This div element has position: relative;

OUTPUT

position: fixed;

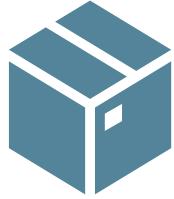
An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled:

Reference for more layout attribute

- https://www.w3schools.com/css/css_inline-block.asp
- https://www.w3schools.com/css/css_align.asp
- https://www.w3schools.com/css/css_combinators.asp
- https://www.w3schools.com/css/css_navbar.asp
- https://www.w3schools.com/css/css_navbar.asp

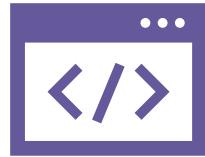


Student's Task



Use

Use the box model to create a styled content box:



Add

Add padding, margins, and a border with different colors.



Design

Design a layout with a header, navigation bar, content area, and footer using CSS.



Thank you



Week 9, 10 & 11

Lecture 9, 10 & 11

Advanced CSS Styling

CSS Math Functions

- The calc() function performs a calculation to be used as the property value.

```
#div1 {  
    position: absolute;  
    left: 50px;  
    width: calc(100% - 100px);  
    border: 1px solid black;  
    background-color: yellow;  
    padding: 5px;  
}
```

OUTPUT

The calc() Function

Create a div that stretches across the window, with a 50px gap between both sides of the div and the edges of the window:

Some text...

The max() Function

- The max() function uses the largest value, from a comma-separated list of values, as the property value.

```
#div1 {  
background-color: yellow;  
height: 100px;  
width: max(50%, 300px);  
}
```

OUTPUT

The max() Function

Use max() to set the width of #div1 to whichever value is largest, 50% or 300px:

Some text...

Resize the browser window to see the effect.

CSS Gradients

```
#grad {  
  background-image: linear-gradient(red,  
yellow);  
}
```

Reference:

https://www.w3schools.com/css/css3_gradients.asp

Linear Gradient - Top to Bottom

This linear gradient starts red at the top, transitioning to yellow at the bottom:



CSS Shadow Effects

CSS Text Shadow

```
h1 {  
    text-shadow: 2px 2px;  
}
```

OUTPUT

Text-shadow effect!

Multiple Shadows

```
h1 {  
    color: white;  
    text-shadow: 1px 1px 2px black, 0 0 25px blue, 0 0 5px  
    darkblue;  
}
```

OUTPUT

Text-shadow effect!

CSS 2D Transforms Functions

► The translate() Function

```
div {  
  transform: translate(50px, 100px);  
}
```

OUTPUT

The translate() Method

The `translate()` method moves an element from its current position:

This div element is moved 50 pixels to the right, and 100 pixels down from its current position.

The rotate() Function

- The rotate() function rotates an element clockwise or counter-clockwise according to a given degree.

```
div {  
    transform: rotate(20deg);  
}
```

Reference:

https://www.w3schools.com/css/css3_2dtransforms.asp

The rotate() Method

The rotate() method rotates an element clockwise or counter-clockwise.

This a normal div element.

This div element is rotated clockwise 20 degrees.

CSS Transitions

► To create a transition effect, you must specify two things:

- the CSS property you want to add an effect to
- the duration of the effect

```
div {  
    width: 100px;  
    height: 100px;  
    background: red;  
    transition: width 2s;  
}
```

► The transition effect will start when the specified CSS property (width) changes value.

CSS Flexbox

- ▶ Flexbox is a layout method for arranging items in rows or columns.
- ▶ A flexbox always consists of:
 - a **Flex Container** - the parent (container) <div> element
 - **Flex Items** - the items inside the container <div>
- ▶

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
```

Reference:

https://www.w3schools.com/css/css3_flexbox.asp

OUTPUT

Create a Flex Container



A Flexible Layout must have a parent element with the *display* property set to *flex*.

Direct child elements(s) of the flexible container automatically becomes flexible items.

Grid-View

```
.menu {  
    width: 25%;  
    float: left;  
}  
  
.main {  
    width: 75%;  
    float: left;  
}
```

Reference:

https://www.w3schools.com/css/css_rwd_grid.asp

OUTPUT

Chania

- The Flight
- The City
- The Island
- The Food

The City

Chania is the capital of the Chania region on the island of Crete. The city can be divided in two parts, the old town and the modern city.

Resize the browser window to see how the content respond to the resizing.

CSS animation-play-state Property

```
div {  
  animation-play-state: paused;  
}  
  
div {  
  animation-timing-function: linear;  
}
```

OUTPUT

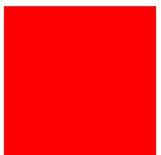
The **animation-play-state** Property

Pause an animation:



The **animation-timing-function** Property

Play an animation with the same speed from beginning to end:



CSS more attributes

Reference:

https://www.w3schools.com/css/css3_borders.asp

https://www.w3schools.com/css/css3_border_images.asp

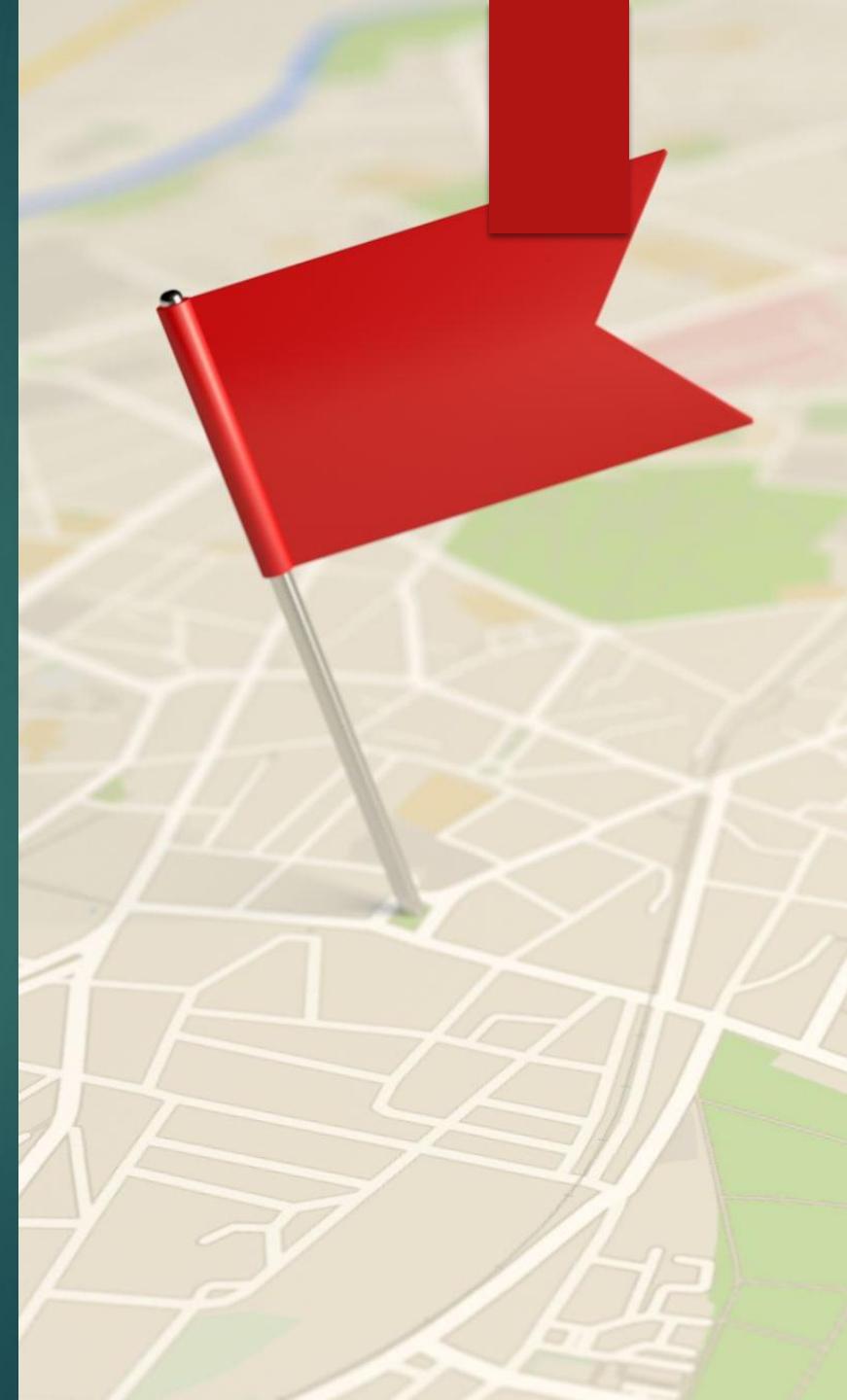
https://www.w3schools.com/css/css3_3dtransforms.asp

https://www.w3schools.com/css/css3_masking.asp

https://www.w3schools.com/css/css3_property.asp

https://www.w3schools.com/css/css_rwd_intro.asp

<https://www.w3schools.com/sass/default.php>



Student's Task



Add a gradient background to the HTML page.



Create animations for buttons using hover effects and CSS transitions.



Use CSS Flexbox to align three items (like a navbar).



Thank you

Week 12 & 13
Lecture 12 & 13

Bootstrap

Introduction to Bootstrap

Introduction

- ▶ Bootstrap is a free front-end framework for faster and easier web development.
- ▶ Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins



Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</head>
<body>
  <div class="container">
    <h1>My First Bootstrap Page</h1>
    <p>This is some text.</p>
  </div>
</body>
</html>
```

OUTPUT

My First Bootstrap Page

This part is inside a .container class.

The .container class provides a responsive fixed width container.

Bootstrap Grids

```
<!DOCTYPE html>

<html lang="en">
<head>

<title>Bootstrap Example</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

</head>

<body>
```

```
<div class="container-fluid">
<h1>Hello World!</h1>
<p>Resize the browser window to see the effect.</p>
<p>The columns will automatically stack on top of each other when the screen is less than 768px wide.</p>
<div class="row">
<div class="col-sm-4" style="background-color:lavender;">.col-sm-4</div>
<div class="col-sm-4" style="background-color:lavenderblush;">.col-sm-4</div>
```

OUTPUT

Hello World!

Resize the browser window to see the effect.

The columns will automatically stack on top of each other when the screen is less than 768px wide.

.col-sm-4

.col-sm-4

.col-sm-4

Reference

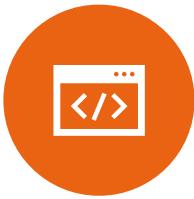
<https://www.w3schools.com/bootstrap/default.asp>

<https://getbootstrap.com/docs/5.3/getting-started/introduction/>

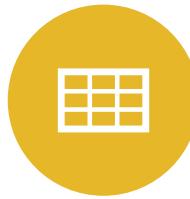
<https://www.youtube.com/watch?v=6zKLvZSYXmE&list=PLdPwRNmUlk0nyw7VoYZvtlU1XIRUfEW-U>



Student's Task



Create a responsive webpage with Bootstrap that includes:
A navigation bar.



A grid layout with three columns.



A styled button using Bootstrap classes.



Thank you

Your Plan Information

Free

✓ Designs Created

-25 / 15

Resetting in 23 days

Designs Created includes designs that are downloaded or sent to drafts. Your design count resets each month.

Enjoy These Perks and More When You Upgrade

- ✓ More designs per month
- ✓ Custom font uploading
- ✓ Premium stock photos and elements

See Tailwind Create Plans

Tips and Tricks

See All

Learn how to create Pins, edit your designs, and more.

Create Beautiful Pins Faster
Than Ever Before

Week 15 & 16

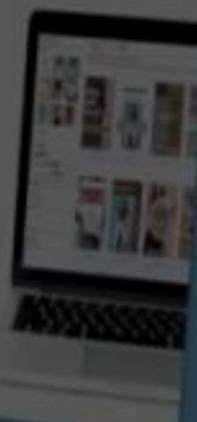
Lecture 15 & 16

Tailwind CSS Basics

Start a New Design

TUTORIAL

MASTER
TAILWIND
CREATE IN
5 MINUTES



Tailwind CSS Basics

Utility-first CSS Framework: Tailwind provides a set of predefined utility classes to style HTML elements directly.

Highly Customizable: It allows developers to extend or overwrite default configurations.

No Predefined Components: Unlike frameworks like Bootstrap, Tailwind focuses on utilities rather than prebuilt UI components.

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <!-- Tailwind CSS CDN link -->
    <link href=
        "https://unpkg.com/tailwindcss@^2/dist/tailwind.min.css"
        rel="stylesheet">
</head>
<body class="m-4">
    <h1 class="text-green-500 text-4xl font-bold">
        Geeksforgeeks
    </h1>
    <p><strong>Tailwind CSS Tutorial</strong></p>
    <p>
        You can use Tailwind CSS as a replacement
        of CSS, this is a framework that increase
        your pace to design any website.
    </p>
</body>
</html>
```

Geeksforgeeks
Tailwind CSS Tutorial
You can use Tailwind CSS as a replacement of CSS, this is a framework that increase your pace to design any website.

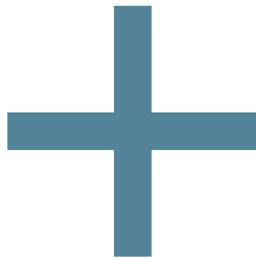
Reference

<https://www.geeksforgeeks.org/introduction-to-tailwind-css/>

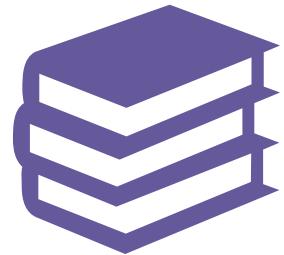
<https://tailwindcss.com/>



Student's Task



Recreate the page you designed in Bootstrap using Tailwind CSS.



Style text elements with Tailwind classes for color, size, and alignment.



Use utility classes to add margins, padding, and borders to elements.



Thank you

Week 17

Lab Test