FRONT END TRAINING

Basic Research and Documentation on HTML

Submitted By:

Nischal Shrestha

Submitted To:

Ajay Gubhaju

Date: 4th March, 2025

ABSTRACT

The following documentation contains research on HTML basics. The documentation contains a step by step learning module for HTML. Visual Studio Code, an integrated development environment (IDE) was used for practical implementation of the learnt resources. The objective of this report was to make my foundation strong on the Front-End spectrum.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

HTML	HyperText Markup Language
IDE	Integrated Development Environment
WWW	World Wide Web
SEO	Search Engine Optimization
URL	Uniform Resource Locator
IP	Internet Protocol
API	Application Programming Interface
BDO	Bi-Directional Override
HSL	Hue Saturation and Lightness
CSS	Cascading Style Sheets
ASCII	American Standard Code for Information Interchange

HTML

Introduction

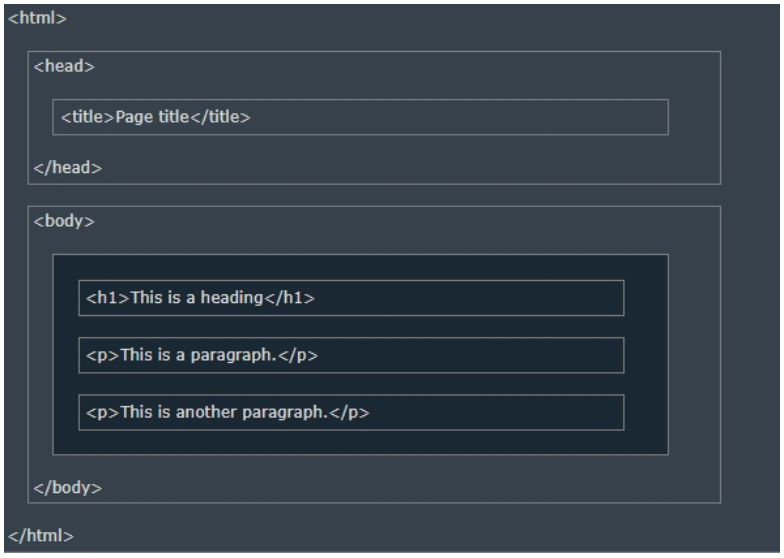
HTML is the most basic building block of the Web that defines the meaning and structure of web content. HTML stands for HyperText Markup Language. “Hypertext” refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, we become an active participant in the World Wide Web (WWW). HTML is **Not Case Sensitive**.

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

A visualization of an HTML page structure is shown below:



The content inside the <body> section will be displayed in a browser whereas the content inside the <title> element will be shown in the browser’s title bar or in the page’s tab.

HTML Editors

A simple text editor is needed to learn HTML. There are numerous HTML editors such as Notepad, TextEdit, Visual Studio Code, etc. to create and modify web pages. For this documentation, we are using an IDE, Visual Studio Code to learn HTML.

Firstly, an HTML code is written in an HTML editor, then it is saved using the ".htm" or ".html" file extension. For e.g. "index.html". And, finally, the result is viewed in any preferred Web Browser.

HTML Documents

All of the HTML documents must begin with a document type declaration: <!DOCTYPE html> which helps browsers to display web pages correctly. It must appear only once, at the top of the page before any HTML tags.

The HTML document itself begins with <html> and ends with </html>.

The visible part of the HTML document is between <body> and </body>.

HTML Elements

An HTML element tells the browser how to display the content. HTML elements label pieces of content such as "this is a heading", "this is a paragraph", etc. which is defined by a start tag, some content, and an end tag.

<tagname> Content </tagname>

Nested HTML Elements

HTML elements can be nested which means that elements can contain other elements. All HTML elements consist of nested HTML elements.

Example

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

<h1>My First Heading</h1>
```

```
<p>My first paragraph.</p>

</body>
</html>
```

Here, <html> element is the root element that defines the whole HTML document.

The <body> element defines the document's body.

The <h1> element defines a heading.

The <p> element defines a paragraph.

Empty HTML Elements

HTML elements with no content are called empty elements. The
 tag defines a line break, and is an empty element without a closing tag.

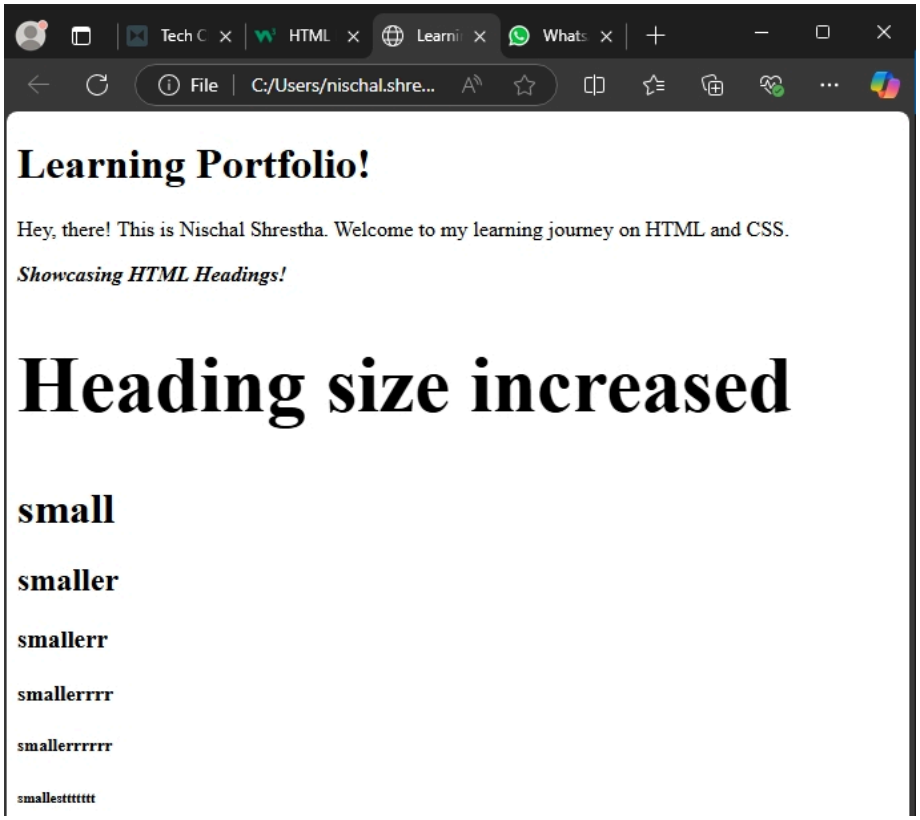
HTML Headings

HTML headings are titles or subtitles that we want to display on a webpage which are defined with the <h1> to <h6> tags. Their importance are:

- Search engines use the headings to index the structure and content of our web pages
- Users usually overview any pages by its headings
- It is important to use headings to show the document structure
- <h1> headings is used for main headings, followed by <h2> headings, and so on

If the user wants a bigger heading, we can specify the size for any heading with the "style" attribute, using the CSS font-size property

<h1 style = "font-size:60px;">Heading size increased</h1>



HTML Paragraphs

<p> element defines a paragraph in HTML. A paragraph always starts on a new line, and browsers automatically add some white space (a margin) before and after a paragraph.

HTML Display

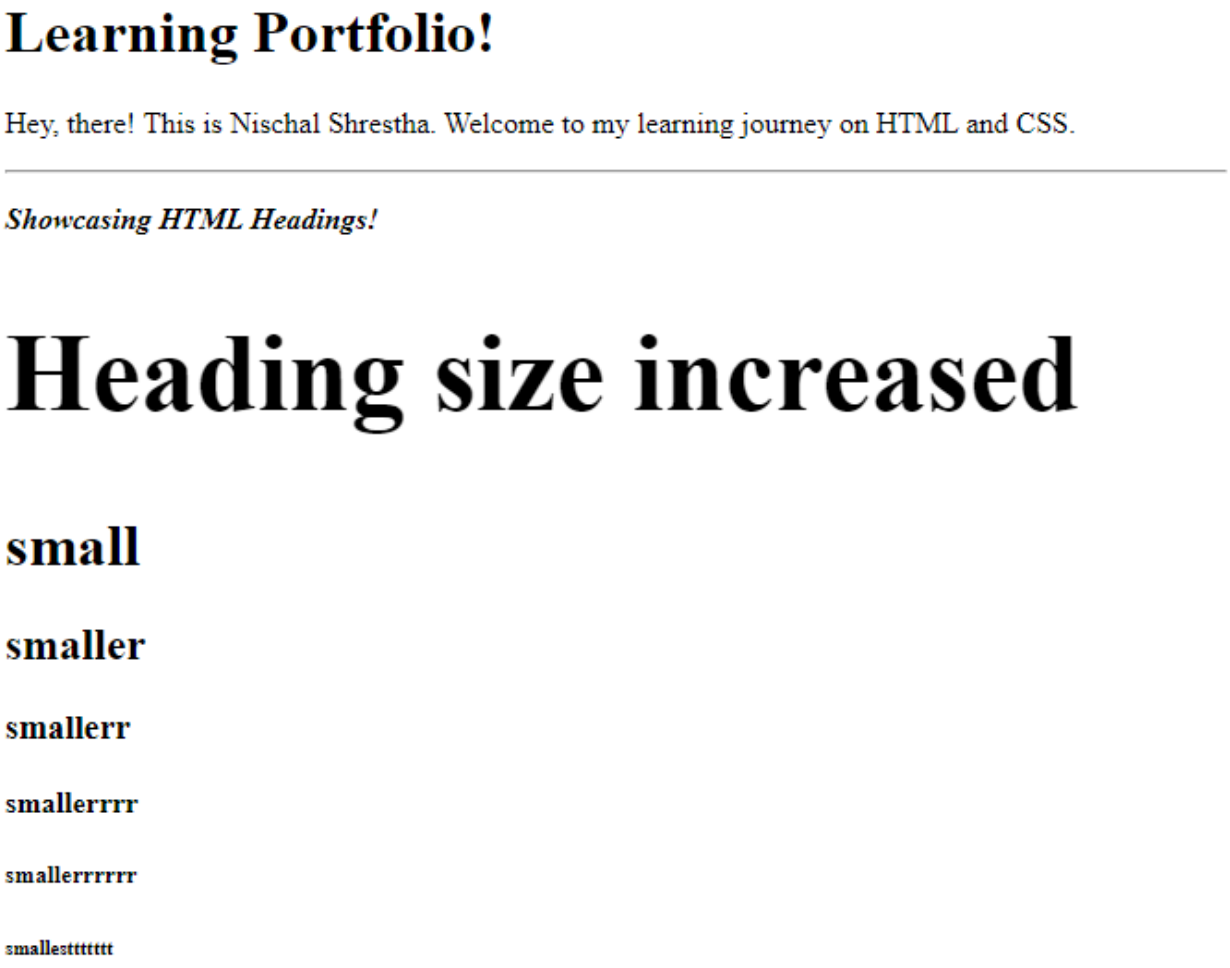
With just HTML, we cannot change the display by adding extra spaces or extra lines in our HTML code; the browser will automatically remove any extra spaces and lines when the page is displayed. Large or small screens, and resized windows will create different results.

HTML <pre> Element

The HTML <pre> element defines preformatted text. The text inside a <pre> element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks.

HTML Horizontal Rules

<hr> tag defines a thematic break in an HTML page, and is often displayed as a horizontal rule. It is also used to separate content (or define a change) in an HTML page.



HTML Line Breaks

 element defines a line break. It is mostly used when we want a line break i.e. a new line without starting a new paragraph.

HTML Attributes

HTML attributes provide additional information about HTML elements. They are always specified in the start tag. They usually come in name/value pairs like: name=”value”

href attribute

The <a> tag defines a hyperlink. The href attribute specifies the URL of the page the link goes to.

 Link Text

src attribute

The <img> tag is used to embed an image in an HTML page. The src attribute specifies the path to the image to be displayed.

<img src=”url”>

There are two ways to specify the URL in the src attribute:

- **Absolute URL**
  - It links to an external image that is hosted on another website. Example:
    src= “https://www.w3schools.com/images/img_girl.jpg"
  - External images might be under copyright. If we do not get permission to use it, we might be in violation of copyright laws. In addition, we cannot control external images; they can suddenly be removed or changed.
- **Relative URL**
  - It links to an image that is hosted within the website.
  - The URL does not include the domain name.
  - If the URL begins without a slash, it will be relative to the current page.
    src="img_girl.jpg"
  - If the URL begins with a slash, it will be relative to the domain.
    src="/images/img_girl.jpg"

P.S.: It is almost always best to use relative URLs as they will not break if we change the domain.

width and height Attributes

The <img> tag should also contain the width and height attributes, which specify the width and height of the image (in pixels). Example:

<img src= “img_girl.jpg” width= “500” height= “600”>

alt Attribute

The required alt attribute for the <img> 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. Example:

<img src= “one_piece.png” alt= “One Piece Logo”>

style Attribute

The style attribute is used to add styles to an element, such as color, font, size, and more. Example:

 Hey, there! This is Nischal Shrestha. Welcome to my learning journey on HTML and CSS.

Hey, there! This is Nischal Shrestha. Welcome to my learning journey on HTML and CSS.

lang Attribute

The lang attribute inside the <html> tag is used to declare the language of the Web page. This is meant to assist search engines and browsers. Example:

<html lang= “en”>

Country codes can also be added to the language code in the lang attribute. So, the first two characters define the language of the HTML page, and the last two characters define the country. Example:

<html lang= “en-US”>

title Attribute

The title attribute defines some extra information about an element. The value of the title attribute will be displayed as a tooltip when you mouse over the element. Example:

<p title= “My name”> Nischal Shrestha </p>

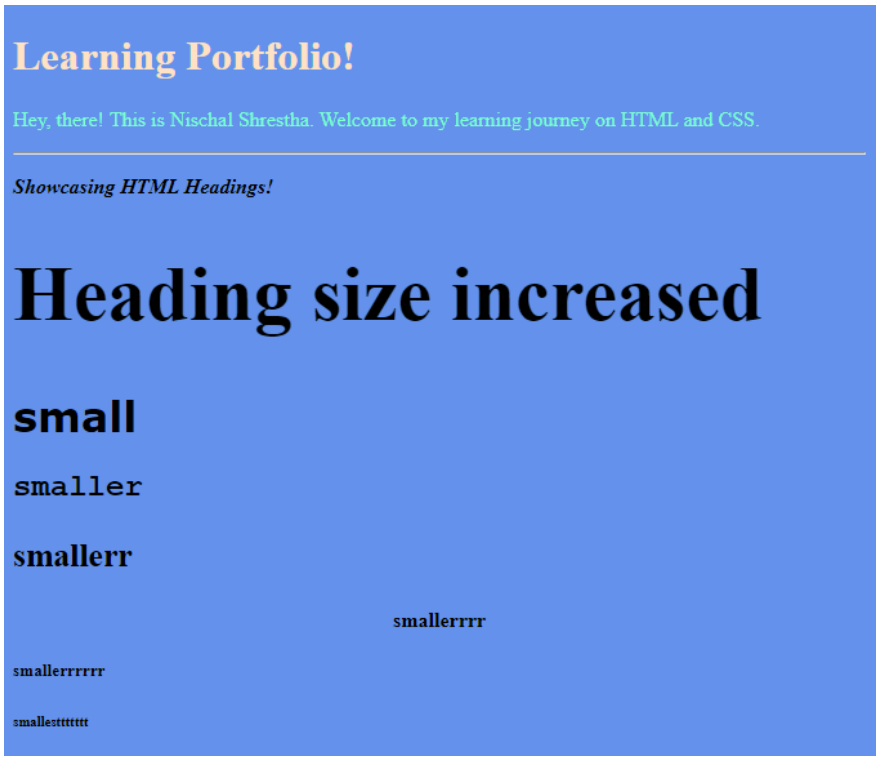
HTML Styles

It is used to add styles to an element such as color, font, size and more.

Syntax:

<tagname style = “property:value;”>

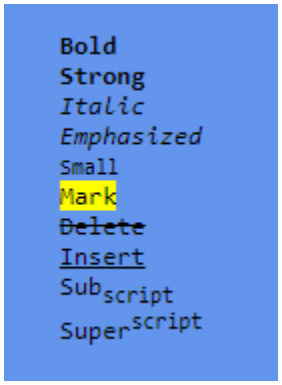
Following is an example which includes demo of “background-color, color, font-family, font-size, text-align”:



HTML Text Formatting

HTML contains several elements for defining text with a special meaning. Following are some of the formatting elements that displays special types of text:

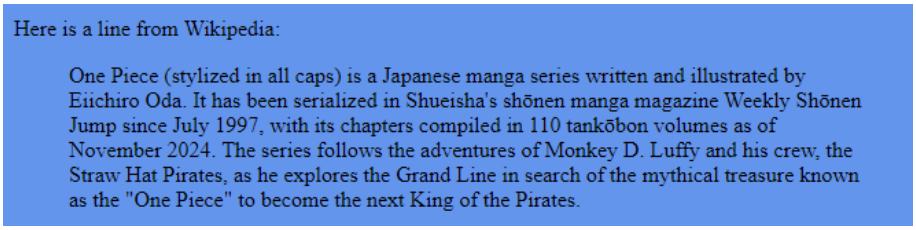
- <b> - Bold text
- - Important text
- <i> - Italic text
- - Emphasized text
- <mark> - Marked text
- <small> - Smaller text
- <del> - Deleted text
- <ins> - Inserted text
- <sub> - Subscript text
- <sup> - Superscript text



HTML Quotation and Citation Elements

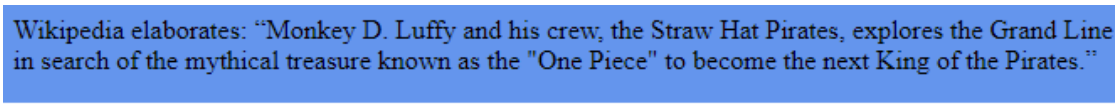
HTML <blockquote>

It defines a section that is quoted from another source. Browsers usually indent <blockquote> elements.



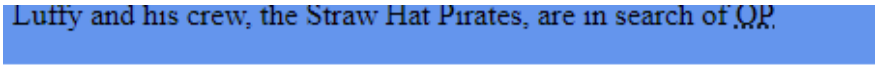
HTML <q>

It defines a short quotation. Browsers normally insert quotation marks around the quotation.



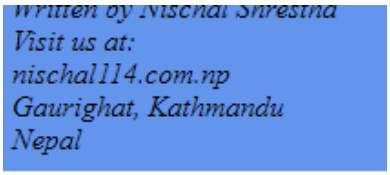
HTML <abbr>

It defines an abbreviation or an acronym, like “Mr.”, “DR.”. It gives useful information to browsers, translation systems and search-engines.



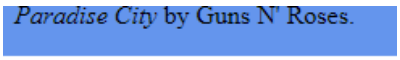
HTML <address>

It defines the contact information for the author/owner of a document or an article which can include an email address, URL, physical address, phone number, social media handle, etc. Text in the <address> element usually renders in italic, and browsers will always add a line break before and after the <address> element.



HTML <cite>

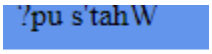
It defines the title of a creative work (e.g. a book, a song, etc).



.

HTML <bdo>

BDO stands for Bi-Directional Override. It is used to override the current text direction.



HTML Comments

Syntax:

<!-- Comments –>

HTML Colors

HTML colors are specified with predefined color names, or with RGB, HEX, HSL, RGBA, or HSLA values. Colors can be added to various elements of the webpages. Some of them are:

- Background
- Text
- Border

HTML RGB

A color can be specified as an RGB value, using the formula:

rgb(red, green, blue)

Each parameter (red, green, blue) defines the intensity of the color with a value between 0 and 255

HTML RGBA

They are an extension of RGB color values with an Alpha channel which specifies the opacity for a color. An RGBA color value is specified with:

rgba(red, green, blue, alpha)

The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all)

HTML HEX colors

A color can be specified using a hexadecimal value in the form: #rrggbb

Here rr (red), gg (green) and bb (blue) hexadecimal values between 00 and ff specify the components of the color.

HTML HSL and HSLA Colors

HSL stands for hue, saturation, and lightness. HSLA color values are an extension of HSL with an Alpha channel (opacity).

hsla(hue, saturation, lightness, alpha)

- Hue is a degree on the color wheel from 0 to 360. 0 is red, 120 is green, and 240 is blue.
- Saturation is a percentage value. 0% means a shade of gray, and 100% is the full color.
- Lightness is also a percentage value. 0% is black, and 100% is white.
- Alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all)

HTML Styles - CSS

CSS stands for Cascading Style Sheets which is used to format the layout of a webpage. It controls the layout of multiple web pages all at once.

Types:

- Inline - by using the style attribute inside HTML elements
- Internal - by using a <style> element in the <head> section
- External - by using a <link> element to link to an external CSS file

Inline CSS

It is used to apply a unique style to a single HTML element. Example:

<h1 style = “color:violet;”>A violet heading</h1>

Internal CSS

It is used to define a style for a single HTML page. Example:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {background-color:antiquewhite;}
      h1 {color:coral;}
      p {color:burlywood;}
    </style>
  </head>
  <body>
    <h1>Heading by Nischal</h1>
    <p>Hey, there! Check out my CSS practice sessions.</p>
  </body>
</html>
```

External CSS

An external style sheet is used to define the style for many HTML pages. Example:

In index.html,

```
<!DOCTYPE html>
<html>
    <head>
        <link rel="stylesheet" href="styles.css">
        <!-- <style>
            body {background-color:antiquewhite;}
            h1 {color:coral;}
            p {color:burlywood;}
        </style> -->
```

```
    </head>
    <body>
        <h1>Heading by Nischal</h1>
        <p>Hey, there! Check out my CSS practice sessions.</p>
    </body>
</html>
```

In styles.css,

```
body{background-color:black;}
h1{color:coral;}
p {color:burlywood;}
```

CSS Colors, Fonts and Sizes

- CSS color property defines the text color to be used
- CSS font-family property defines the font to be used
- CSS font-size property defines the text size to be used

Example:

In CSS,

```
body
{
    background-color:black;
}
h1
{
    color:coral;
    font-family:Verdana, Geneva, Tahoma, sans-serif;
    font-size:300%;
}
p
{
```

```
    color:burlywood;
       font-family:'Franklin  Gothic  Medium',  'Arial  Narrow',
Arial, sans-serif;
    font-size:150%;
}
```

CSS Border

It Defines a border around an HTML element. Example:

In CSS,

```
body
{
    background-color:black;
}
h1
{
    color:coral;
    font-family:Verdana, Geneva, Tahoma, sans-serif;
    font-size:300%;
}
p
{
    border: 2px solid powderblue;
    color:burlywood;
       font-family:'Franklin  Gothic  Medium',  'Arial  Narrow',
Arial, sans-serif;
    font-size:150%;
}
```

CSS Padding

It defines a padding (space) between the text and the border. Example:

In CSS,

```
body
{
    background-color:black;
}
h1
{
    color:coral;
    font-family:Verdana, Geneva, Tahoma, sans-serif;
    font-size:300%;
}
p
{
    border: 2px solid powderblue;
    padding: 30px;
    color:burlywood;
       font-family:'Franklin Gothic Medium', 'Arial Narrow',
Arial, sans-serif;
    font-size:150%;
}
```

CSS Margin

It defines a margin (space) outside the border. Example:

In CSS,

```
body
{
    background-color:black;
}
h1
{
    color:coral;
    font-family:Verdana, Geneva, Tahoma, sans-serif;
    font-size:300%;
}
p
{
    border: 2px solid powderblue;
    padding: 30px;
    margin: 50px;
    color:burlywood;
       font-family:'Franklin Gothic Medium', 'Arial Narrow',
Arial, sans-serif;
    font-size:150%;
}
```

Link to External CSS

External style sheets can be referenced with a full URL or with a path relative to the current web page. Example:

In CSS,

```
<!DOCTYPE html>
<html>
    <head>
                                        <link     rel="stylesheet"
href="https://www.w3schools.com/html/styles.css">
            </head>
    <body>
        <h1>Heading by Nischal</h1>
        <p>Hey, there! Check out my CSS practice sessions.</p>
    </body>
</html>
```

HTML Links

Links allow users to click their way from page to page. HTML links are hyperlinks. We can click on a link and jump to another document. It does not have to be a text, it can be an image or any other HTML element.

Syntax:

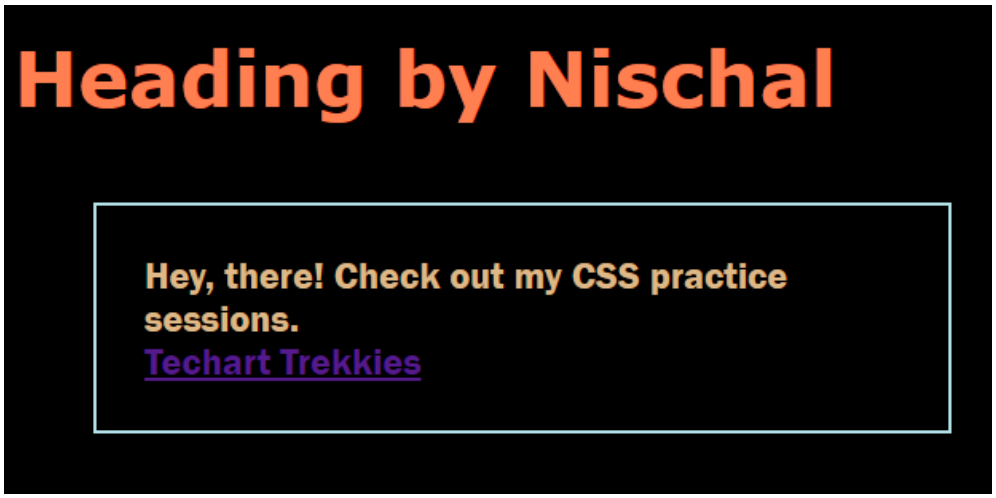
 Link Text

The most important attribute of the <a> element is the href attribute, which indicates the link's destination. The link text is the part that will be visible to the reader. Clicking on the link text, will send the reader to the specified URL address.

Example:

```
<!DOCTYPE html>
<html>
    <head>
        <link rel="stylesheet" href="styles.css">
    </head>
    <body>
        <h1>Heading by Nischal</h1>
```

```
        <p>Hey, there! Check out my CSS practice sessions.<br>
                                <a   href="https://www.techarttrekkies.com/"
target="_blank">Techart Trekkies</a>
        </p>
    </body>
</html>
```



The target Attribute

By default, the linked page will be displayed in the current browser window. To change this, we must specify another target for the link. The target attribute specifies where to open the linked document. The target attribute can have one of the following values:

- _self - Default. Opens the document in the same window/tab as it was clicked
- _blank - Opens the document in a new window or tab
- _parent - Opens the document in the parent frame
- _top - Opens the document in the full body of the window

Absolute URLs VS Relative URLs

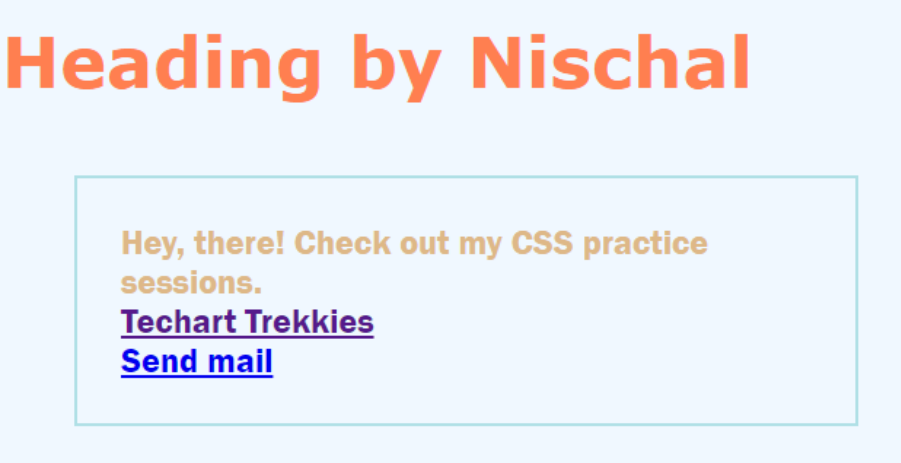
An absolute URL is a full web address in the href attribute whereas a local link (a link to a page within the same website) is specified with a relative URL without the "https://www" part)

Link to an Email Address

We use “mailto:” inside the href attribute to create a link that opens the user’s email program to let the end user send a new email.

Example:

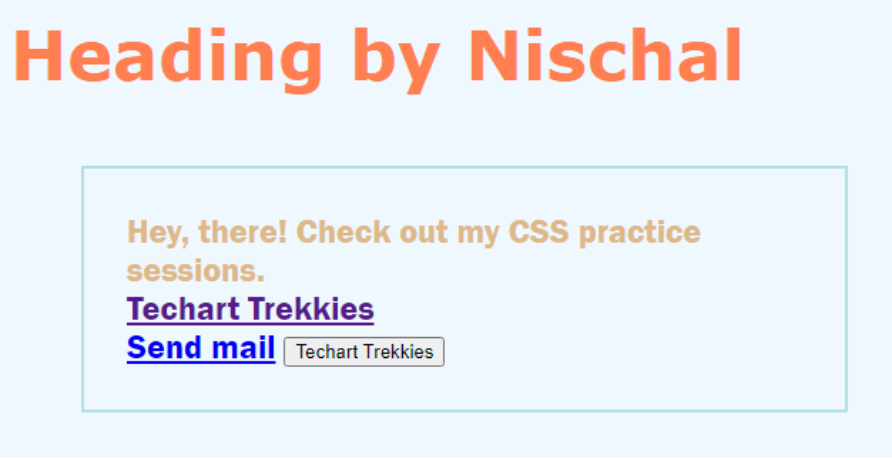
Send email



Button as a link

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

Example:



Link Titles

The title attribute specifies extra information about an element. The information is most often shown as a tooltip text when the mouse moves over the element.

Example:

```
 <a href="https://www.techarttrekkies.com/" target="_blank" title="Go to
Techart Trekkies Website">Techart Trekkies</a>
```

HTML Link Colors

An HTML link is displayed in a different color depending on whether it has been visited, is unvisited, or is active.

By default, a link will appear like this (in all browsers):

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

Example:

```
<style>
a:link{
    color:chocolate;
    background-color: transparent;
    text-decoration: none;
}
a:visited{
    color:pink;
    background-color: transparent;
    text-decoration: none;
}
a:hover{
    color:red;
    background-color: transparent;
    text-decoration: underline;
}
a:active{
    color:yellow;
    background-color: transparent;
```

```
    text-decoration: underline;
}
</style>
```

HTML Links - Create Bookmarks

HTML links can be used to create bookmarks, so that readers can jump to specific parts of a web page. Bookmarks can be useful if a web page is very long.

To create a bookmark - first create the bookmark, then add a link to it.

When the link is clicked, the page will scroll down or up to the location with the bookmark.

Example:

First, use the id attribute to create a bookmark:

```
<h1 style ="font-size: 60px;" id="heading">Heading size increased</h1>
```

Then, add a link to the bookmark, from within the same page:

```
  <a href="#heading">Jump to heading</a>
```

We can also add a link to a bookmark on another page:

```
<a href="index.html#heading">Jump to heading</a>
```

HTML Images

The HTML <img> tag is used to embed an image in a web page.

Images are not technically inserted into a web page; images are linked to web pages. The <img> tag creates a holding space for the referenced image.

The <img> tag is empty, it contains attributes only, and does not have a closing tag.

The <img> tag has two required attributes:

- src - Specifies the path to the image
- alt - Specifies an alternate text for the image

Syntax:

<img src= “url” alt= “alternate text”>

The src Attribute

The required src attribute specifies the path (URL) to the image.

P.S.: When a web page loads, it is the browser, at that moment, that gets the image from a web server and inserts it into the page. Therefore, we have to make sure that the image actually stays in the same spot in relation to the web page, otherwise other users or visitors will get a broken link icon. The broken link icon and the *alt* text are shown if the browser cannot find the image.

The alt Attribute

The required *alt* attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the src attribute, or if the user uses a screen reader). The value of the *alt* attribute should describe the image.

Image Size - Width and Height

The *style* attribute is used to specify the width and height of an image.

Example:

```
<img src="one_piece.png" alt="One Piece Logo" style="width: 250px;height: 100px;">
```

We can also use the *width* and *height* attributes.

Example:

```
<img src="one_piece.png" alt="One Piece Logo" width="250" height ="100">
```

The *width* and *height* attributes always define the width and height of the image in pixels.

P.S.: It is always recommended to specify the width and height of an image. If width and height are not specified, the web page might flicker while the image loads.

Width and Height, or Style?

The width, height, and style attributes are all valid in HTML.

However, using the style attribute is the best as it prevents the style sheets from changing the size of images.

Image Floating

We can use the CSS *float* property to let the image float to the right or to the left of a text.

Example:

```
   <img src="C:\Users\nischal.shrestha\Desktop\Nischal
Shrestha\one_piece.png" alt="One Piece Logo"
style="float:right;width:150px; height:50px">
```

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.

Example:

```
<img src="C:\Users\nischal.shrestha\Desktop\Nischal
Shrestha\one_piece.png" alt="One Piece Logo"
style="float:right;width:150px; height:50px" usemap="#omap">
            <map name="omap">
                <area shape="circle" coords="10,10,10,10" alt="One Piece"
href="https://myanimelist.net/anime/21/One_Piece/video" target="_blank">
                <area shape="circle" coords="100,100,100,100" alt="One
Piece" href="https://onepiece.fandom.com/wiki/One_Piece_Wiki"
target="_blank">
            </map>
```

The image is inserted using the <img> tag and adds a *usemap* attribute.

Then, we need to add a <map> element which is used to create an image map that is linked to the image by using the required *name* attribute.

The Areas

Then, we add the clickable areas. A clickable area is defined using an <area> element.

Shape. We must define the shape of the clickable area, and we can choose one of these values:

- rect - defines a rectangular region
- circle - defines a circular region
- poly - defines a polygonal region
- default - defines the entire region

We must also define some coordinates to be able to place the clickable area onto the image.

Shape="rect"

The coordinates for shape="rect" come in pairs, one for the x-axis and one for the y-axis.

So, the coordinates 34,44 is located 34 pixels from the left margin and 44 pixels from the top. The coordinates 270,350 is located 270 pixels from the left margin and 350 pixels from the top. This gives up enough data to create a clickable rectangular area.

Example:

```
<area shape="rect" coords="34, 44, 270, 350" href="computer.htm">
```

Shape="circle"

To add a circle area, first locate the coordinates of the center of the circle: 337,300

Then specify the radius of the circle: 44 pixels

Example:

```
<area shape="circle" coords="337, 300, 44" href="coffee.htm">
```

Shape="poly"

The shape="poly" contains several coordinate points, which creates a shape formed with straight lines (a polygon). This can be used to create any shape. Like maybe a croissant shape!

We have to find the x and y coordinates for all edges of the place we want to map.

Example:

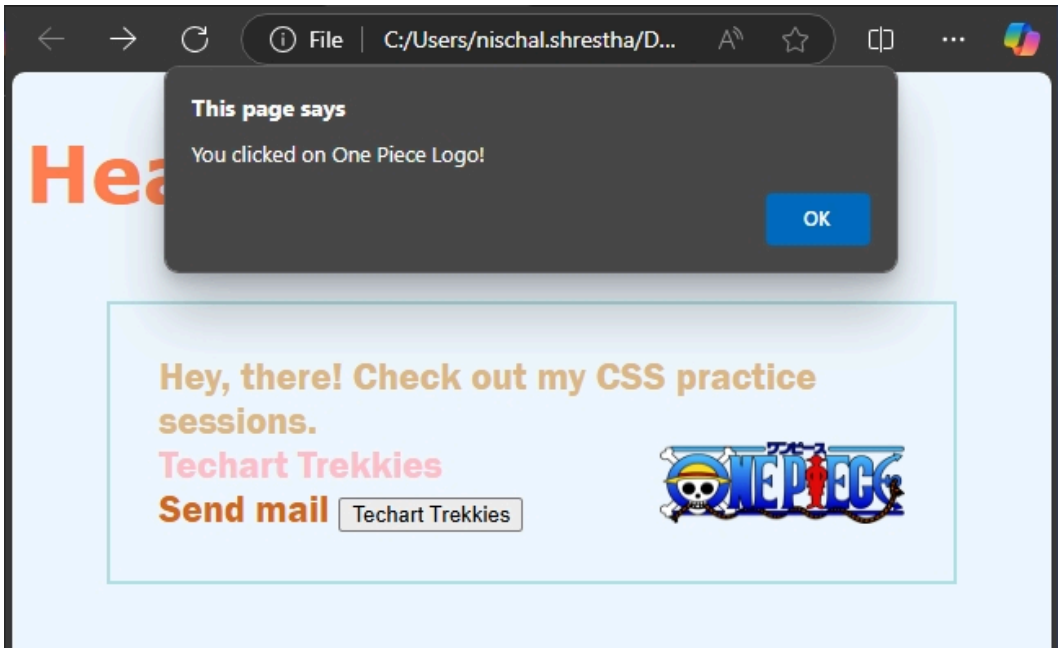
<area shape="poly" coords = "140,121,181,116,204,160,204,222,191,270,140,329,85,355,58,352,37,322,40,259,103,161,128,147" href="croissant.htm">

Image Map and JavaScript

A clickable area can also trigger a JavaScript function. We just need to add a click event to the <area> element to execute a JavaScript function.

Example:

```
  <map name="omap">
                <area shape="circle" coords="10,10,10,10" alt="One Piece"
href="https://myanimelist.net/anime/21/One_Piece/video" target="_blank">
                <area shape="circle" coords="100,100,100,100" alt="One
Piece" href="https://onepiece.fandom.com/wiki/One_Piece_Wiki"
target="_blank" onclick="myFunction()">
            </map>
            <script>
                function myFunction(){
                    alert("You clicked on One Piece Logo!");
                }
            </script>
```



HTML Background Images

To add a background image on an HTML element, we use the HTML *style* attribute and the CSS *background-image* property.

Example:

```
<p style="background-image: url('luffy.jpg');">
            Here is a line from Wikipedia:
        </p>
```

Background Image on a Page

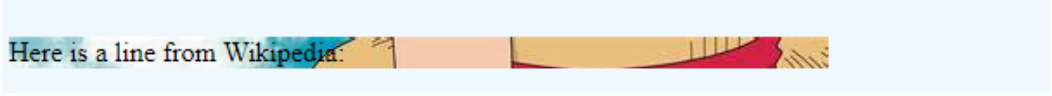
If we want the entire page to have a background image, we must specify the background image on the <body> element.

Background Repeat

To avoid the background image from repeating itself, set the *background-repeat* property to *no-repeat*.

Example:

```
 <p style="background-image: url('luffy.jpg'); background-repeat:
no-repeat;">
            Here is a line from Wikipedia:
        </p>
```

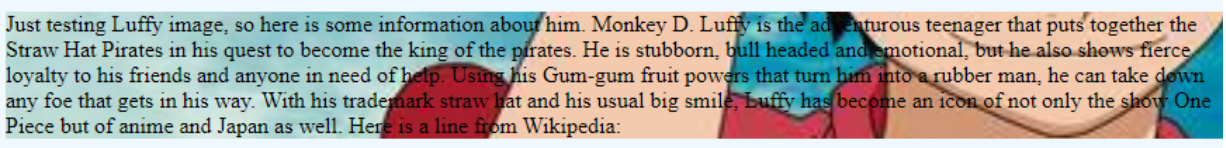


Background Cover

If we want the background image to cover the entire element, we can set the *background-size* property to *cover*. Also, to make sure the entire element is always covered, set the *background-attachment* property to *fixed*. This way, the background image will cover the entire element, with no stretching (the image will keep its original proportions).

Example:

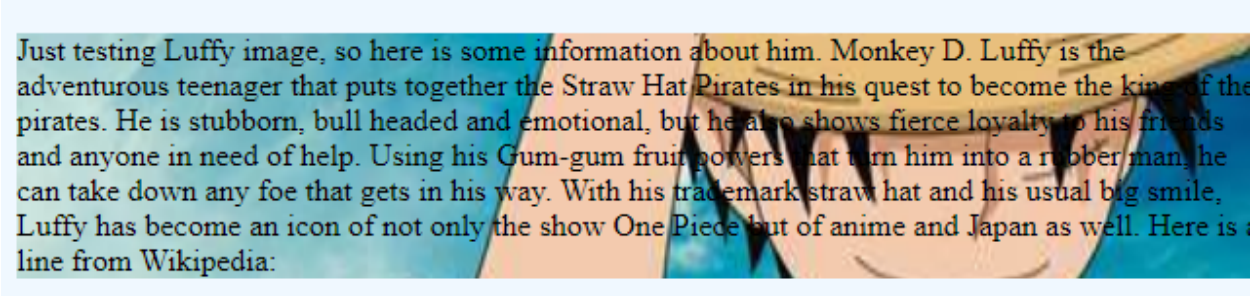
```
   <p style="background-image: url('luffy.jpg'); background-repeat:
no-repeat; background-attachment: fixed; background-size: cover;">
            Just testing Luffy image, so here is some information about
him. Monkey D. Luffy is the adventurous teenager that puts together the
Straw Hat Pirates in his quest to become the king of the pirates. He is
stubborn, bull headed and emotional, but he also shows fierce loyalty to
his friends and anyone in need of help. Using his Gum-gum fruit powers
that turn him into a rubber man, he can take down any foe that gets in his
way. With his trademark straw hat and his usual big smile, Luffy has
become an icon of not only the show One Piece but of anime and Japan as
well.
            Here is a line from Wikipedia:
        </p>
```



Background Stretch

If we want the background image to stretch to fit the entire element, we can set the *background-size* property to *100% 100%*.

```
<p style="background-image: url('luffy.jpg'); background-repeat:
no-repeat; background-attachment: fixed; background-size: 100% 100%;">
            Just testing Luffy image, so here is some information about
him. Monkey D. Luffy is the adventurous teenager that puts together the
Straw Hat Pirates in his quest to become the king of the pirates. He is
stubborn, bull headed and emotional, but he also shows fierce loyalty to
his friends and anyone in need of help. Using his Gum-gum fruit powers
that turn him into a rubber man, he can take down any foe that gets in his
way. With his trademark straw hat and his usual big smile, Luffy has
become an icon of not only the show One Piece but of anime and Japan as
well.
            Here is a line from Wikipedia:
        </p>
```



HTML <picture> Element

It allows users to display different pictures for different devices or screen sizes. It gives web developers more flexibility in specifying image resources. The <picture> element contains one or more <source> elements, each referring to different images through the srcset attribute. This way the browser can choose the image that best fits the current view and/or device. Each <source> element has a media attribute that defines when the image is the most suitable.

Example:

```
<picture>
  <source media="(min-width: 650px)" srcset="img_food.jpg">
  <source media="(min-width: 465px)" srcset="img_car.jpg">
  <img src="img_girl.jpg">
</picture>
```

When to use the Picture Element

There are two main purposes for the <picture> element:

- Bandwidth

  If we have a small screen or device, it is not necessary to load a large image file. The browser will use the first <source> element with matching attribute values, and ignore any of the following elements.

- Format Support

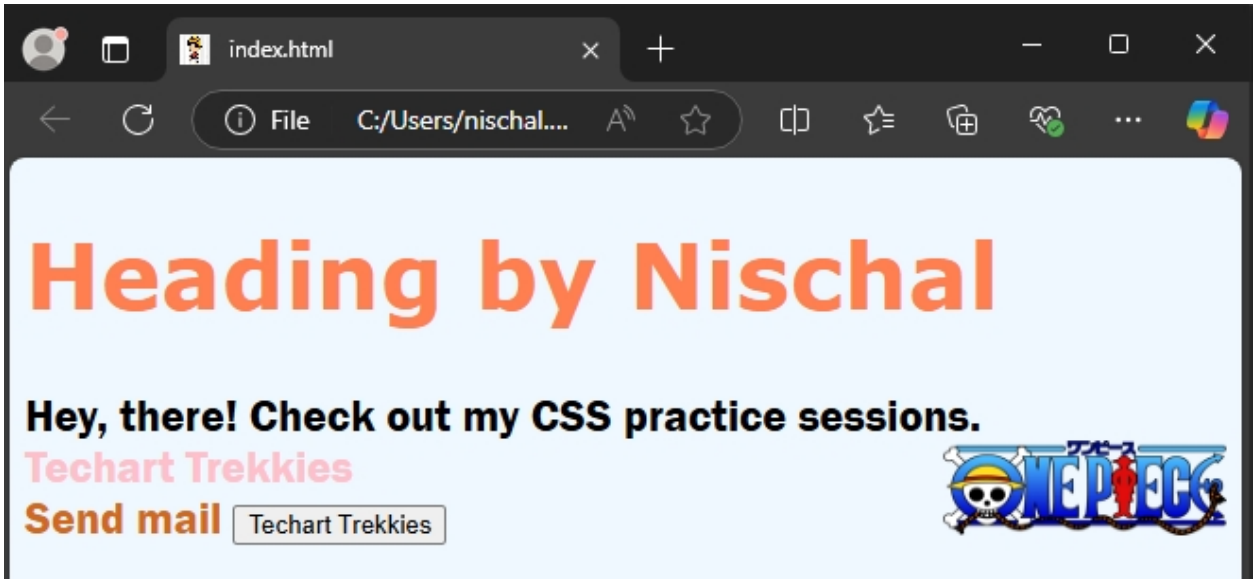
  Some browsers or devices may not support all image formats. By using the <picture> element, we can add images of all formats, and the browser will use the first format it recognizes, and ignore any of the following elements.

HTML Favicon

A favicon is a small image displayed next to the page title in the browser tab. We can use any image we like as our favicon. A favicon is a small image, so it should be a simple image with high contrast.

Example:

```
<link rel="icon" type="image/x-icon"
href="C:\Users\nischal.shrestha\Desktop\Nischal Shrestha\favicon.ico">
```



HTML Page Title

The title should describe the content and the meaning of the page. The page title is very important for search engine optimization (SEO). The text is used by search engine algorithms to decide the order when listing pages in search results.

The <title> element:

- defines a title in the browser toolbar
- provides a title for the page when it is added to favorites
- displays a title for the page in search engine-results

HTML Tables

It allows web developers to arrange data into rows and columns.

Table Cells

It is defined by a <td> and a </td> tag. It can contain all sorts of HTML elements: text, images, lists, links, other tables, etc.

Table Rows

It starts with a <tr> and ends with a </tr> tag.

Table Headers

It is defined by <th> tag.

Headers for Multiple Columns

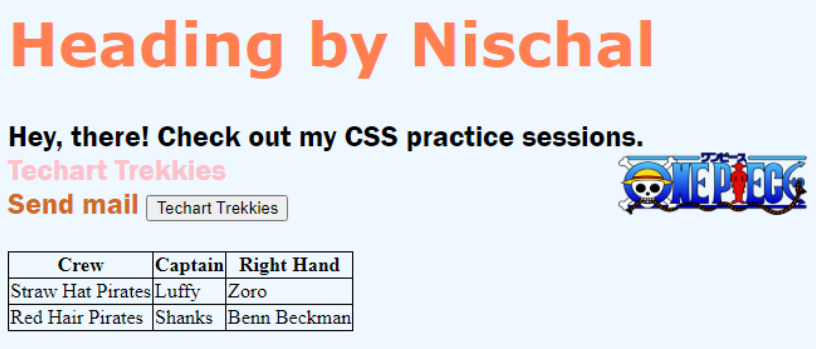
We can have a header that spans over two or more columns.

Identification

Name		Address
Nischal	Shrestha	Gaurighat
Roronoa	Zoro	Fiction

Table Borders

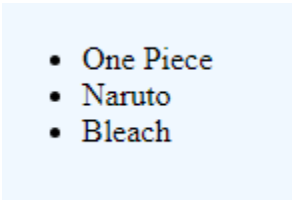
To add a border, use the CSS border property on table, th, and td elements. To avoid having double borders like in the example above, set the CSS *border-collapse* property to *collapse*.



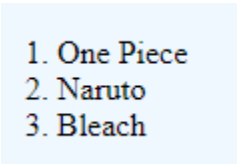
HTML Lists

It allows web developers to group a set of related items in lists. It's types are:

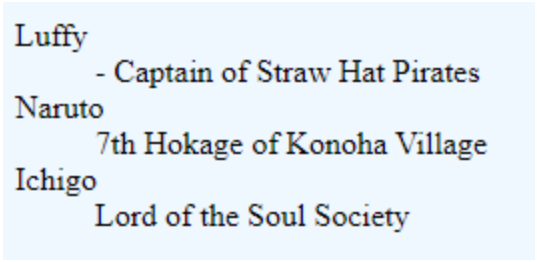
- Unordered HTML List
    - It starts with <ul> tag.
    - Each list item starts with the <li> tag.



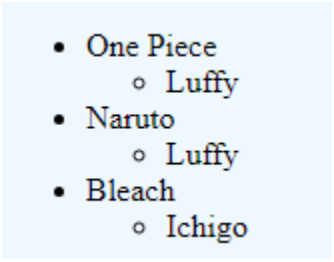
- Ordered HTML List
    - It starts with <ol> tag.
    - Each list item starts with the <li> tag.



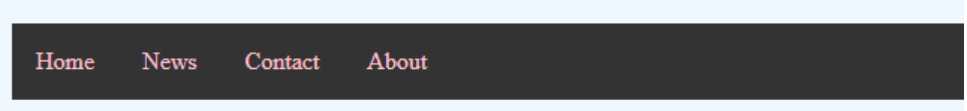
- HTML Description List
    - It is a list of terms, with a description of each term.
    - <dl> tag defines them
    - <dt> tag defines the term (name)
    - <dd> tag describes each term



- Nested HTML List



- Horizontal List



Block-level Elements

A block-level element always starts on a new line, and the browsers automatically add some space (a margin) before and after the element. A block-level element always takes up the full width available (stretches out to the left and right as far as it can). Two commonly used block elements are: <p> and .

The <p> element defines a paragraph in an HTML document.

The element defines a division or a section in an HTML document.

Inline Elements

An inline element does not start on a new line.

An inline element only takes up as much width as necessary. Commonly used inline elements are: and <a>.

HTML class Attribute

It is used to specify a class for an HTML element. Multiple HTML elements can share the same class. The *class* attribute is often used to point to a class name in a style sheet. It can also be used by JavaScript to access and manipulate elements with the specific class name.

Syntax:

To create a class; we need to write a period (.) character, followed by a class name. Then, define the CSS properties within curly braces {}.

HTML id Attribute

The HTML id attribute is used to specify a unique id for an HTML element. We cannot have more than one element with the same id in an HTML document. The value of the id attribute must be unique within the HTML document. The id attribute is used to point to a specific style declaration in a style sheet. It is also used by JavaScript to access and manipulate the element with the specific id.

Syntax:

Write a hash character (#), followed by an id name. Then, define the CSS properties within curly braces {}.

P.S.: The id name is case sensitive. The id name must contain at least one character, cannot start with a number, and must not contain whitespaces (spaces, tabs, etc.)

Difference Between Class and ID

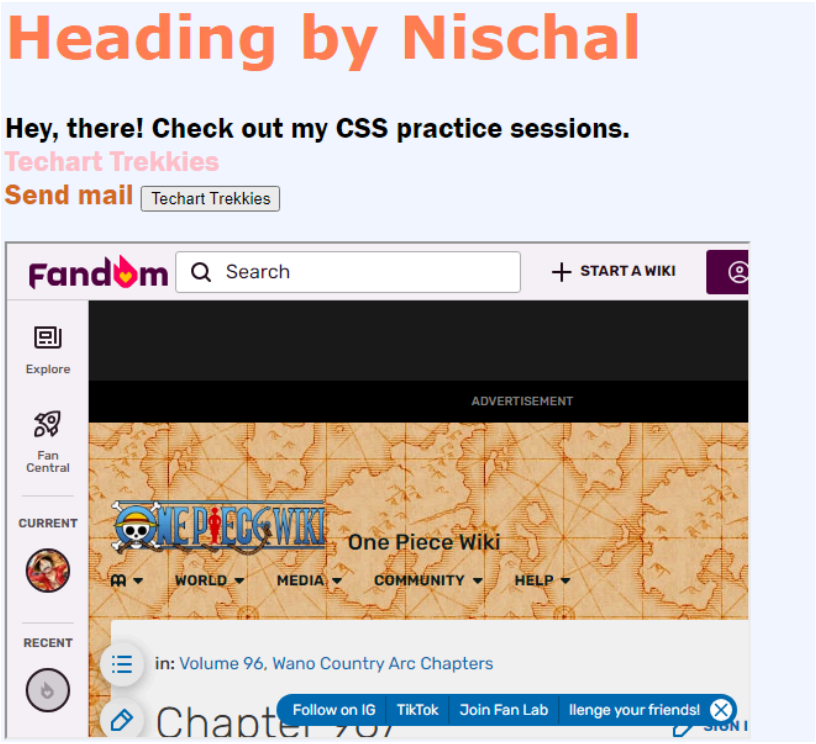
A class name can be used by multiple HTML elements, while an id name must only be used by one HTML element within the page.

HTML iframes

It is used to display a web page within a web page. An inline frame is used to embed another document within the current HTML document.

Syntax:

<iframe src= “url” title= “description”></iframe>



HTML Responsive Web Design

Responsive web design is about creating web pages that look good on all devices. A responsive web design will automatically adjust for different screen sizes and viewports. It is about using HTML and CSS to automatically resize, hide, shrink, or enlarge a website, to make it look good on all devices (desktops, tablets, and phones).

Setting The Viewport:

To create a responsive website, add the following <meta> tag to all your web pages

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Responsive Web Design - Frameworks

There are numerous CSS Frameworks that offer responsive design which are free, and easy to use. Some of them are: W3.CSS, Bootstrap, etc.

HTML Semantic Elements

Semantic elements means elements with a meaning. A semantic element clearly describes its meaning to both the browser and the developer.
Examples of non-semantic elements: and - Tells nothing about its content.
Examples of semantic elements: <img>, <table>, and <article> - Clearly defines its content.

HTML Uniform Resource Locators

A URL is another word for a web address. A URL can be composed of words (e.g. w3schools.com), or an Internet Protocol (IP) address (e.g. 192.68.20.50). Most people enter the name when surfing, because names are easier to remember than numbers.
Web browsers request pages from web servers by using a URL which is used to address a document (or other data) on the web.

URL Encoding

URLs can only be sent over the Internet using the ASCII character-set. If a URL contains characters outside the ASCII set, the URL has to be converted. URL encoding converts non-ASCII characters into a format that can be transmitted over the Internet. URL encoding replaces non-ASCII characters with a "%" followed by hexadecimal digits. URLs cannot contain spaces. URL encoding normally replaces a space with a plus (+) sign, or %20.

HTML Forms

Introduction

An HTML form is used to collect user input. The user input is most often sent to a server for processing. The HTML <*form*> element is used to create an HTML form for user input.

<form> Element

Syntax:

<form>
form elements
</form>

The <*form*> element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.

<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.

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

Text Fields

The <input type="text"> defines a single-line input field for text input.

The <label> Element

The <label> tag defines a label for many form elements. It is useful for screen-reader users, because the screen-reader will read out loud the label when the user focuses on the input element. It also helps users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the <label> element, it toggles the radio button/checkbox. The for attribute of the <label> tag should be equal to the id attribute of the <input> element to bind them together.

The Action Attribute

The action attribute defines the action to be performed when the form is submitted. Usually, the form data is sent to a file on the server when the user clicks on the submit button.

The Target Attribute

The target attribute specifies where to display the response that is received after submitting the form. The target attribute can have one of the following values:

Value	Description
_blank	The response is displayed in a new window or tab
_self	The response is displayed in the current window
_parent	The response is displayed in the parent frame
_top	The response is displayed in the full body of the window
framename	The response is displayed in a named iframe

The Method Attribute

The method attribute specifies the HTTP method to be used when submitting the form data. The form-data can be sent as URL variables (with method="get") or as HTTP post transaction (with method="post"). The default HTTP method when submitting form data is GET.

Notes on GET:

- Appends the form data to the URL, in name/value pairs
- NEVER use GET to send sensitive data! (the submitted form data is visible in the URL!)
- The length of a URL is limited (2048 characters)
- Useful for form submissions where a user wants to bookmark the result
- GET is good for non-secure data, like query strings in Google

Notes on POST:

- Appends the form data inside the body of the HTTP request (the submitted form data is not shown in the URL)
- POST has no size limitations, and can be used to send large amounts of data.
- Form submissions with POST cannot be bookmarked

The Autocomplete Attribute

The autocomplete attribute specifies whether a form should have autocomplete on or off. When autocomplete is on, the browser automatically completes values based on values that the user has entered before.

The Novalidate Attribute

The novalidate attribute is a boolean attribute. When present, it specifies that the form-data (input) should not be validated when submitted.

HTML Multimedia

Introduction

Multimedia on the web is sound, music, videos, movies, and animations. Multimedia comes in many different formats. It can be almost anything you can hear or see, like images, music, sound, videos, records, films, animations, and more. Web pages often contain multimedia elements of different types and formats.

Multimedia elements (like audio or video) are stored in media files. The most common way to discover the type of a file, is to look at the file extension. Multimedia files have formats and different extensions like: .wav, .mp3, .mp4, .mpg, .wmv, and .avi.

HTML Video

The HTML <video> element is used to show a video on a web page.

The controls attribute adds video controls, like play, pause, and volume. It is a good idea to always include width and height attributes. If height and width are not set, the page might flicker while the video loads. The <source> element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format. The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.

Example:

```
 <video width="320" height="240" controls>
      <source src="C:\Users\nischal.shrestha\Desktop\Nischal
Shrestha\mov_bbb.mp4" type="video/mp4">
      Demo video
</video>
```

HTML Audio

The HTML <audio> element is used to play an audio file on a web page. The controls attribute adds audio controls, like play, pause, and volume. The <source> element allows you to specify

alternative audio files which the browser may choose from. The browser will use the first recognized format. The text between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element.

HTML APIs

HTML DOM API

The HTML DOM API is made up of the interfaces that define the functionality of each of the elements in HTML, as well as any supporting types and interfaces they rely upon. The functional areas included in the HTML DOM API includes:

- Access to and control of HTML elements via the DOM.
- Access to and manipulation of form data.
- Interacting with the contents of 2D images and the context of an HTML <canvas>, for example to draw on top of them.
- Management of media connected to the HTML media elements (<audio> and <video>).
- Dragging and dropping of content on webpages.
- Access to the browser navigation history
- Supporting and connective interfaces for other APIs such as Web Components, Web Storage, Web Workers, Web Socket, and Server-sent events.

HTML Geolocation API

It is used to locate a user's position. The HTML Geolocation API is used to get the geographical position of a user. Since this can compromise privacy, the position is not available unless the user approves it.

The getCurrentPosition() method is used to return the user's position.

HTML Drag and Drop API

In HTML, any element can be dragged and dropped. Drag and Drop is a very common feature. It is when we "grab" an object and drag it to a different location.

First of all: To make an element draggable, set the *draggable* attribute to true.

HTML Web Storage API

HTML web storage; better than cookies. With web storage, web applications can store data locally within the user's browser. Before HTML5, application data had to be stored in cookies, included in every server request. Web storage is more secure, and large amounts of data can be stored locally, without affecting website performance. Unlike cookies, the storage limit is far larger (at least 5MB) and information is never transferred to the server. Web storage is per origin (per domain and protocol). All pages, from one origin, can store and access the same data.

HTML Web Workers API

When executing scripts in an HTML page, the page becomes unresponsive until the script is finished. A web worker is a JavaScript running in the background, without affecting the performance of the page. It runs in the background independently of other scripts, without affecting the performance of the page. You can continue to do whatever you want: clicking, selecting things, etc., while the web worker runs in the background.

HTML SSE API

Server-Sent Events (SSE) allow a web page to get updates from a server. A server-sent event is when a web page automatically gets updates from a server. This was also possible before, but the web page would have to ask if any updates were available. With server-sent events, the updates come automatically. Examples: Facebook/Twitter updates, stock price updates, news feeds, sport results, etc.

Conclusion

This report includes basic concepts of HTML which provides in-depth understanding of HTML structures, elements, attributes and many more. This documentation aims to provide a strong foundation for anyone who is trying to gain mastery in the front-end spectrum of development. For advance or further understanding in the field, CSS for styling, Javascript for dynamic structure is needed.