HTML Introduction

Internet: Network of networks

1969, ARPANet develops a network for Military Communication
1975, NSFNet, used for educational purpose
Later on; big organizations and government departments are used to have their own network
Situation late 80s; we had a lot of networks that are big in size but the problem in that all these networks were working in isolation meant they were not connected to one another.
Vint Cerf proposes connecting all these networks together to make a big global network.
The network that is created after joining all these networks is called a network of networks i.e. internet.

To access the internet we have to use a URL (Uniform Resource Locator)

WWW (World Wide Web):

It refers to the interconnection of documents that are available on net such that this allows to traverse and navigate web pages in a human-friendly way.
97% of the time when people use the internet then they are using WWW so people misunderstand the WWW and the internet as the same thing.

Client:

Anything that demands services from another is called a client.

Server:

Anything that provides services to another is called a server.

Protocol:

Set of rules that is used to govern/regulate communication.

HTTP (Hyper-Text Transfer Protocol):

The protocol that is used for communication between client and server in WWW.

HTTPS (Hyper-Text Transfer Protocol Secure):

HTTP + SSL (Secure Socket Layer)

Encryption: conversion of readale text to unreadable text
Decryption: conversion of unreadable text to readable text

URL (Uniform Resource Locator):

- It is a String which is used to locate resources available on the internet.
- Format of URL: **protocol://subdomain.domain-name.tld/path-of-resource**
- Example: https://vivmagarwal.notion.site/HTML-CSS-links-ebc1fdbef109471291cfe0886515ef01

Here:

Protocol: **https**
Subdomain: **vivmagarwal**
Domain-name: **notion**
Site: **tld**
Path-of-resource: **HTML-CSS-links-ebc1fdbef109471291cfe0886515ef01**

Web Browser:

- An application software that is used to traverse, navigate and access the content available on the internet. e.g. Google Chrome, Mozilla firefox

Web Server:

- A computer/software that contains the content to be accessed through web.

HTTP-Response Status Code:

2xx: Success
3xx: In Process
4xx: Client Error
5xx: Server Error

IP address:

An Internet Protocol (IP) address is the unique identifying number assigned to every device connected to the internet.

Used to identify a machine on the internet. e.g. 192.0.0.25

- Smallest IP address: 0.0.0.0
- Largest IP address: 255.255.255.255

Domain name System:

- The Domain Name System (DNS) turns domain names into IP addresses, which browsers use to load internet pages.

MIME: Multipurpose internet mail extension

- A MIME type is a label used to identify a type of data.
- It is used so software can know how to handle the data.
- It serves the same purpose on the Internet that file extensions do on Microsoft Windows.
- Example, if a server says "This is **text/html**" the client can go "Ah, this is an HTML document, I can render that internally", while if the server says "This is **application/pdf**" the client can go "Ah, I need to launch the FoxIt PDF Reader plugin that the user has installed and that has registered itself as the application/pdf handler."

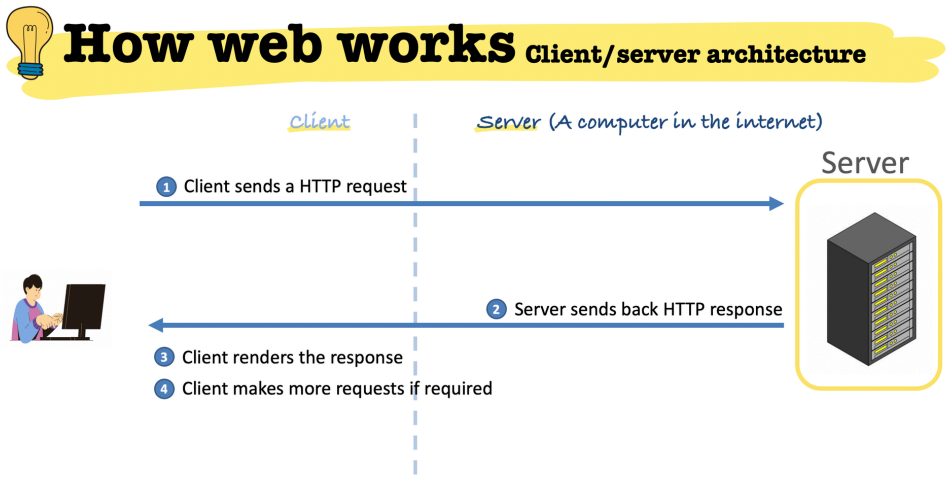
Backend vs Frontend:

- Backend example:
  - https://jsonplaceholder.typicode.com/
  - Backend App returns something similar to this: https://jsonplaceholder.typicode.com/photos?_start=0&_limit=5

- Frontend example:
  - https://codepen.io/drupalastic/pen/rNpXKxN?editors=1010

Student Activity:

- Read the documentation - https://jsonplaceholder.typicode.com/
- Try to load all the users in your browser [simple JSON object - only the code that comes from the backend server]



The client requests a service

- we open a browser (client)
- we hit a url for example: www.google.com (Uniform Resource Location)
    - the client sends a message to the server asks for a resource
    - Resources can be - web pages, images, video files, fonts, stylesheets
    - This message is formatted based on a protocol called `HTTP`
    - In other words, HTTP (Hypertext transfer protocol) is a standard structure (or protocol) that clients and servers use to communicate over internet.
    - With an `HTTP request`, the client communicates to the server, what it is looking for
    -

The server provides the service

- The server listens to the message
- It figures out what the client is asking
- It sends a message back to the client
- This message is called an `HTTP response`

The client gets back a message

- For example, the response can be an html page.

```
<!DOCTYPE html>
<html>
...
  <link rel="stylesheet" href="styles.css">
...
  <img class="footer-sub__logo"
src="https://unsplash.com/assets/core/logo-black-df2168ed0c378fa5506b1816e75eb379d06cfcd0af0
1e07a2eb813ae9b5d7405.svg">
...
</html>
```

- Browser constructs a DOM in case its an HTML document (Document Object Model)
- Browser discovers references to other resources in the html document like images, stylesheets, font etc.
- For each resource the browser sends separate HTTP requests to same or other servers to fetch that resource
- These requests can be parallel
- Once the browser has all the necessary resources, it renders the HTML document (or displays it)

HTML: Hyper Text Markup Language

HTML: Hypertext Markup Langauge
Hypertext: clickable link on web
Markup langauge: used for formatting of documents
Father of HTML: Tim-Berners Lee
First Version: 1991
Latest version: HTML 5
Extension: .html

Hypertext:

- Those text that contains hyperlinks (hyperlinks allows non sequential access of text).
- normal text does not contains hyperlinks so it does not allows non sequential access of text. Example:- Notepad text.

Markup language:

- Any tag based language is called markup language.
- Tag looks likes this:- <tag>

Why Html ?:

- To create web page

What is a webpage:

- Those page that are open through browser is called web-page.
- A Webpage is a document that is designed to be displayed in a web browser.
- It can contain various types of content, such as text, images, videos, interactive elements.
- Some of the browser sofwares are: **Google Chome, Internate Explorer, Mozila Firefox, Safari**, etc.

A Webpage can be classified into 2 categories:

1. Static Webpage
2. Dynamic Webpage

Difference between Static and Dynamic Webpage:

Static Page	Dynamic Page
In static web pages, Pages will remain same until someone changes it manually. In static web pages, Information is change rarely.	In dynamic web pages, Content of pages are different for different visitors. In dynamic web page, Information are change frequently.
Static Web Page takes less time for loading than dynamic web page.	Dynamic web page takes more time for loading.
Static web pages are written in languages such as: HTML, JavaScript, CSS, etc.	Dynamic web pages are written in languages such as: CGI, AJAX, ASP, ASP.NET, etc.
Static web pages does not contain any application program	Dynamic web pages contains application program for different services.
Static web pages require less work and cost in designing them. Static Web Pages are simple in terms of complexity.	Dynamic web pages require comparatively more work and cost in designing them. Dynamic web pages are complicated.
The static web pages generally do not use databases. They do not require data redecoration very often.	The dynamic web pages use databases because it involves frequent data and info redecoration.

Is HTML a programming language?

HTML is not a programming language - it's a markup language. Once we dive into a programming language like Javascript, you will see that they write logic that can be executed, but writing HTML is more like formatting your essay, Giving it a structure & marking it with a marker.

HTML document:

- It is a text file, saved with **.html** extension.
- We create structure of a webpage by using various HTML tags.
- browser software reads the html document and compose them as a web page.
- browser software does not display the tags but uses the tags to interpreate the content of the tag.

HTML provides number of tags to design the structure of a webpage.

Every tag is enclosed with angle bracket, example <tag>

HTML tags are of two types:

1. Pair-tags.
2. Non-pair tags.

Pair-tags:

- All the pair-tags are the combination of start tag and end tag.
  Example: <body> </body>
- They are like a container. We can place other elements within these tags.

Non-pair tag:

- These tags are standalone tags. They are self closing tags, they don't have closing tag.
  Example: <br/>, <input/> <img/>

Attributes:

- Every tag provides number of properties, we call them as an attribute.
- Each attribute is a combination of attribute-name and attribute-value.
  Example: ****

Here we are using 3 attributes inside the <img> tag:

1. src = "a1.jpg"
2. height = "100px"
3. width = "200px"
4. Alt = "image-name"

Note: attrib ute values should be in double quotes.

HTML Element:

- A tag and its contents together is called an attribute.
  Example:

- HTML is a case insensitive language.

- It is an error-less language, it does not display any error message, if any invalid tag we use, it simply ignores that.

First HTML page example:

index.html:

```
<!DOCTYPE html>
<html>
        <head>
                <title>My First Web Page</title>
        </head>
        <body>
                University
        </body>
</html>
```

Let us talk about these tags one by one:

<!DOCTYPE html>: Doctype declaration

- The Doctype tells the web browser which version of HTML the page is written in. This ensures that the browser renders the page correctly according to the rules of that HTML version.

<html>:

- The <html> tag encapsulates all the content of an HTML document. It tells the browser that everything inside it is HTML content.
-
- `The lang attribute`
- It's important for accessibility. Screen readers use this attribute to pronounce words correctly. It also helps search engines understand the primary language of the content.
  Example: **<html lang="en">**

<head>:

- The <head> section contains meta information, links to stylesheets, scripts, and other resources that are essential for the proper presentation and behavior of the web page. Inside the head tag.

<body>:

- The <body> tag contains everything that a user will see and interact with on the web page. This is where you put your content: text, images, videos, links, and more.

Text related tags:

HTML headings: HTML has total 6 levels; block; bold

h1 has highest font size biggest, h6 has highest font size smallest

<h1> </h1>
<h2> </h2>
<h3> </h3>
<h4> </h4>
<h5> </h5>
<h6> </h6>

An Example

```
<!DOCTYPE html>
<html>
	<head>
		<title>My First Web Page</title>
	</head>
	<body>
		<h1 align=”center”> University</h1>
		<h2> University 1</h2>
		<h3> University 2</h3>
		<h4> University 3</h4>
		<h5> University 4</h5>
		<h6> University 5</h6>
	</body>
</html>
```

Output

<p> tag: paragraph tag

- Block, default alignment: left, font size: 16px font-family: Times new roman.

An Example:

```
<!DOCTYPE html>
<html>
	<head>
		<title>My First Web Page</title>
	</head>
	<body>
		<p align=”left”>
			Lorem Ipsum is simply dummy text of the printing and typesetting
industry. Lorem Ipsum has been the industry's standard dummy text ever since the
1500s, when an unknown printer took a galley of type and scrambled it to make a type
```

```
specimen book. It has survived not only five centuries, but also the leap into
electronic typesetting, remaining essentially unchanged. It was popularised in the
1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more
recently with desktop publishing software like Aldus PageMaker including versions of
Lorem Ipsum.
		</p>

		<p align=”right”>
			Lorem Ipsum is simply dummy text of the printing and typesetting
industry. Lorem Ipsum has been the industry's standard dummy text ever since the
1500s, when an unknown printer took a galley of type and scrambled it to make a type
specimen book. It has survived not only five centuries, but also the leap into
electronic typesetting, remaining essentially unchanged. It was popularised in the
1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more
recently with desktop publishing software like Aldus PageMaker including versions of
Lorem Ipsum.
		</p>

		<p align=”center”>
			Lorem Ipsum is simply dummy text of the printing and typesetting
industry. Lorem Ipsum has been the industry's standard dummy text ever since the
1500s, when an unknown printer took a galley of type and scrambled it to make a type
specimen book. It has survived not only five centuries, but also the leap into
electronic typesetting, remaining essentially unchanged. It was popularised in the
1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more
recently with desktop publishing software like Aldus PageMaker including versions of
Lorem Ipsum.
		</p>
		<p align=”justify”>
			Lorem Ipsum is simply dummy text of the printing and typesetting
industry. Lorem Ipsum has been the industry's standard dummy text ever since the
1500s, when an unknown printer took a galley of type and scrambled it to make a type
specimen book. It has survived not only five centuries, but also the leap into
electronic typesetting, remaining essentially unchanged. It was popularised in the
1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more
recently with desktop publishing software like Aldus PageMaker including versions of
Lorem Ipsum.
		</p>
	</body>
</html>
```

Output:

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

**
:**

- insert a new line (line break)

<b>/:

- both display in bold; text in is considered important.

<i>/:

- both display in italic; text in is considered important.

<u>:

- underline

<mark>:

- used to highlight a text

<super>:

- used for superscript

<sub>:

- used for subscript

<ins>:

- inserted text; display: underline

<del>:

- for text to be deleted

<hr/ >: It is a non-pair tag

- Horizontal line
- **Attribute:**
    - color
    - size

Special symbol in HTML:

- ** ** space
- **<** <
- **>** >
- **&**
- **©** copyright
- **®** registration

An Example:

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

```
        <title>My First Web Page</title>
    </head>
    <body>
        <strong>Lorem Ipsum is simply dummy text</strong> of the printing and
        typesetting industry. Lorem Ipsum has been the industry's standard dummy
        text ever since the 1500s, when an unknown printer took a galley of type
        and scrambled it to make a type specimen book. It has survived not only
        five centuries, but also the leap into electronic typesetting, remaining
        essentially unchanged. It was popularised in the 1960s with the release
        of Letraset sheets containing Lorem Ipsum passages, and more recently
        with desktop publishing software like <strong><em>Aldus PageMaker
        including versions of Lorem Ipsum.</em></strong>
                            <br />
                            <hr />
                            <em>Lorem Ipsum is simply dummy text</em> of the
        printing and typesetting industry. Lorem Ipsum has been <ins>the
        industry's standard dummy text ever</ins> since the 1500s, when an
        unknown printer took a galley of type and scrambled it to make a type
        specimen book. It has survived <del>not only five centuries</del>, but
        also the leap into electronic typesetting, remaining essentially
        unchanged. It was popularised in the <mark>1960s</mark> with the release
        of Letraset sheets containing Lorem Ipsum passages, and more recently
        with desktop publishing software like <u><em><strong>Aldus PageMaker
        including versions of Lorem Ipsum.</strong></em></u>
            <br /><br />
            <hr />
            (a + b)<sup>2</sup> = a<sup>2</sup> + 2ab + b<sup>2</sup>
        <br />
            The chemical formula of water is H<sub>2</sub>O
        <br />
            This is after 4    spaces;
        <br />
            This is &lt;h1&gt; tag
        <br />
    </body>
</html>
```

Output:

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like ***Aldus PageMaker including versions of Lorem Ipsum.***

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived ~~not only five centuries~~, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like ***Aldus PageMaker including versions of Lorem Ipsum.***

$(a + b)^2 = a^2 + 2ab + b^2$
The chemical formula of water is H_2O
This is after 4 spaces;
This is <h1> tag

Advance Text formatting tags:

<dl><dt><dd>:

- **<dl>:** The Description List element
- **<dt>:** The Description Term/Title
- **<dd>:** The Description details

Example:

```
<dl>
  <dt>Coffee</dt>
       <dd>Black hot drink</dd>
  <dt>Milk</dt>
       <dd>White cold drink</dd>
</dl>
```

Output:

Coffee
    Black hot drink
Milk
    White cold drink

<cite>:

Usage: The `<cite>` element in HTML is used to reference the title of a creative work. This can be a book, a play, a movie, a painting, a sculpture, a song, or any other work that has a distinct title.

Example:

My favorite book is <cite>The Compound Effect</cite> by Darren Hardy

Display: By default, browsers usually render `<cite>` content in italics.

Semantic importance: Using semantic elements like `<cite>` in your HTML helps make your content more understandable for search engines, assistive technologies (like screen readers for the visually impaired), and other machines or tools that parse web content. This can aid in accessibility and SEO (search engine optimization).

<dfn>:

Usage: Represents the defining instance of a term.

Example:

A <dfn>computer</dfn> is a device that can be instructed to carry out sequences of arithmetic or logical operations automatically.

Purpose: The `<dfn>` tag is used to highlight the introduction or first instance of a term in a context where it will be defined.

Display: By default, most browsers render the `<dfn>` content in italics.

Attributes:

`title` : This attribute is commonly used with **`<dfn>`** to provide a brief definition of the term.

Example:

The <dfn title="A mythical creature often depicted as a large, scaly, fire-breathing serpent.">dragon</dfn> is a common figure in medieval folklore.

<bdo>: Bi-directional Override

Usage: Overrides the current text direction.

Attribute:

- **ltr:** default direction
- **rtl:** right direction

Example:

<bdo dir="rtl">Welcome to chitkara</bdo> text.

Usage Scenario: The `<bdo>` element can be handy when trying to display characters in a way that's against their natural or default directionality. For instance, if you have an Arabic word (which is inherently RTL) that you want to display in an LTR context for artistic or specific layout purposes.

It is used to mirror the text.

<bdi>: Bi-directional isolate

Usage: Isolates a span of text that might be formatted in a different direction from other text outside it.

Example: Useful for user-generated content where the text direction isn't known in advance.

User: <bdi>نص</bdi> has logged in.

Purpose: The main purpose of the `<bdi>` tag is to allow a piece of text to be isolated from its surrounding text, ensuring it's displayed in its intended direction even if it's different from the rest of the content. This is especially useful for user-generated content where the text direction isn't known in advance.

Usage Scenario: Consider social media posts, comments, or usernames where users from different language backgrounds contribute content. Without `<bdi>`, a username in Arabic or Hebrew (which are right-to-left languages) might distort the layout of English content (which is left-to-right) or vice versa.

Example:

```
<ul>
     <li><bdi>Albert</bdi>: 10 posts</li>
     <li><bdi>سليم</bdi>: 5 posts</li>
</ul>
```

<abbr>: The abbreviation element

- The <abbr> tag defines an abbreviation or an acronym, like "HTML", "CSS", "Mr.", "Dr.", "ASAP", "ATM".

  Attribute: title

  Example:

```
The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.
```

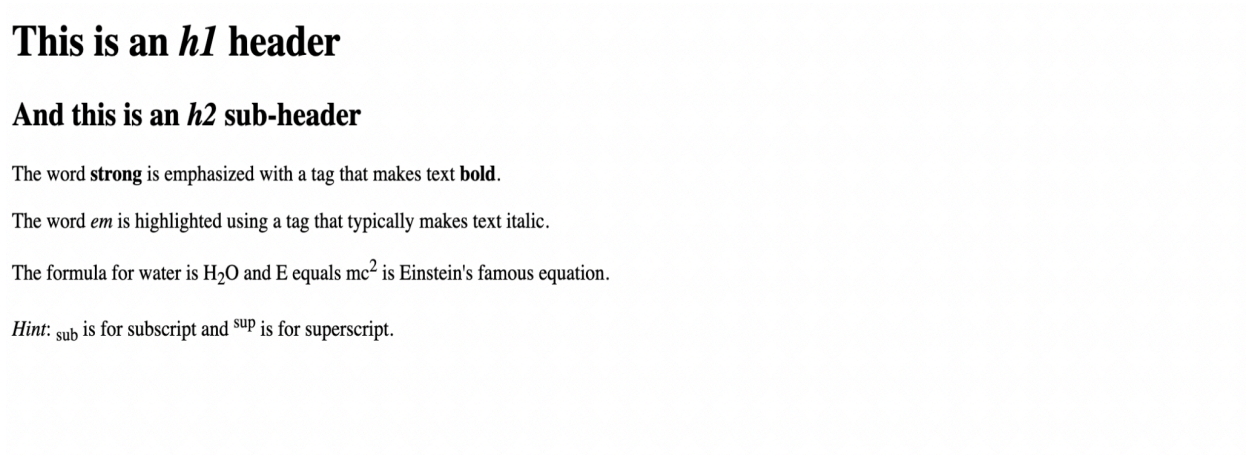
<pre>: Preformated text

- Text in a <pre> element is displayed in a fixed-width font, and the text preserves both spaces and line breaks.

  Example:

```
<pre>
       function fun1(){
              console.log("Welcome to SAGE");
       }
<pre>
```

Assignment1:



Working with lists:

1. Ordered List (<ol>)

Usage: The <ol> tag is for ordered list it has block display

We need to make use of **<li>** tag as a sub tag to represent each items.

Example:

```
<ol>
      <li>Delhi</li>
      <li>Mumbai</li>
      <li>Kolkata</li>
      <li>Chennai</li>
</ol>
```

Attributes:

1. **type:** is used to define number format 1 (default value), 'A', 'a', i, I
2. **start:** is used define by what number to start; value should be numeric

2. Unordered List (<ul>)

Usage: The <ul> tag is for ordered list it has block display

The **type** attribute is used to define number format **disc** (default value), **square** & **circle**

Example:

```
<!DOCTYPE html>

<html>
	<head>
		<title>My First Web Page</title>
	</head>
	<body>
			<ol type='A' start='4'>
				<li>Open website Sageuniersity.com</li>
				<li>Make login</li>
				<li>Search for permission letter</li>
				<li>Enter roll number and hit enter</li>
				<li>Download permission letter in PDF
format</li>
			</ol>

			<hr />

			<ul type='square'>
				<li>be on time</li>
				<li>Do not cheat</li>
				<li>take lunch before coming to exam hall</li>
				<li>keep stationary with you</li>
			</ul>

	</body>

</html>
```

Output:

D. Open website Sageuniersity.com
E. Make login
F. Search for permission letter
G. Enter roll number and hit enter
H. Download permission letter in PDF format

- be on time
- Do not cheat
- take lunch before coming to exam hall
- keep stationary with you

Working with links:

In the vast world of the internet, imagine every website as an island. These islands are unique, rich with content, and offer different experiences. But what if these islands were isolated? How would you travel

between them? This is where the HTML `<a>` tag, commonly known as the anchor element, comes into play. Think of it as the bridge that connects these islands.

The `<a>` tag is used to create links, allowing users to navigate from one page to another, be it within the same website or to a completely different one.

- It is a pair tag : <a> : Inline element
- Unvisited link are displayed in blue color but the visited link are displayed in the purple color.

Attributes:

1. **href** (hypertext reference): It contains the URL to be opened once link is clicked.
2. **title**: contains text for the tooltip
3. **target**: where to open the link. The possible values are as follow:

    a) **_self**: default value; means link will be opened in the same tab
    b) **_blank**: means link will be opened in new tab

Types of links:

1. **External Link:** The link that takes you to another website. Always we have mention absolute URL in the href.
2. **Internal Link:** The link that takes you to another page of same website. This is recommended to use relative URL.
3. **Internal document reference:** The link that take you some section of same web page. The id attribute is used to identify an element in a web page such that id of page must be unique across the page.

```
<a href="#id-of-element-to-jump">display text</a>
```

Absolute URL & Relative URL:

Absolute URL:

- It's a full web address. If you think of it as an address, it's similar to giving someone your full home address including your city, state, and country.

- It start with protocol; this URL can be used anywhere no matter whether we are creating link in the same website or other website
  e.g. https://www.amazon.in/some-product

Relative URL:

- These are partial addresses that are relative to the current page. Imagine giving someone directions to your kitchen, while they're already in your living room.

- A relative URL is always given with respect to location of file which contains the link. Consider the following folder structure

```
E-commerce project folder (http://ecommerceproject.com/)
```

```
                |
-----------------------------------
|           |     |      |        |
|           |     |      |        |
CSS        JS  admin    user     images
style.css  a.js A.html  B.html   sun.jpg
```

Situation-01 (using absolute URL)::In A.html we want to create link for B.html

```
<a href='http://ecommerceproject.com/user/B.html'>Click here</a>
```

Situation-02 (using relative URL)::In B.html we want to create link for A.html

```
<a href="../admin/A.html">Click Here</a>
```

More Than Just Web Pages:

Links can point to various resources:

- Images: `<a href="images/sunset.jpg">View Sunset Image</a>`
- PDFs: `<a href="files/report.pdf">View Report</`

You can even prompt a user to download a resource directly:

Click to download the sunset image

Navigating Within a Page:

a) Fragments: Let's say a webpage is a multi-story building. If you want to guide someone to a specific floor or room, you'd use fragments.

```
<h2 id="summary">Summary</h2>

<a href="#summary">Jump to Summary</a>
```

b) Jump to Top Link: If you're at the base of this building and want a quick elevator to the top,

```
<a href="#">Back to Top</a>
```

Extra Tips for Enhancing User Experience:

Mailto Links: Want your users to send you an email directly by clicking a link? Here you go:

```
<a href="mailto:example@example.com">Send us an email</a>
```

Title Attribute: Provide a brief tooltip description when hovering over the link:

Example

Example1:

LinkOne.html:

```
<!DOCTYPE html>
<html>
	<head>
		<title>Linking Example</title>
	</head>

	<body>

	  <!-- this is an external link -->

	   <a href="https://www.google.com/">click to search</a>

	   <hr />

	   <!-- this is internal link -->

	   <a href="LinkTwo.html">Click to see another page</a>

	   <hr />

	    <a href="#para2">Click to jump on second para</a> | <a href="#para3">Click to jump on third para</a>

    <hr />

    <h3>The first paragraph</h3>
    <p>
      Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum
    </p>
```

```
 <h3 id="para2">The second paragraph</h3>
    <p>
      Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum
    </p>

 <h3 id="para3">The third paragraph</h3>
    <p>
      Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum
    </p>

<a href="#">Back to top</a>

</body>

</html>
```

LinkTwo.html

```
<!DOCTYPE html>
<html>
        <head>
                <title>Linking Example</title>
        </head>
        <body>
           <h1>This is dummy text</h1>
        </body>
```

```
</html>
```

Example2:

1. **Home Page:** Introduction and links to other pages.
2. **Books Page:** Displaying a list of books.
3. **Contact Page:** A page with contact information

1. Home Page (`index.html`):

```
<!DOCTYPE html>
<html>

  <head>
    <title>Bookstore Home</title>
  </head>

  <body>
    <h1>Welcome to Our Online Bookstore</h1>

    <ul>
      <li><a href="index.html">Home</a></li>
      <li><a href="books.html">Books</a></li>
      <li><a href="contact.html">Contact</a></li>
    </ul>

    <p>
      Discover a wide range of books in various genres. Dive into the world of
      literature with us.
    </p>
  </body>
</html>
```

2. Books Page (`books.html`):

```
<!DOCTYPE html>
<html>
  <head>
    <title>Our Books</title>
  </head>
  <body>
```

```
    <h1>Our Books</h1>

    <ul>
      <li><a href="index.html">Home</a></li>
      <li><a href="books.html">Books</a></li>
      <li><a href="contact.html">Contact</a></li>
    </ul>

    <ul>
      <li>"The Great Gatsby" by F. Scott Fitzgerald</li>
      <li>"Pride and Prejudice" by Jane Austen</li>
      <li>"To Kill a Mockingbird" by Harper Lee</li>
    </ul>
  </body>
</html>
```

3. Contact Page (`contact.html`):

```
<!DOCTYPE html>

<html>
  <head>
    <title>Contact Us</title>
  </head>
  <body>
    <h1>Contact Information</h1>
    <ul>
      <li><a href="index.html">Home</a></li>
      <li><a href="books.html">Books</a></li>
      <li><a href="contact.html">Contact</a></li>
    </ul>

    <p>Email: <a href="mailto:info@bookstore.com">info@bookstore.com</a></p>
    <p>Phone: +123 456 7890</p>
  </body>
</html>
```

The <img /> tag:

- Used to insert the images. It's an empty tag with inline display.
- **Attributes:**
  - **src**: contains URL of the image. It may contains absolute of relative URL
  - **title**: text for the tooltip
  - **width**: width of the image
  - **height**: height of the image

- **alt**: alternative text to be displayed if the image is not available
- **loading**: This is a newer attribute that can be set to lazy or eager. If set to lazy, it indicates to the browser that the image can be loaded lazily (e.g., when it's near the viewport). This can be very useful for performance, especially with large images below the fold.

Attach sunset image in the b.html file using absolute URL:

- Copy the image link from the internate

```
<img
src="https://img.freepik.com/free-photo/sunset-time-tropical-beach-sea-with-coconut-palm-tree_74190-1075.jpg" width='100' height='100' alt=”Sunset image” />
```

Attach sunset.jpg in the a.html file using relative URL

- Download the image into the computer and put it inside the images folder (create images folder inside the current workspace folder).

```
<img src="./images/sunset.jpg" width='100' height='100' alt=”Sunset image” />
```

The <audio> & <video> tags:

- The introduction of the `<audio>` and `<video>` elements in HTML5 was a significant move towards standardizing media playback across the web, reducing the need for plugins like Flash. Let's delve into each element:

`<audio>` tag:

- The `<audio>` element is used to embed sound content in documents.

Attributes:

- **src**: The URL of the audio to embed.
- **controls**: If present, the browser will offer playback controls.
- **autoplay**: The audio will start playing as soon as it's ready.
- **loop**: The audio will start over again every time it finishes.
- **muted**: Specifies that the audio output should be muted.
- **preload**: Hints to the browser about how the audio should be loaded when the page loads (`none`, `metadata`, or `auto`).

Example:

```
<audio controls>

      <source src="sound.mp3" type="audio/mpeg">
      <source src="sound.ogg" type="audio/ogg">
```

```
    Your browser does not support the audio element.

</audio>
```

Here, we provide both an MP3 and an Ogg file for the audio to ensure it plays on as many browsers as possible. The message "Your browser does not support the audio element." serves as a fallback for browsers that don't support the <audio> element.

<video> tag:

- The <video> element is used to embed video content.

Attributes (most are similar to <audio>):

- **src**: The URL of the audio to embed.
- **controls**: If present, the browser will offer playback controls.
- **autoplay**: The audio will start playing as soon as it's ready.
- **loop**: The video will start over again, every time it finishes.
- **muted**: Specifies that the audio output should be muted.
- **preload**: Hints to the browser about how the audio should be loaded when the page loads (none, metadata, or auto)
- **poster**: An image to show as a placeholder before the video plays.
- **width** and **height**: Set the size of the video player.

Example:

```
<video width="320" height="240" controls>

    <source src="movie.mp4" type="video/mp4">
    <source src="movie.ogg" type="video/ogg">
    Your browser does not support the video tag.

</video>
```

Fallbacks & Support:

1. Multiple Formats: Always provide multiple formats for your media to accommodate different browsers. For example, MP3 and Ogg for audio, MP4 and Ogg for video.
2. Text Fallback: Inside the <audio> or <video> tag, include a message for browsers that don't support these tags.

3. JavaScript Fallback: For more advanced fallbacks, you can use JavaScript to detect whether the browser supports certain media formats and react accordingly.
4. Content Delivery Networks (CDNs): Some CDNs and services offer video players that come with fallbacks built-in, automatically serving the right format based on the user's browser.
5. Subtitles and Captions: The <track> element can be used within the <video> tag to add subtitles or captions.

<iframe> tag:

- The <iframe> HTML element represents a nested browsing context, embedding another HTML page into the current one.

  Example:

Assignment: Embedding Youtube video

Objective: Embed a YouTube video into your web page.

Instructions:

1. Create the standard HTML document structure.
2. Set the title of the page to "Embedded Video".
3. Inside the body, use the `<h1>` tag to display "My Favorite Video".
4. Below the title, write a short paragraph explaining why you like the video.
5. Now, go to YouTube, find your favorite video, click on the "Share" button below the video, and select "Embed". This will give you an `<iframe>` code.
6. Copy the `<iframe>` code and paste it into your HTML document below your paragraph.

Hints:

- The `<iframe>` code from YouTube should be used as-is without any modifications.

HTML table: <table> tag

An HTML table is used to represent data in a tabular format, i.e., in rows and columns. This structure can be immensely useful for displaying information in a structured manner, like financial statements, calendars, and various kinds of lists.

<table> :

- It is a block level element.
- It is a paired tag

Attributes:

- **width** (in px, in %, default: auto means content-fit)
- **border** (default: 0 means no border, numerical value)
- **cellpadding**: define gapping between cell border and content
- **cellspacing**: define spacing between two consecutive cells, numerical

`<thead>:`

- This element wraps the set of rows that describe the table headers. Although not strictly required, it's useful for differentiating the header row(s) from the body rows.

`<tbody>:`

- This element wraps the set of rows that describe the actual data in the table. It is also not strictly required, but it's a good semantic element to differentiate the main content of the table from the headers and footers.

`<tfoot>:`

- This element wraps the set of rows that describe the table footers. It can be useful for displaying summaries, totals, or footnotes for the table's data.

`<tr>:`

- This element represents a table row. It wraps a set of table cells, which can be either header cells (<th>) or standard cells (<td>).

`<th>:`

- This element represents a header cell. By default, its content is bold and centered.

<td>:

- This element represents a standard table cell.

`<caption>:`

- used to create heading of table which will be displayed at the top of table (outside row)

Example1:

```html
<table border="1">
    <!-- Table Header -->
    <thead>
        <tr>
            <th>Month</th>
            <th>Product A Sales ($)</th>
            <th>Product B Sales ($)</th>
            <th>Total Sales ($)</th>
        </tr>
    </thead>

    <!-- Table Body -->
    <tbody>
        <tr>
            <td>January</td>
            <td>1200</td>
            <td>800</td>
            <td>2000</td>
        </tr>
        <tr>
            <td>February</td>
            <td>1300</td>
            <td>850</td>
            <td>2150</td>
        </tr>
        <tr>
            <td>March</td>
            <td>1250</td>
            <td>900</td>
            <td>2150</td>
        </tr>
    </tbody>

    <!-- Table Footer -->
    <tfoot>
        <tr>
            <td>Total</td>
            <td>3750</td>
            <td>2550</td>
            <td>6300</td>
        </tr>
```

```
    </tfoot>
</table>
```

rowspan and colspan attribute of <td> or <th> tag:

- rowspan and colspan both are the attribute of the <th> or <td> element.
- They provide the same functionality as "merge cell" in MS-excel.

colspan: it allows to span the width of the cell to more than one cell or column.

rowspan:- it allows to span the height of the cell of more than one cell or row.

Example:

```
<table align="center" border="2">

   <caption><strong>Saving Table</strong></caption>


    <tr>
        <td colspan="3" align="center">Welcome</td>
    </tr>

    <tr>
        <th>Month</th>
        <th>Saving</th>
        <th>Saving for Holidays</th>
    </tr>

    <tr>
      <td>january</td>
      <td>500</td>
      <td rowspan="2">1000</td>
    </tr>

    <tr>
      <td>february</td>
      <td>400</td>
    </tr>
```

```
</table>
```

Note: We can put 1st and 2nd <tr> tag inside the <thead> tag and remaining <tr> tags inside the <tbody> tag and inplace of <td> tag we can make use <th> tag inside the <thead> tag in above example.

The <meta> tag:

Meta tags are vital components within the <head> section of an HTML document. They provide metadata about the document's content or give instructions to browsers on how to display it. While not directly visible on the page, meta tags can influence how web pages are described and displayed in search results.

Here are some of the most common meta tags and their purposes:

`Charset (<meta charset="UTF-8">):`

- **Description:** Specifies the character encoding for the HTML document.
- **Importance:** It ensures characters in the document are rendered correctly, especially for languages with special characters.

`Viewport (<meta name="viewport" content="width=device-width, initial-scale=1">):`

- **Description:** Defines how a page should appear on mobile devices and is crucial for responsive design.
- **Importance:** It ensures the website scales and sizes properly on all devices, especially mobile ones.

`Description (<meta name="description" content="A brief summary of the webpage's content">):`

- **Description:** Provides a short and concise summary of the webpage's content.
- **Importance:** Search engines often display this description in search results. A good description can entice users to click on your link.

`Keywords (<meta name="keywords" content="keyword1, keyword2, ...">):`

- **Description:** Originally used to tell search engines what keywords the page should rank for.
- **Importance:** Today, this tag holds less weight in search engine ranking algorithms due to over-optimization and keyword stuffing in the past. Some argue its relevance is minimal now.

`Author (<meta name="author" content="Name of the author">):`

- **Description:** Specifies the author of the webpage.
- **Importance:** Lets readers know who created the content, though it's not widely used for SEO purposes.

`Refresh (<meta http-equiv="refresh" content="5;url=https://example.com/">):`

- **Description:** Automatically refreshes the page or redirects to another after a specified number of seconds.
- **Importance:** Can be useful in certain situations like after a form submission or for timed redirects, but overuse can degrade user experience.

`Social Media Tags (Open Graph, Twitter Cards, etc.):`

- **Description:** These are specific meta tags used to define how content appears when shared on social platforms like Facebook or Twitter. They might define a preview image, title, or description.
- **Importance:** Ensures that when users share your content on social media, it's presented in an attractive and consistent manner.
  Just google it

These are just some of the meta tags available. Depending on your website's needs, you might use additional meta tags or skip some of the ones mentioned above. However, meta tags as a whole play a crucial role in influencing how a website interacts with browsers, search engines, and users.

<details> and <summery> tag:

- The <summary> tag defines a visible heading for the <details> element. The heading can be clicked to view/hide the details.

  Note: The <summary> element should be the first child element of the <details> element.

  Example:

```
<details>
    <summary>Click to expand/collapse</summary>
    <p>This is the hidden content that can be toggled.</p>
    <p>This is the hidden content that can be toggled.</p>
```

```
        <p>This is the hidden content that can be toggled.</p>
        <p>This is the hidden content that can be toggled.</p>
        <p>This is the hidden content that can be toggled.</p>
    </details>
```

Forms:

Forms are a fundamental aspect of web development and are used to gather information from users. An HTML form is defined using the `<form>` element. Inside the form, you can have various types of input elements to collect different kinds of data.

<form> tag:

- It is a paired tag and block level element.

  Attributes:

  1. **name**: name of the form
  2. **action**: where to take form data after submission
  3. **method**: define form submission method i.e. **get** or **post**

Here are some commonly used form elements:

1. `<input>`

The `<input>` element is the most versatile form control. It can be configured in various ways depending on its `type` attribute. Some popular input types include:

- **`text`: For single-line text input.**
- **`password`: Similar to `text` but hides the characters as they're typed.**
- **`radio`: Lets users select one option among a set.**
- **`checkbox`: Lets users select zero or more options of a limited number of choices.**
- **`submit`: A button to submit the form.**
- **`reset`: A button to reset the form to its initial state.**
- **`file`: For file uploads.**
- **`hidden`: A hidden data field, which doesn't show to the user but can store data to be sent with the form.**

- `date`, `time`, `datetime-local`: For date and time input.
- `email`: For email input. It can validate if the inputted text is in the format of an email address.
- `url`: For URL input. Validates the input format as a URL.
- `number`: Input for numbers. You can set minimum, maximum, and step values.

Attribute	Description
placeholder	Sets a hint text that will be displayed in the input field until the user starts typing. In this case, the placeholder is "CHI5009".
type	Specifies the type of input field. The value "text" indicates a standard text input field.
id	Assigns a unique identifier to the input element. This can be used to reference the element in JavaScript or CSS. In this case, the ID is "name".
name	Specifies the name of the input element. This is often used to identify the input field when submitting a form. In this case, the name is "name".
required	Indicates that the input field must be filled in before the form can be submitted. If the user tries to submit the form without entering a value, an error message will be displayed.
autocapitalize	Controls how the input field handles capitalization. The value "characters" indicates that only the first character of each word should be capitalized.
autofocus	Automatically focuses the input field when the page loads, placing the cursor inside it.
minlength	Sets the minimum number of characters allowed in the input field. In this case, the minimum length is 3.
maxlength	Sets the maximum number of characters allowed in the input field. In this case, the maximum length is 50.

2. `<textarea>`

Allows multi-line text input.

3. `<select>`

Used to create a drop-down list. The options within the list are defined using `<option>` elements.

4. `<button>`

A clickable button.

5. `<label>`

Provides a textual description for a form control. It helps in improving accessibility. When a user clicks on a label, it gives focus to the form element it is associated with.

The `for` attribute in a `<label>` tag establishes a connection between the label and an associated form element, typically an `<input>` element. It allows users to click on the label to interact with the corresponding form element.

Here's how it works:

1. **Assign the same value to the `for` attribute in the `<label>` tag and the `id` attribute in the associated `<input>` element.**
2. **When a user clicks on the label, the browser automatically focuses on the `<input>` element with the matching `id`.**

Example:

HTML

```
<label for="username">Username:</label>

<input type="text" id="username">
```

Example:

```
    <form><!-- Name field -->
    <label for="fullname">Full Name:</label>
    <input type="text" id="fullname" name="fullname" required>


    <br/><br/>

    <!-- Email field -->
    <label for="email">Email:</label>
    <input type="email" id="email" name="email" required>


    <br/><br/>
```

```
    <!-- Password field -->
    <label for="password">Password:</label>
    <input type="password" id="password" name="password" required>


    <br/><br/>

    <!-- Gender selection using radio buttons -->
    <label>Gender:</label>

    <label for="male">Male</label>
    <input type="radio" id="male" name="gender" value="male">

    <label for="female">Female</label>
    <input type="radio" id="female" name="gender" value="female">

    <br/><br/>


    <!-- Dropdown for choosing a department -->
    <label for="city">City:</label>
    <select id="city" name="city">
        <option value="">Select City</option>
        <option value="Indore">Indore</option>
        <option value="Bhopal">Bhopal</option>
        <option value="Jabalpur">Jabalpur</option>
    </select>

    <br/><br/>

    <!-- Textarea for additional comments -->
    <label for="comments">Comments:</label>
    <textarea id="comments" name="comments" rows="4" cols="50"></textarea>

    <br/><br/>



    <!-- Checkbox to multiple courses -->
    <label>Select Courses: </label>

    <input type="checkbox" id="java" name="courses" value="Java">
    <label for="java">Java</label>

    <input type="checkbox" id="node" name="courses" value="Node">
    <label for="node">Node</label>

    <input type="checkbox" id="react" name="courses" value="React">
    <label for="react">React</label>

    <input type="checkbox" id="ds" name="courses" value="DS">
    <label for="ds">Data-Structure</label>
```

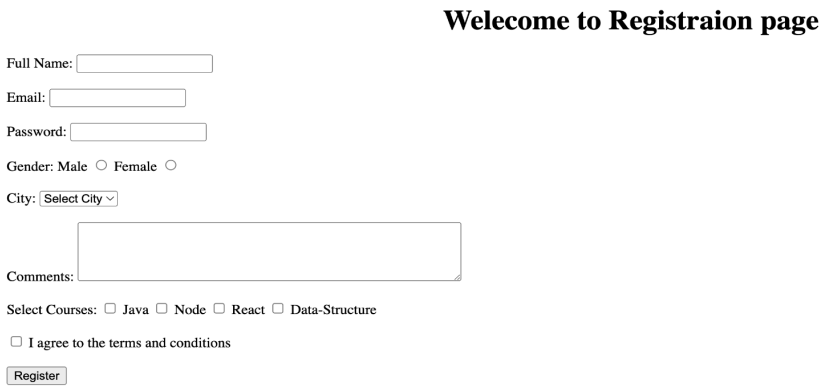
```
    <br/><br/>


    <!-- Checkbox to agree to terms -->
    <input type="checkbox" id="terms" name="terms" value="agree" required>
    <label for="terms">I agree to the terms and conditions</label>


    <br/><br/>

    <!-- Submit button -->
    <input type="submit" value="Register">
</form>
```

Output:



<fieldset> and <legend> tag:

- The `<fieldset>` tag: is used to create a box around an element, it is used to group related elements in a form.
- The `<legend>` tag: used to add text/caption as heading of the box.

Example:

```
<!DOCTYPE html>

<html>
  <head>
    <title>The Picture Example</title>
  </head>


  <body>

         <h1 align="center">Welecome to Login page</h1>


         <fieldset>

           <legend>Login Form</legend>

           <form>

              <table cellpadding="3" cellspacing="3">
                <tr>
                   <td><label for="username">Username:</label></td>
                   <td><input type="text" name="username" id="username" /></td>
                </tr>

                <tr>
                  <td><label for="password">Password:</label></td>
                  <td><input type="password" name="password" id="password"/></td>
                </tr>

                <tr>
                   <td><input type='reset' value='Reset Form' /></td>
                   <td><input type='submit' value='Submit Form' /></td>
                </tr>

         </table>

    </form>

  </fieldset>

  </body>
</html>
```

Output:

Welecome to Login page

Login Form

Username:

Password:

Reset Form Submit Form

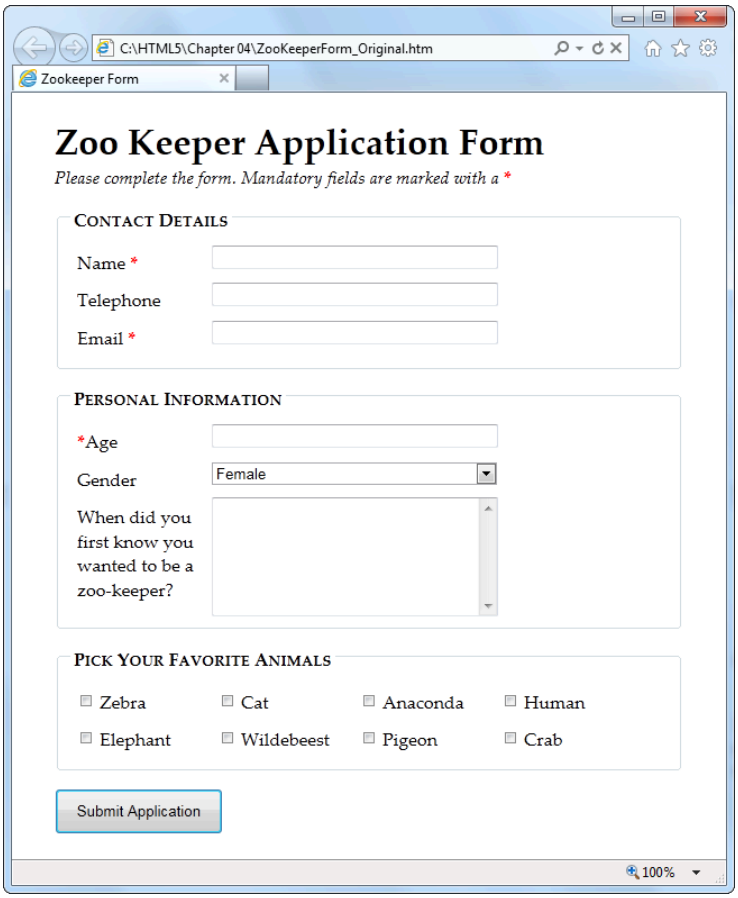
Assignment-01:

Create an HTML file (e.g. first_page.html) that specifies a page that contains a heading and two paragraphs of text. Use the HTML tags <h1>, </h1>, and in this exercise. As the texts in the heading and paragraphs you can use any texts you like.

- Add an unordered list to this web page. An unordered list should look like the following when it is shown by a browser:
  - An unordered list can be specified with the tags <ul> and .
  - An unordered list typically contains a number of list items that can be specified with tags <li> and .
  - After you have created your unordered list, check out what happens when you convert it to an ordered list by replacing the tags <ul> and with <ol> and , respectively.

- Add an image to your web page. In this exercise you must use the <img> tag. As an image, you can use any .jpg or .png file you find on the Internet.

Assignment-02:



HTML5 Structural Elements:

- There are a group of HTML elements that are created to broadly give our html documents more structure.
- These structural elements are categorized into 2 categories:
    1. **Non-Semantic** ,
    2. **Semantic** <header>, <nav>, <main>, <artical>, <section>, <aside>, <footer>

Structural elements (non-semantic)

In HTML, while many elements carry semantic meaning (e.g., `<header>`, `<article>`, `<nav>`), some elements are primarily used for structuring and styling content without adding specific meaning. Two of the most common non-semantic elements are `<div>` and `<span>`.

`<div>` (Division Element):

1. **Description:** The `<div>` is a block-level container element. It's often used to group content or other elements together for styling (with CSS) or scripting (with JavaScript) purposes.
2. **Use Cases:**
    - Wrapping sections of a page for styling: For instance, centering a block of content or applying a background color to a specific section.
    - As a container for layout purposes: For instance, creating columns or grid layouts using CSS.
    - JavaScript interaction: It's often easier to manipulate or animate a group of elements if they're contained within a `<div>`.

Example:

```
<div style="background-color: green; height: 200px; width: 200px;">
    <h1>Title</h1>
    <p>Some text here.</p>
</div>
```

`<span>` (Span Element):

1. **Description:** The `<span>` is an inline-level container element. It is used to target inline content or elements for styling or scripting without interrupting the flow of the content.
2. **Use Cases:**
    - Styling specific parts of text: For instance, changing the color or background of specific words within a paragraph.
    - JavaScript interaction: Such as adding tooltips to specific pieces of text.
    - Wrapping text for specific behaviors: Such as preventing line breaks in a specific span of text.

Example:

```
<p>This is <span style="color: red;">highlighted</span> text.</p>
```

Key Differences:

Display Value: By default, `<div>` has a display value of `block`, which means it takes up the full width of its parent container and starts on a new line in the flow of content. `<span>`, on the other hand, has a display value of `inline`, so it does not start on a new line and only takes up as much width as necessary.
Typical Use: `<div>` is typically used for grouping larger blocks of content or elements, while `<span>` is used for smaller, inline elements or pieces of text.

Importance of Using Them Appropriately:

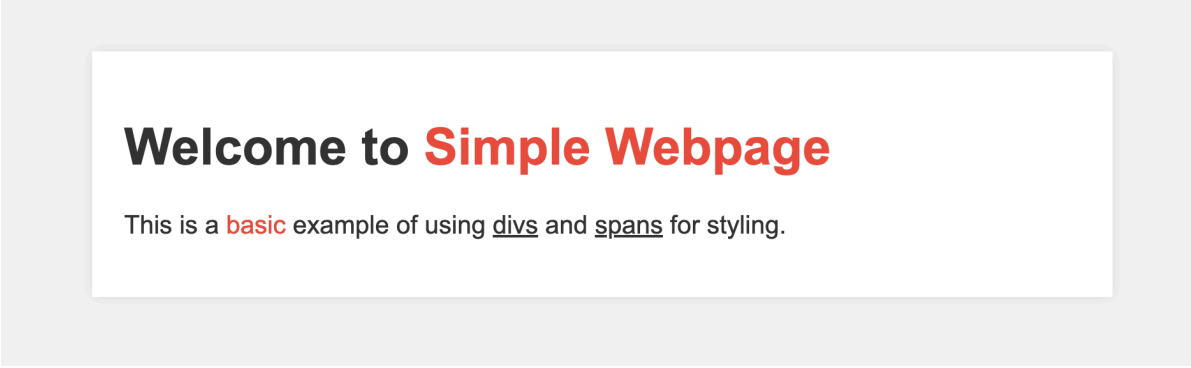
While `<div>` and `<span>` are non-semantic and don't provide inherent meaning by themselves, using them judiciously is essential:

1. **Accessibility**: Over-reliance on non-semantic elements can make a webpage less accessible. Screen readers and assistive technologies rely on the semantic meaning of elements to provide a better experience. Always prefer semantic elements when possible.
2. **Maintainability**: Structured, well-organized code is easier to read, update, and maintain. Using non-semantic elements indiscriminately can result in "div soup" or "span spaghetti", making the code harder to understand.
3. **Styling & Behavior**: Both `<div>` and `<span>` are immensely useful when semantic elements don't fit the need. They offer flexibility in applying styles and behaviors where semantic tags might not be appropriate or available.

In conclusion, while `<div>` and `<span>` are non-semantic, they are invaluable tools in web design and development. It's essential to balance their use with semantic elements to create accessible, maintainable, and functional web content.

Student Activity:

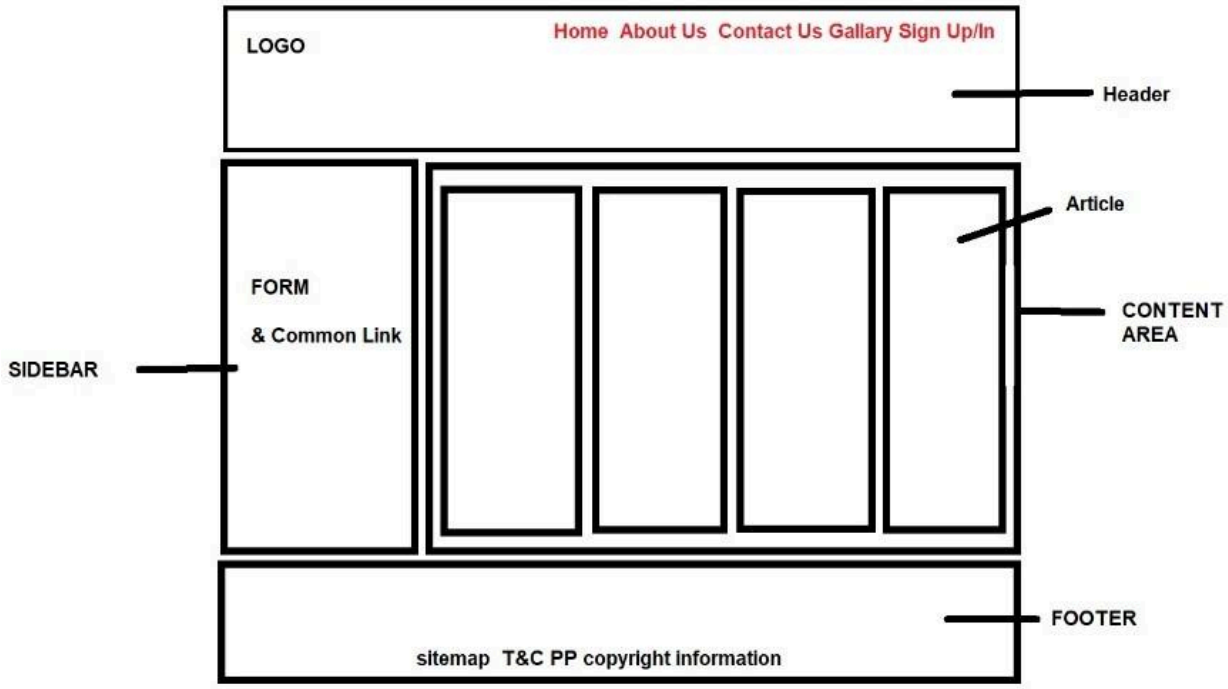
End Result:



Structural elements (semantic)

Semantic HTML elements are those that introduce meaning to web content. Unlike non-semantic elements like `<div>` and `<span>`, which don't convey specific meanings on their own, semantic elements clearly define their content for both the browser and the developer.

Using semantic elements not only makes the code more readable and maintainable but also benefits accessibility tools like screen readers and search engine optimization (SEO) efforts.



Here are some commonly used semantic HTML structural elements:

1. `<header>`

- **Description:** Represents a container for introductory content or navigation links. Often contains the website logo, website's title, and main navigation.

  Example:

```
<header>
        <h1>Website Name</h1>
        <nav>
                <!-- navigation links -->
```

```
            </nav>
        </header>
```

2. <nav>

- **Description:** Denotes a section with navigation links, either within the current document or to other documents.

  Example:

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

3. <main>

- Description: Represents the main content of a document. There should be only one <main> per page, and it should be unique from content that is repeated across a set of documents such as site navigation, header, or footer.

  Example:

```
<main>
  <article>
    <!-- main content -->
  </article>
</main>
```

4. <article>

- **Description:** Represents self-contained content that makes sense on its own and can be independently distributed or syndicated.

  Example:

```
<article>
  <h2>Blog Post Title</h2>
  <p>Blog post content...</p>
</article>
```

5. `<section>`

- **Description:** Represents a standalone section of a document, like tabs of content or a set of content grouped under a theme.

  Example:

```
<section>
  <h2>Section Title</h2>
  <p>Section content...</p>
</section>
```

6. `<aside>`

- **Description:** Used for content that is tangentially related to the main content and can be considered separate. Often seen as sidebars in a design.

  Example:

```
<aside>
  <h3>Related Links</h3>
  <ul>
    <!-- list of links -->
  </ul>
</aside>
```

7. `<footer>`

- **Description:** Represents the footer of a document or a section, typically containing information about the author, copyright information, or related documents.

  Example:

```
<footer>
  <p>Copyright (c) 2023</p>
</footer>
```

Example of Semantic tags:

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

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

    <!---Semantic tags in HTML are elements that convey meaning about the structure and content of a web page to both the browser and the developer.-->
</head>


<body>

    <header> <!--- Represents the header of a section or the entire page. It typically contains site branding, navigation menus, and other introductory content. --->
        <h1>Welcome to My Web Page</h1>

        <nav> <!---Defines a navigation menu, such as a site menu or a table of contents for a page-->
            <ul>
                <li><a href="#">Home</a></li>
                <li><a href="#">About</a></li>
                <li><a href="#">Services</a></li>
                <li><a href="#">Contact</a></li>
            </ul>
        </nav>
    </header>


    <main> <!---Represents the main content of the document. There should be only one <main> element per page-->

        <section> <!---Represents a thematic grouping of content within a page. It helps to organize content into meaningful sections.-->
            <h2>About Us</h2>
            <p>Welcome to our website! We are a team of passionate individuals dedicated to providing you with valuable information.</p>
        </section>

        <section>
            <h2>Services</h2>
            <ul>
                <li><strong>Web Design:</strong> We create beautiful and functional websites.</li>
                <li><strong>Graphic Design:</strong> Our designs will <mark>make your brand stand out.</mark></li>
                <li><strong>SEO:</strong> We optimize your website for search engines.</li>
            <!---<strong> is used to emphasize text.-->
            <!-- <mark>  Represents text that has been highlighted or marked for reference, often with a yellow background. -->
            </ul>
        </section>

        <section>
            <h2>Contact Us</h2>
```

```
            <address> <!---<address> is used to define contact information.-->
                <p>Email: <a href="mailto:contact@example.com">contact@example.com</a></p>
                <p>Phone: <a href="tel:+123456789">+1 (234) 567-89</a></p>
            </address>
        </section>

    </main>

    <footer> <!---It typically contains copyright information, contact details, and other closing content.-->
        <p>&copy; 2023 My Web Page</p>
    </footer>

</body>
</html>

<!-- article:  It can be used for blog posts, news articles, forum posts, and other similar content. -->
```

ChatGPT Chat: https://chat.openai.com/share/e154d280-ce3f-48cb-9710-2689e2512af2

Benefits of Using Semantic HTML:

1. **Accessibility:** Semantic elements are crucial for screen readers and other assistive technologies. They convey the structure and role of the content, helping users navigate and understand it better.
2. **SEO:** Search engines reward well-structured content, and semantic elements make it easier for search engines to identify and index the content properly.
3. **Maintainability:** Semantic elements convey meaning, making it easier for developers to understand the structure and purpose of content in a document.
4. **Styling:** CSS frameworks and browser defaults often provide styles tailored to semantic elements, giving you a head start in designing.

In conclusion, using semantic HTML elements enhances the clarity, accessibility, and discoverability of web content. Developers should aim to use these elements appropriately to create well-structured and meaningful web documents.

Lab Assignment1:

1. Create a meta tag that helps refresh the page after every 5 sec.
2. Question: Implement a user registration form with the following requirements using HTML:

3. Include fields for the user's name, email, and password with appropriate input types and attributes (e.g., minlength, maxlength, required).

4. Include a dropdown menu for the user to select their department (options: Human Resources, Information Technology, Finance).

5. Implement radio buttons for the user's gender (options: Male, Female). Ensure that the user can only select one option.

6. Include checkboxes for food preferences, allowing the user to select multiple options (options: Pizza, Burger, Salad).

7. Add a file input to allow users to upload a profile picture, with the accepted file types limited to PNG and JPEG.

8. Ensure the form is properly structured and follows best practices for accessibility.

Lab Assignment2:

- Building a Comprehensive HTML Form

Problem Statement:

You are tasked with creating an online application form for a fictional Tech Conference named "TechFusion 2024". The conference caters to professionals, students, and enthusiasts from various technology domains. Your task is to design a comprehensive registration form that captures essential information from the attendees.

Instructions:

1. **Personal Details:**
    - Create fields for the applicant's first name and last name. These fields should be mandatory.
    - Include a field for email, which should be validated for the correct email format.
    - Add a phone number field. This field should allow only numeric values.
2. **Profile Type:**
    - Use radio buttons to determine the type of attendee:
        - Professional
        - Student
        - Enthusiast
    - Only one of these options can be selected at a time.
3. **Areas of Interest:**

- Provide a dropdown (`<select>`) that lets the user select their primary domain of interest. Some options can be:
  - Web Development
  - Data Science
  - Cloud Computing
  - Artificial Intelligence
  - Cybersecurity
- This field should be mandatory.

4. **Workshops:**

- The conference is hosting multiple workshops. Allow attendees to select multiple workshops they're interested in using checkboxes:
  - "Intro to Quantum Computing"
  - "Virtual Reality on the Web"
  - "Cybersecurity 101"
  - "AI in Medical Imaging"
- Attendees can select more than one workshop.

5. **Additional Information:**

- Include a `<textarea>` for attendees to provide any additional information or specific requirements they might have. Limit the characters to 300.

6. **Preferred Days:**

- The conference spans over three days. Using a group of checkboxes, allow attendees to select which days they plan to attend:
  - Day 1
  - Day 2
  - Day 3

7. **Meal Preference:**

- Include a set of radio buttons to capture meal preferences:
  - Vegetarian
  - Non-Vegetarian
  - Vegan
  - No Preference

8. **Terms & Conditions:**

- Add a checkbox for "I agree to the terms and conditions." This box should be mandatory to check before form submission.

9. **Submission:**

    - Finally, include a "Submit" button to submit the form.

10. Ensure all the mandatory fields are marked and validations are appropriately set for fields like email and phone number.