HTML5 & CSS3

A Step-By-Step Guide for Beginners to build and Design Responsive and Engaging Websites with HTML5 and CSS3

ANDREAS MAURER

HTML5 & CSS3

A Step-By-Step Guide for Beginners to build and Design Responsive and Engaging Websites with HTML5 and CSS3

ANDREAS MAURER

Table of Content

- Combined selectors and its types

Chapter 07:The Box Model

- The tag
- Internal & External distance
- Width and height
- The CSS Box Model

Chapter 08: Borders

- CSS Borders and Types of Borders
- Rounded Border with border radius
- Border-Collapse
- Border Styles

Chapter 09: Size indication

- Size and types of size
- Using appropriate size

Chapter 10: Types of Colors

- Introduction
- Predefined colors
- Hex Colors
- RGB () & RGBA () Notations
- HSL () & HSLA () Notations

Chapter 11: Fonts

- Font-family
- Font-style
- Font-size
- Font-Google
- Embedded/External fonts

Chapter 12:Inline & Block elements, images

- Inline vs Block Element
- The Display features
- Integrate images
- Page structuring in HTML5

Chapter 13: Positioning of element

- Introduction
- Absolute/ Relative position
- Fixed/stickey and static position

Chapter 14: Links in HTML

- Introduction to hyperlinks
- Styling Hyperlinks

- Link Bookmarks

Book Blurb

This book is about the main and basic concept of HTML & CSS which is required to become a front-end developer. The complexity of concepts in not much pro. Either you are beginner or you are developer. You can take help from this. The pre requisite of this book is nil either if you are layman you can understand the language of the book.

Structure of Book

Book is consisting of total 15 chapters from basic to pro levels followed by glossary. Every chapter is a prerequisite of next chapter. You cannot jump to next chapter if you are not good in concepts of previous chapter. Every chapter contain multiple section followed by sub section. Our main focus is on concept of the topic. The Book is different from other book as we have explained the concepts in detail followed by appropriate example. Also, this book contains no large content. Only the concepts which are helpful will be explain in book. So, there is no

wastage of time. Once you learn the basic concepts you can explore it by yourself easily. In sort this book is essential guide to learn html and as a beginner. Best of luck!

Chapter-01

Choosing Text Editor

In this Chapter, you will learn about

1.1: What is Text Editor and Why we need it?

1.2: Some Famous Text Editors

1.3: ATOM Text Editor and it's working

1.4: Installation of browser Google Chrome

Section 1.1: What is Text Editor and Why we need it?

Text Editor

A Text Editor is Computer program that permits you to create, open, view, and alter plain content documents. Text Editors are provided with Operating Systems and programming advancement bundles, and can be utilized to change files, for example, arrangement documents, documentation files and many other programming language source code. The simple text editor example is NOTEPAD which is built in software in windows. You can create, edit, view simple text files with notepad but notepad editor is not good enough for editing programming language source code.

A simple Text Editor have following features in it:

- cut, copy & Paste
- text formatting
- find & replace
- Undo & Redo

Why we need Text Editors?

Text Editors are used by a wide variety of purposes and by wide variety of people. Anyone who need to write, edit, or read text can simply use text editors like notepad. Software Programmers, App & Web developers use text editors to read, write & edit source code of many programming & Markup languages. Text editor use for source code is the primary purpose of text editors and there are many other features of text editing software are built to help these users read and write code. As we are going to learn HTML & CSS language so we also need text editor and we use ATOM text editor for this purpose.

Section 1.2: Some Famous Text Editors

Some editors are simple and have limited functions while some editors are offer wide range of functions and plugins. Windows operating systems come with the simple notepad, but software developers need more advance editors. Some famous Text Editors are given below:

- Sublime Text
- ATOM
- Emacs
- TextMate
- Notepad++

All these editors are used to write source code in many languages like C, C++, Python, JavaScript HTML&CSS and many more.

Section 1.4: ATOM Text Editor and it's Working

Atom is a free and open source text and source code editor windows, Linux and macOS developed by GitHub. It supports many plugins which add much more functionalities in editor. Atom allow users to install and use third-party packages and themes to customize the features and looks of the editor. As a default, ATOM's package can apply HTML, CSS, C, C++, Python, JavaScript and many other languages. As we are going to learn HTML & CSS, follow step guideline which are given below to download and install it.

1. Go to official atom.io website and download .exe file accordance with your operating system.
2. Run this .exe file once downloaded.
3. Go to Operating system menu bar and search ATOM
4. Click it to open it.

Congratulation! You have successfully download and install ATOM text editor to your OS. You will have many options on the right side of ATOM from where you can select your favorite theme, customize the font styling or font size and most importantly you can install different package from there. Some important package which you need for HTML & CSS **are atom live server, Emmet, autosave on change and pigments** . Go to Install package option and install these packages before writing HTML & CSS code. Furthermore, you can choose your best font type from side bar. Creating a file and write a code will be describe in chapter-03.

Section 1.4: Installation of browser Google Chrome

When we are writing a source code, we need to check the output of our code. As HTML & CSS are used for front-end web development so we have to check the code output on google chrome browser. You can check the output on another browser as well. Following is the quick guide to download and install Chrome browser in windows operating.

1. Go to official Google website and download chrome browser .exe file
2. Run this file as an administrator.

Go to your operating system setting and set the chrome as a default browser.

Chapter-02

Introduction to HTML & CSS

In this chapter you will learn:

2.1: What is HTML & CSS?

2.2: Use of HTML and CSS in Web Dev

2.3: Way to learn HTML and CSS

Section 2.1: What is HTML & CSS?

HTML

HTML stands from *Hyper Text Markup Language* for creating and describing web pages. It consists of a series of elements which tells the browser how to display the content. HTML is used by browser to manipulate text, images and other content to display it in user required format. HTML was created by **Tim-Berners-Lee in 1991** which was the first version named as HTML 1.0 but the standard version was published in 1991 which was HTML 2.0. Today there are five versions of HTML and we will learn fifth version i.e. HTML 5. HTML files has **.html** extension.

CSS

CSS stands for *Cascading Style Sheet* which describe how HTML elements are to be displayed on screen on in any other media for user view end. CSS saves our time by controlling the multiple pages layout at once. CSS allows us to apply styles on multiple pages independently with HTML. We can apply one CSS file to the multiple HTML files. CSS is easy to understand and learn in less time and it also provide powerful control on displaying HTML documents in browser. CSS file have ***.css*** extension.

Section 2.2: Use of HTML and CSS in Web Dev

Web development consists of two layers:

1. front-end layer
2. Backend layer

Front-end web development which is also known as user-side development means that of producing HTML, CSS for a Web Application so that a user can see and interact with them directly effectively. In front-end we have to care about the understanding level of the user. We have to build front-end pages by keeping in mind that client is a lay man user and cannot understand complex functionalities. The programmer of front-end dev is called frontend developer.

Back-end web development means to handle the behind-the-scenes functionality of website. It's code that connects the web to a database, manages user connections, and powers the web application itself. The language that is use for backend is python, c++ and many more. The programmer of back end dev is called backend developer.

HTML & CSS in Front-end web dev.

We will use HTML & CSS for front-end web development because of the following reasons:

- HTML is easy to use and understandable and does not have many complex syntax
- HTML is totally free of cost. All you need is the laptop or PC computer.
- Mostly HTML is all you need
- HTML is most search engine friendly.
- Most importantly all the browser supports HTML. You can run your source code in every browser

Similarly, CSS:

- **CSS Save a lot of time:** we can use one css file presentation on multiple HTML documents. Hence, we don't need to create another again for separate HTML files
- **Clean Coding Technique:** CSS is simple language and browser would not have much struggle to decode CSS file.
- **Easy Maintenance:** CSS can be maintained very easily. If you want to change the style or color on Multiple Blocks of same type then you have to simply update only one CSS

block.

- **Gives extra style to HTML:** CSS gives superior style to HTML documents. You can give a better presentation to you HTML file by using it instead of HTML attributes.

Section 2.3: Way to learn HTML and CSS

It is said that practice makes a man perfect so this also implies on HTML and CSS. The first thing you have to do is to understand the concept. But understanding concepts is not enough. You have to practice it with you hand. The more you practice the more you good in coding. Although, this book containing all the basic stuff but it is on you to explore the content in much deeper understanding. There are many online platforms to explore the content you can choose any of them.

Chapter 03

Getting Started with HTML

In this chapter, you will learn about

Section 3.1: The HTML Framework

Framework

Set of logical structure, rules, ideas and guidelines which is used to build something purposeful is called framework. These logical set of rules is to build the common and real-world problems. For example, in the process of making brick there is certain rules and regulations which we have to follow to make a brick. These rules and regulation are simply called framework.

The HTML Framework

In the programming world, a framework is a predefined set of classes and function which can be used to interact system software, to manage hardware devices process input/output.

In web development, as mentioned earlier, we have to standardize our code to user required output. For this there a lot of packages, files and folder structures predefined in HTML. These packages help us to build websites from basic to advance level.

The basic purpose of these framework is to save the time of programmer. The reason is that most of websites have similar structure with a bit difference. So, HTML provide us common structure which can be use again and again without going in deep.

Here we will describe the example of CSS framework. When we want to make a grid in CSS then CSS allows us to use many different elements which help the website to make it good. So, this is CSS framework.

Some famous HTML frameworks are Bootstrap, Angular and jQuery etc.

Section 3.2: What is doctype?

When we start writing HTML code on editor, we have to give Document type first. This document type tells the browser which type of code written in file. This document type is names as doctype in HTML. Every HTML document must start with doctype. The syntax to write doctype is below:

<!doctype>

We have to further specify the name and version of HTML in the above tag. There are five version of HTML tags. We discuss only two older version doctype for understanding

1. For HTML 1.1:

<!doctype html PUBLIC "-//W3C//DTD XHTML 1.1//EN">

2. For HTML 4.01:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

We will be using HTML 5 in this book. The Doctype of html is so simple. The syntax of html 5 doctype is following:

<!doctype html>

Html 5 is most the most using and latest version of html. That's why it's doctype is so simple and it is understood during compilation of file that this html means HTML version 5.

NOTE: HTML document type is not case sensitive. You can write it one of the following methods:

- <!DOCTYPE html>
- <!DocType html>
- <!Doctype html>
- <!doctype html>

Google Chrome, Microsoft Edge, Firefox, safari, opera Support the HTML doctype.

Section 3.3: Writing first "Hello World" program

We will now write our first "Hello World" program by using html 5. Simple create new file in ATOM write the following code, save it by keeping the extension of file *.html* and run it by using browser which we have installed (Chrome) Following is the "Hello world" program:

1. *<!DOCTYPE html>*
2. *<html>*

3. *<head>*
4. *<title>my first html program</title>*
5. *</head>*

6. *<body>*
7. *Hello world*
8. *</body> </html>*

This is basic and simple format of html code file. In the first line we declare the html doctype. From the 2nd line we write our first html tag which has closing tag at the end of file. We will describe remaining code in next section 3.4

Section 3.4: Elements /Tags & Page Tittle in HTML

The html tags are of two types user defined and pre-defined. Every tag can either be open or closing tag and the set of these tags is called html element. HTML Tags defines the basic structure view of page. Opening and closing html are both have a same syntax except for one slash "/". Tags are written with in less than & greater than i.e.

- <> Starting tag
- </> Ending tag

In the above example of hello world program <html>, <tittle>, <head> and <body> are the opening tags while </html>, </tittle>, </head>, </body> are closing tags.

NOTE: In html file both open and closing tags are must. We can put only one tag at a time. Every tag must have its correspondence tag.

Page Tittle

Every html file has its own page tittle which is show when we run the code. For this you have to write tittle tag in html code and give your own page tittle in that tag. For example, in above hello world program the page tittle is "my first html program "which is enclosed in <head> tag.

Every html document has its own body and head tag. Body defines the content structure in page. While head define the metadata of html file like page title etc.

Exercise

Q1: Write a html program to print your full name with your roll number with the page tittle of "Contact Info".

Ans:
```
<!DOCTYPE html>
    <html>
    <head>
  <title> Contact Info </title>
    </head>
    <body>
    <p> Name: John Kemble </h1>
    <p> Roll number: LE2345</p>
    </body> </html>
```

Chapter 04:

Headings, Comments, Paragraph & Text Formatting in HTML

In this chapter, you will learn about

4.1: HTML Attributes

4.2: Using headings

4.3: comments in html & CSS

Section 4.1: HTML Attributes

An HTML attribute is a name value pair that we put inside the opening html tag.HTML attributes gives the additional information about the element. Every HTML attribute has its name and value. For example, when we start our writing our html code then the first tag in our code is <html>. Suppose we want to specify the language of page in html tag. Then we write

<html lang=" en">

Here, lang is the attribute name and "en" stands for English. You can write "ar" for Arabic and "ur" for urdu and so on. Attribute value always come after equal operator and enclosed in single or double quotes.

ID Attribute

Id attribute is just like your roll number. This is unique for all HTML elements. The syntax for adding id attribute is following:

*** First paragraph> ***

*** First paragraph> ***

These two lines show the two same elements of html code. But these two elements have different ids. So, while adding some CSS fonts or adding additional information in these elements we can access these elements with unique ids.

Class Attribute

The id is just for one element but class is for group of elements just like your roll number and your class id. The syntax is same as the id attribute:

*** Apple ***

*** Mango ***

*** Orange ***

There are 3 different elements but they have same class named as Fruits. We can access these elements whose class is fruits.

Some Important Attributes

There are a lot of attributes in html. Following is the list of most using attributes:

- **Tittle Attribute** This is line

Tittle attribute is used to give the tittle name to the element. When we move our cursor to the element, browser will show that specific tittle.

- **Style Attribute** This is line

Style attribute is used to manage the style of element. We will discuss it in CSS section in detail.

- **Href Attribute** <a Href="Some page link">Welcome

Href attribute is used inside hyperlink tag <a>. Href take us to the specified link when we click on it.

- **src/alt** <img src="img_typo.jpg" alt="Girl with a jacket">

src attribute is use for Source. Src attribute can be used without alt attribute but when sometime picture is not available then the text given in alt attribute will be shown there instead of image. We will discuss in detail in image section.

Section 4.2: Using Headings

Headings are basic part of the page which define the basic structure of web page. When user visits any website the first thing, they look at is the headings of the content. For this purpose, HTML 5 provide us Heading tag from h1 to h6. Heading tag are placed with in the body tag in html code. Following is the syntax and output for all six type of headings.

- H1 <h1> Heading h1 </h1>
- H2 <h2> Heading h2 </h2>
- H3 <h3> Heading h3 </h3>
- H4 <h4> Heading h4 </h4>
- H5 <h5> Heading h5 </h5>
- H6 <h6> Heading h6 </h6>

The output of above tags is:

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

NOTE: The size of these six type headings remain same. However, you change it by using CSS Style attribute font-size. HTML heading are to use for making text bold or big. It is only for heading.

Section 4.3: Comments in html & CSS

When we write code in any language, then we give comments to our code which is not actually the part of code but coder convivence. Similarly, in HTML there is also comment tag available for programmer convivence. The content which is written inside the comments will not be shown in the web page. The syntax of giving comments is following:

*** This is paragraph ***

<! -- This is a comment-->

*** This is paragraph ***

The output of this code will be:

This is paragraph

This is paragraph

NOTE: You can write whatever you want in comment section. But the main purpose is to write the information regarding code.

Section 4.4: HTML Paragraph & Line Breaks

Paragraph

Writing in paragraph will increase the look of page. Every paragraph starts from new line also.HTML-5 provide us paragraph tag which automatically add new line and some white when we use it. In simple words you can say it is block of text. The syntax to write paragraph tag is:

*** You are learning HTML ***

*** You are learning CSS ***

The output of above code will be:

You are learning HTML

You are learning CSS

NOTE: As you are learning Paragraphing in html. There is another tag in html which will provide page breaking in we page. This tag is <hr>. This <hr> tag has no closing tag. This tag simply add line in page. For example:

*** You are learning HTML ***

<hr>

*** You are learning CSS ***

The output of above code will be:

You are learning HTML

You are learning CSS

This line indicates the page break.

Line Break

When we write two paragraphs then it is understood that second paragraph will start from new line. But what if want line break within the paragraph? For this purpose, html-5 provide us
 tag which have no closing tag just like <hr> tag. The example is following:

*** Hello class fellows
welcome to HTML Paragraphing
 Hope you are all fine ***

The output will be:

Hello class fellows

Welcome to HTML paragraphing

Hope you are all fine

NOTE: The tag which has no its closing tag is called empty tag like
, <hr>. Line break tag is useful where we poems, quotes etc. Moreover,
 is not for adding space between paragraphs it is only for adding line break.

Section 4.5: Highlighting Text

To highlight any content or text in web page HTML provide us tag named as mark tag. Mark tag is not an empty tag. The text you to want to highlight is written inside this mark tag. The example of mark tag is following:

*** Hello everyone! Hope you like <mark>welcome party</mark> ***

The output will be:

Hello everyone! Hope you like welcome party

NOTE: Mark value set the background to yellow and text color to black as default highlight. But you change the using CSS Mark attribute:

```
mark {
    background-color: yellow;
    color: black;
}
```

You can change the value yourself.

Section 4.6: Bold, Italic & Underline

Bold

HTML provide us tag to bold the text. is not an empty tag. See following example:

*** This is HTML and CSS Book ***

The output will be:

*This is **HTML** and **CSS** Book*

We can also use CSS tag span with attribute style to bold the text i.e.

*** hello everyone Hope you are all fine text. ***

The output will be:

Hello everyone **Hope you are all fine**

NOTE: There are also some other tags available in html for text formatting like tag for emphasizing the content, <small> for smaller text and for strong text. You can explore these tags by yourself.

Italic

To make text italic there exist a tag names as which is also not an empty tag. The text you want to make italic put inside the tag. For example:

** This is italic text example **

The output will be:

This is italic text example

Underline

There exists <u> tag for underlining the text. The example for underlining text is:

** This is <u>underline</u> text **

Output: **This is <u>underline</u> text**

In simple words, you can remember that b is for bolding the text, I for italic text and u for underline the text just like in Microsoft office.

Section 4.7: Abbreviation

For adding abbreviation in of any word like HTML, CSS, WHO we use <abbr> tag with attribute named as style. See following two examples:

The <abbr title="World Health Organization">WHO</abbr> is an international organization

The <abbr title="Hypertext Markup Language">HTML</abbr> stands for hypertext markup language.

Output: WHO is an international organization

HTML stands for hypertext markup language

You can also give to any other website page in tittle and set method to on click. So, when user click on that page. The link will take him/her to the desired page.

Section4.8: Deleted, Inserted or stricken

Deleted

For deleting the text on page, we use <del> tag which is also not an empty tag. For example:

*** My favorite coding language is ~~c++~~ ***

Output: ***My favorite coding language is ~~c++~~***

Inserted

After deleting text if we want to show inserted text then we use <ins></ins> tag. for example

*** My favorite coding language is ~~c++~~<ins>HTML<ins> ***

Output: ***My favorite coding language is ~~c++~~ HTML***

The inserted text will be shown as an underlined text.

Stricken

The Stricken text in html shown that the given information(text) in no longer correct or have expired. For this purpose, we use <s></s> tag in html. Example is below:

*** <s>you should try html </s> ***
*** try CSS Now ***

Output:

~~You should try html~~

Css Now

Section 4.9: Subscript & Superscript

During designing web page, we need to write some text as subscript and some text as superscript. Subscript means below the line and superscript means upper side on the line. For this purpose, html provides us and tag respectively i.e.

*** An example of _{subscripted} ***

*** An example of^{superscripted} ***

Output:

An example of ^{superscripted}

An example of $_{subscripted}$

Exercise

Q1: Write a command to set HTML File language to Arabic.

Ans: <html lang=”ar”>

Q2: Write a HTML class “HTML” and put 3 students in it.

Ans: *<p class =” HTML”> Jason Roy *

*<p class =” HTML”> JOHN *

*<p class =” HTML”>William *

Q3: Write a HTML Code in which you use line break, Bold, underlined, subscript and inserted text.

Ans: * Hi everyone!
 I am Living in London and studying from <u>Howard</u>. Now I am ~~18~~<ins>19<ins>. And I am using <sub>html<sub> now *

Chapter 05

CSS Basics

In this chapter, we will learn about:

5.1: CSS Introduction

5.2: Inline Stylesheet

5.3: Internal Stylesheet

5.4: External Stylesheet

5.5: Styling list with CSS

5.6: Define and implementation of table

5.7: Styling table with CSS

Section 5.1: CSS Introduction

CSS means that Cascading Style Sheet. Adding color in the background, change the font-size of text, Change the fore and background color of text, select appropriate devices and screen size,

how much space we want to place between the text is all done by using CSS and sheet which is containing all this information is called style sheet.

CSS can be added in html in three types.

1. Inline Style Sheet
2. Internal Style Sheet
3. External Style Sheet

Now, we will explain all these style sheets in coming sections.

Section 5.2: Inline Stylesheet

Adding CSS style inside html line is called inline styling. Inline styling uses CSS attribute named as style. Following is the example of using inline style sheet. Inside paragraph tag.

*** Inline stylesheet ***

Output:

Inline stylesheet

Section 5.3: Internal Stylesheet

As we have described earlier, there are two parts of html file i.e. head and body part. we add style sheet in the body section. But when we add style sheet in head tag by using reference of elements then it will become internal style sheet. See following example:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>

<h1>heading example</h1>
<p>paragraph example</p>

</body>
</html>
```

OUTPUT:

heading Example

Paragraph example

When we give elements name in internal style sheet. It will automatically detect the styles and add apply these specific styles on that elements.

Section 5.4: External Stylesheet

As obvious from name, that external means a style sheet outside the html coding file. Actually, when we lengthy codes then it difficult to add inline or internal style sheet in the html because the code become more complex in this way. So, we make a separate file of style sheet extension as .css (CSS file extension) and link this file to our html file and this way our code become comprehend and understandable. Suppose, we make a file of style sheet named as ExternalStylsheet.css then we add styles in html file in following way:

ExternalStylsheet.css:

```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

HTML Code

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="Externalstylesheet.css">
</head>
<body>

<h1>Heading Example</h1>
<p>paragraph Example</p>

</body>
</html>
```

OUTPUT:

Heading Example

Paragraph example

NOTE: We have used rel, href and link attribute in head tag. We will explain it briefly in last section where we learn how CSS file link with html file.

NOTE: We have seen only color and background color example but you can explore more attributes like font size, font color etc.

Section 5.5: Styling list with CSS

Styling list means giving bullets, name and letters to some specified list. For example:

1. Apple
2. Banana
3. Mango

This is list with styling by numbers. CSS provide us two types of styling list. Unordered styling list and Ordered Styling list.

Unordered Styling List

In unordered styling list, we mark the list with bullets, images and smileys etc. The CSS attribute for this purpose is <ul>. The method is to write ul and then dot operator with class or element value in html file. The syntax is following:

```
ul.x {
        list_style_types: y          }
```

where x is class or element value used in html file and y is the type of styling means circle or square etc.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
ul. cir {
  list-style-type: circle;
}
ul. Sir {
  list-style-type: square;
}
</style>
</head>
<body>
```

```
<p>Unordered list example:</p>
<ul class="cir">
  <li> Apple </li>
  <li> Banana </li>
  <li> Mango </li>
</ul>
<ul class="b">
  <li> Apple </li>
  <li> Banana</li>
  <li> Mango</li>
</ul>
<p>Example of ordered lists:</p>
</body>
</html>
```

OUTPUT:

Unordered list example:

- Apple
- Banana
- Mango

- Apple
- Mango
- Banana

NOTE: Here we use a tag which used to define a list in html.

List Style Image:

List style image attribute is used to mark the list by image. Following is the syntax:

```
ul {
  list-style-image: url('sqpurple.gif');
}
```

you can add other picture in url by giving the name with extension .gif.

List Style position

List style position is used to define the position of marking the list. There are two types of list style position which inside and outside. The syntax is following:

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

```
<head>
<style>
ul.out {
  list-style-position: outside;
}
ul. b {
  list-style-position: inside;
}
</style>
</head>
<body>
<h2>list-style-position: outside (default):</h2>
<ul class="out">
  <li>Coffee – you are learning<br> outside </li>
  <li>you are learning outside </li>
</ul
<ul class="ins">
  <li> you are learning<br>inside styling list </li>
  <li>you are leaning inside styling list </li>
</ul>
</body>
</html>
```

OUTPUT:

OUTSIDE:

- you are learning
  outside
- you are leaning outside.

INDSIDE:

- you are leaning

inside styling list

- you are learning inside styling list

Ordered Styling List

In ordered styling list, we use letters and numbers to mark our list. The CSS attribute which is used for this purpose is <ol>. The syntax is same as unordered. We just give concise example here. You can write it with html:

```
ol.c {
  list-style-type: upper-roman;
}

ol. d {
  list-style-type: lower-alpha;
```

}

Suppose we have written the input which we have used in above example then output will be:

I. Apple
II. Banana
III. Mango

a. Apple
b. Banana
c. Mango

Styling with Colors

We can also mark the list with different colors. The syntax is following for both Ordered and unordered list.

```
ol {
  background: #ff9999;
  padding: 20px;}

ul {
  background: #3399ff;
  padding: 20px;
}

ol li {
  background: #ffe5e5;
  padding: 5px;
  margin-left: 35px;
}

ul li {
  background: #cce5ff;
  margin: 5px;
}
```

You can run it in browser with html file to check the output.

NOTE: There is a default setting in list_image_type attribute. You can remove these default setting by giving value none to this attribute and 0 value to padding and margin.

List Style:

There exists a tag named as list style in css to apply list style on all the file:

```
ul {
 list-style: square inside url("image.gif");
}
```

This style will be applied to all the domain of file.

Description List

Description list is used to define the list with our own text bulleted. **<dl>** tag define description list, **<dt>** tag define the bullet text and **<dd>** tag define description data for example:

Example:

```
<dl>
  <dt> Apple </dt>
  <dd> Apple is gracious fruit</dd>
  <dt> Banana </dt>
  <dd> Banana is useful for bones </dd> </dl>
```

OUTPUT:

Apple

Apple is gracious fruit

Banana

Banana is useful for bones

Section 5.6: Define and Implement Table

Html provide us **<table>** tag to create a table. Every table has its rows, cells and headers.

- For table header **<th>** tag is used
- For table row **<tr>** tag is used
- Each data cell is defined with **<td>** tag

Example:

```
<table>
 <tr>
  <th> Name</th>
  <th> RollNum </th>
 </tr>
 <tr>
  <td> John </td>
```

```
    <td> 1243 </td>
  </tr>
  <tr>
    <td> William </td>
    <td> 8765 </td>
  </tr>
</table>
```

OUTPUT:

Name	RollNum
John	1243
William	8765

NOTE: As a default, table header stored in centered align and bold style while table data are not bold and left aligned.

Table Name & Table Id

<caption> tag is used to give the table name inside <table> name. The syntax is following:

```
<table>
    <caption> Table name </caption>
<table>
```

For uniquely identifying table we use **id** attribute. The syntax is:

```
<table id =" table is" > </table>
```

Section 5.7: Styling tables with CSS

For CSS attributes to style the tables:

- **Border** attribute is used to apply border on table

```
table, th, td {
    border: 2px black;
}
```

- **Collapse border** is used for collapsing border

```
table, th, td {
    border: 2px black;
    border-collapse: collapse;
}
```

- **Border-spacing** is used for adding spacing between columns

```
table {
    border-spacing: 5px;
}
```

- **Text-align** is used for text alignment

```
th {
    text-align: right;
}
```

- **Padding** is used for padding the cells data

```
th, td {
padding: 15px;
}
```

- **Rowspan** is used to merge the rows

  <th rowspan= "2"> two rows merged </th>

- **Colspan** is used to merge the columns

  <th Colspan= "2"> Two columns merged </th>

NOTE: We can also define sperate block for styling by using table id.

Exercise

Q1: Write a code to create ordered list of 3 fruits starting with number 40, In roman letters, in uppercase letter.

Ans:

```
<ol start="40">
<li> Apple </li>
<li> banana </li>
<li> Mango </li>
  </ol>

  <ol type="i">
<li> Apple </li>
<li> banana </li>
<li> Mango </li>
  </ol>

  <ol type="A">
<li> Apple </li>
<li> banana </li>
<li> Mango </li>
  </ol>
```

Q2: Write a html code to create a table of two columns with table header Country and Capital and then write CSS code by using external style sheet with id of table. Border should not be collapsing, Border color should be red, Internal distance would be 5px, Table header must be left aligned.

Ans: <! - -**HTML CODE - - >**

```
<table id =” q2”>
<tr>
<th> Country </th>
        <th> Capital <th>
</tr>
<tr>
<td> Pakistan </td>
<td> Islamabad </td>
</tr>
<tr>
<td> Australia </td>
<td> Sydney </td>
</tr>
</table>
<! -- CSS File - ->
#q2 {
Border: 2 px red;
Border-collapse: NONE;
Padding: 5px
Text-aligned: left;
}
```

Chapter 06

CSS Selectors

In this chapter, you will learn about:

Section 6.1: Introduction to CSS Selectors

CSS Selectors is one of the basic things that a beginner should learn first. This is the core feature of CSS. CSS Selector are used to apply styles, font-size simply you can say rules to html elements, class and attributes etc. CSS Selector apply rules by selecting the specific Element from html. The syntax to write selector is following:

```
Selector
{
    Property: Value;
}
```

Where Selector is may be the html elements, tag, class or id. Property is the Elements you want to apply rules i.e. color, size, font family. and value is how would you like to change the style of element. There are many types of CSS selectors which we will define in next sections:

Section 6.2: Simple CSS Selectors

Simple CSS Selector apply the rule by selecting the html elements which is based on their id, name or class: For example:

```
<head>
<style>
p {
    color: red;
    text-align: center;
}
</style>
</head>
<body>
<p> Hello Everyone you learning CSS Selector </p>
<p> I hope you are all fine! </p>
</body>
```

OUTPUT:

Hello Everyone you learning CSS Selector

I hope you are all fine!

As p is the tag to write html paragraph so all the paragraph in html file will be changed to red color and their alignment would be center.

NOTE: We have ignored some tag in above tag just to give you precise example.

Section 6.3: Attribute Selectors

CSS attribute selector change the style on the basis the html attributes. This Select the HTML element and apply colors and font styles on it. The syntax is:

```
a[attribute_name] {
  background-color: red;
   text-align: left;
}
```

 attribute blank selector

 Attribute Top Selector

OUTPUT:

attribute blank selector

Attribute Top Selector

This piece of code selects the element named as a and apply the red color and make the alignment left on attribute named as attribute_name.

```
a[target="CSS3"] {
  background-color: yellow;
   text-align: left;
}
```

if this code is applied on the above html code then the output will be:

attribute blank selector
Attribute Top Selector

This CSS code select all element of "a" and apply defined rules on attribute CSS3

Section 6.3: Type Selectors and Inheritance

Inheritance

Inheritance is nothing but a parent-child relationship. Child inherit the properties of parent. This property achieved in html by using division tag . See following example:

```
<div id=" maintag">
        <p> this is the paragraph p1 </p>
        <h1> This is heading h1 </h1>
        <p> This is paragraph p2 </p>
</div>
```

In this example, the parent tag is div with id maintag and p & h1 are the child tag. The rules defined in parent tag will be inherited by all its child tag.

NOTE: There are some tag which does not inherit properties from parent for example anchor <a> which is used to provide link in the page.

Type Selectors

Type selectors apply rules on all the nodes in html file by identifying the node name. For example:

A {

Define some CSS rules; }

This will apply rules on elements in html whose node name is A

Section 6.5: Combined Selectors

Combinators

To select more than one id or class elements in CSS we use combinators which define the relationship between the selectors. Combinators means to define the combine multiple selectors at a time. There are four types on combined selectors:

1. Child Selectors
2. Descendant Sectors
3. Adjacent sibling selector
4. General sibling selector

Child Selectors

Child selector will select only the first Childs to define CSS rules. If we use nested inheritance and apply CSS on outer tag. Then the specified rules will apply on inner elements of first div tag or outer elements of second div (or any other) tag. The syntax is:

div > p {

Define some CSS rules; }

These defined CSS rules will on the first child of div tag. For example:

```
<head>
<style>
div > p {
  background-color: red;
  text-align: left;
}
</style>
</head>
<body>
<div>
  <p>First Paragraph</p>
  <p>Second paragraph</p>
```

<section> Third paragraph </section>

* </body>*

OUTPUT:

First paragraph

Second paragraph

Third paragraph

Nth-child Concept

CSS3 provide us some built in functions and rules for our connivance. Nth-child is used to apply rules on some selected elements. See following syntax:

- **p: nth-child(n):** nth-child(n) is used to apply on the child at n position where n is 1,2,3, 4, … For example, nth-child (1) will select only first child of parent.2 will select only second child and so on.
- **P: nth-child(odd):** This will select the odd number child i.e. 1,3,5, … and apply rules on it.
- **P: nth-child(even):** This will select even number child i.e. 2,4,6, … and apply rules on it.
- **P: nth-child(an+b):** If you want to select your sequence of child then use nth-child(an+b) where a&b is 0,1,2,3, … for example nth-child(4n+0) will select multiple of fours child.

Descendant Sector

The descendant selector will select all matching tags inside parent irrespective of nested tags. The syntax is simple i.e.

Div p {

Some CSS rules; }

Example:

div p {

background-color: red;

text-align: left;

}

 First Paragraph

 Second paragraph

<section> *Third paragraph* </section>

OUTPUT:

First paragraph

Second paragraph

Third paragraph

Adjacent sibling selector

Adjacent sibling selector select the immediate sibling after the parent tag. The syntax is:

div+p {

Some CSS rules; }

Example:

```
div + p {
  background-color: red;
  text-align: left;
}
<div>
  <p> First paragraph </p>
</div>
<p> second paragraph </p>
<p> Third paragraph </p>
```

OUTPUT:

First paragraph

Second paragraph

Third paragraph

General sibling selector

General sibling selector select all the sibling elements of specified parent tag. The arithmetic operator use in general sibling is ~. The syntax is:

div ~ p {

Some CSS rules; }

Example:

```
div + p {
  background-color: red;
  text-align: left;
}
<div>
  <p> First paragraph </p>
</div>
<p> second paragraph </p>
<p> Third paragraph </p>
```

OUTPUT:

First paragraph

Second paragraph

Third paragraph

Summary of Combinators

After creating parent child relationship, the following tag with their arithmetic operator is:

1. Child Selectors (>)
2. Descendant Sectors (white space)
3. General sibling selector (~)
4. Adjacent sibling selector (+)

Some Other Selectors

Here, we discussed only number of selectors but there are many other selectors like universal selector which select all the elements in html file, grouping selectors which make a group of two selectors and then filter out elements from that group, Column combinators select all the specified columns. You have to explore these selectors yourself.

Exercise

Q1: Write a code to on three html elements, one should be heading and other two are paragraph and then set the color of paragraph to red.

Ans:
```
<style>
    p {
        color: red;   }
</style> <body>
<h1> Heading </h1>
<p> paragraph1 </p>
<p> Paragraph2 </p>  </body>
```

Q2: Write a code to on four html elements (h1, h2, p, p) and then by using selector set the h1, p, p color to red.

Ans:
```
<style>
h1, p {
 color: red;   }
</style> <body>
<h1> heading1 </h1>
<h2>heading2</h2>
<p>paragraph1</p>
<p>paragraph2</p>
```

Chapter 07

The Box Model & The div

In this chapter you will learn about,

7.1: The Tag

7.2: Internal & External distance

7.3: Width and height

7.4: The CSS Box Model

Section 7.1: The Tag

Div stands for divison which is used to create a block or section of html elements. It is the divison of elemnts contained html elements which is used to define CSS rules in CSS file like we use in previous chapter. It is a non empty tag.you can create nested div by giving ids or class type ids to each of them. The inside elements inhrits the properties of div tag (parent tag inheritence concept). In defualt, there will be line break when we inserted new div block. The Syntax is:

```
<div id="name">
    <p> paragraph </p>
    <h1> heading </h1>
    <div id="name1">
    <p> paragraph2 </p>
    </div>
</div>
```

NOTE: By using div tag we can remove the default setting of CSS by giving value block i.e.

```
div {
    display: block;     }
```

This will block all the default CSS Fonts and other settings.

Section 7.2: Internal & External Distance

Internal Distance (Padding)

Internal distance is referred padding in CSS. By using the padding properties, we insert, increase the internal space of the content. CSS provide us full control on padding. Padding is of four types:

1. Padding top
2. Padding bottom
3. Padding left
4. Padding right

We use padding properties inside div tag (you can also use in other tags as well) as the div block can be inherited by its child easily. For example:

```
div {
  padding-top: 50px;
  padding-right: 30px;
  padding-bottom: 50px;
  padding-left: 80px;
}
```

It will set all div elements content with these internal spacing.

Ways of written padding attributes

Padding inside any can be written in four ways.

1.
```
div {
  padding: 25px 50px 75px 100px;
}
```

if the padding has four values, then it will be top, right, bottom and left respectively i.e.
in this example:
top value=25px, right value=50px, bottom value=75px, left value= 100px

2.
```
div {
  padding: 25px 60px 90px;
}
```

if the padding has three values, then first value is top value, second value is for both right and left value and third value is for bottom value i.e.
in this example:
top value=25px, right value=60px, left value= 60px, bottom value=90px

3.
```
div {
  padding: 25px 60px;
}
```

if the padding has two values, then first value is for top and bottom value and second value is for both right and left value i.e.
in this example:
top value=25px, bottom value=25px, right value=60px, left value= 60px

4.
```
div {
  padding: 45px;
}
```

if the padding has one values, then this value is for all top, bottom, right and left value
in this example:

top value=45px, bottom value=45px, right value=45px, left value= 45px

External Distance (Margin)

External distance is referred Margin in CSS. By using the Margin properties, we create space outside the content of elements. Like padding, Margin is also of four types: i.e. top, right, bottom and left. The syntax is also same as padding:

```
div {
  Margin-top: 60px;
  Margin-right: 20px;
  Margin-bottom: 50px;
  Margin-left: 10px;
}
```

Ways of written Margin attributes

Padding inside any can be written in four ways.

1.
```
div {
  Margin: 30px 60px 80px 90px;
}
```

if the padding has four values, then it will be top, right, bottom and left respectively i.e.
top value=30px, right value=30px, bottom value=80px, left value= 90px

2.
```
div {
  Margin: 25px 50px 75px;
}
```

same like padding if margin has three values then:
top value=25px, right value=50px, left value= 50px, bottom value=75px

3.
```
div {
  Margin: 30px 60px;
}
```

First value is for top and bottom, second value is for right and left.
top value=30px, bottom value=30px, right value=60px, left value= 60px

4.
```
div {
  Margin: 30px;
}
```

same values of all attributes:

top = 30px, bottom = 30px, right= 30px, left = 30px

NOTE: we can also use auto value instead of giving custom value in Margin or we can also inherit values by using keyword inherit in place of custom value i.e.

```
div {
  margin: auto;
}
div {
   margin-left: 200x;
}
h1.ex1 {
  margin-left: inherit;
}
```

Section 7.3: Width and height

The width and height are used to set the spacing within the block of the element. Padding and margin is not including in width/height properties. It only set the height/width of element inside the padding.

The height and width can have one of the following properties.

- **Custom length:** We can give our own custom length in px, cx etc. for example:
  ```
  div {
  height: 150px;
  width: 100px;
  background-color: red;
  }
  ```
- **Percentage:** we can give percentage value in height/width attribute. Browser will calculate the percentage of element containing block.
  ```
  div {
    height: 25%;
    width: 25%;
  }
  ```
- **Auto/inherit:** The value of width and height can be auto (browser will set it) or inherited (inherit from parent).
  ```
  div {
    height: auto;
    width: auto;
  }
  ```
- **Max-width/height:** If we set the values of height/width very large then there is a possibility that it goes out from screen and then browser add horizontal Scroll. To overcome on this problem, we use max-width and max-height attribute.
  ```
  div {
  max-width: 400px;
  max-width: 300px;
  background-color: red;
  }
  ```

Section 7.5: The CSS Box Model

Every html has four properties i.e. margin, border, padding and content. Including all these attributes html elements logical view considered as box. This box is called CSS box. We then apply CSS rules on this box. Following is the logical view of CSS Box.

1. **Margin:** Outer layer of element and transparent.
2. **Border:** Border is not a transparent and outside the padding
3. **Padding:** Padding is inside spacing of html element.
4. **Content:** The text, pictures, videos anything you want to insert in page.

Example

```
div {
  width: 150px;
  border: 12px solid red;
  padding: 25px;
  margin: 0px;
}
```

Calculation of total height/width of an element

The total html element width can be calculated by using the below formula:

Total width= width+ padding-left + padding-right + border-left + border-right + left-margin + right-margin

Similarly, Total height is:

Total height = height + padding-top + padding-bottom + border-top + border-bottom + margin-top + margin-bottom.

Exercise

Q1: write a code to set background color red and width 100 px.

Ans: div {

  background-color: lightblue;

  width: 200px;

}

Q2: Write a CSS Code to set the margin, padding and border to 30px each and width should be 100px.

Ans:
```
div {
  width: 100px;
  padding: 25px;
  border: 25px solid navy;
  margin: 25px;
  }
```

Chapter 08

CSS Borders

In this chapter, you will learn about:

8.1: CSS Borders and Types of Borders

Section 8.1: CSS Borders and Types of Borders

CSS provide full control to set the borders of html elements. The syntax to set the border of elements is following:

NameOfElement {

Border: Value; }

Heading 5 in red table

example:

Paragraph in double border table

```
h5 {
  border: 5px solid red;
}

p {
  border: double;
}

<h5> Heading 5 in red table </h5>

<p> Paragraph in double border table </p>
```

There are four types of table:

1. Border top
2. Border Bottom
3. Border right
4. Border left

Border-Top

The attribute to set the upper border of html element is border-top i.e.

Heading 3 in top border

division in dotted top border

```
h3 {
  border-top: 4px dotted yellow;
}

div {
  border-top: 4px dotted blue;
}

<h3> Heading 3 in top border </h3>
```

* division in dotted top border *

Border-Bottom

The attribute to set the upper border of html element is border-bottom i.e.

Heading 4 in top border

division in dotted top border

```
h4 {
  border-bottom: 4px dotted yellow;
}

div {
  border-bottom: 4px dotted blue;
}
<h3> Heading 4 in top border </h3>
<div> division in dotted top border </div>
```

Border-Left

The attribute to set the upper border of html element is border-left i.e.

Heading 4 in left border

division in dotted l eft border

```
h4 {
  border-left: 4px dotted yellow;
}

div {
  border-left: 4px dotted blue;
}
<h3> Heading 4 in top border </h3>
<div> division in dotted top border </div>
```

Border-right

The attribute to set the upper border of html element is border-right i.e.

Heading 4 in Right border

division in dotted r ight border

```
h4 {
  border-right: 4px dotted yellow;
}

div {
  border-right: 4px dotted blue;
}
<h3> Heading 4 in right border </h3>
<div> division in dotted right border </div>
```

NOTE: Making borders one by one is not a good practice. We have use it for understanding but we can add all the borders and their values only in one CSS tag: i.e.

Heading h4

```
h4 {
  border-top: 4px dotted red;
border-bottom: 4px dotted blue;
border-left: 4px dotted yellow;
border-right: 4px dotted black;
}
<h4> Heading h4 </h4>
```

Section 8.2: Rounded Border with border radius

There are four types of border with respect to radius:

1. Normal Border
2. Round Border
3. Rounder Border
4. Roundest Border

Normal border has corner of angle of 90. The remaining three types are bases on the how much the border is round.

The attribute to set the border radius in CSS is border-radius. See following example:

```
p1 {
  border: 2px solid red;  }
div {
  border: 2px solid red;
  border-radius: 5px;
}
```

Multiple ways to written radius value:

1. **Border-radius 25px 30 px 2px 4px**

In case of four values, first value is for top left corner, second value is sets top-right corner, third value sets bottom-right corner and the fourth value is for bottom left corner of the element.

2. **Border-radius 5px 6px 4px**

In case of three values, first value is for top left corner, second value is sets top-right and bottom-left corner and the fourth value is for bottom right corner of the element.

3. **Border-radius 5px 6px**

First value is for top-left & bottom-right corner and second value sets the value of other two corners.

4. **Border-radius 5px 6px**

In case of only one value is given, all the corners of border value will be same to which value is provided.

Section 8.3: Border collapse

Border collapse means to select the border of html elements separated or collapse into a single a single border. The CSS attribute use for this purpose is border collapse. Following is the use of border collapse:

```
table, td, th {
  border: 1px solid black;
}
#SeperateTable {
  border-collapse: separate;
}
#CollapseTable {
  border-collapse: collapse;
}
<h2> Separate Border </h2>
<table id="SeparateTable">
  <tr>                <! --tr tag is for table row-->
    <th> Name </th>            <! --th tag is table header -->
    <th> Roll No </th>
  </tr> <tr>
    <td> John </td>             <! --td tag is for table data in cells -->
    <td> 1234 </td>
  </tr> <tr>
    <td> Hamas </td>
    <td> 3452 </td>
  </tr>
</table>
<h2> collapse Border </h2>
<table id="CollapseTable">
  <tr>
    <th> Name </th>
    <th> Roll No </th>
```

```
  </tr> <tr>
    <td> John </td>
    <td> 1234 </td>
  </tr> <tr>
    <td> Hamas </td>
    <td> 2345 </td>
  </tr>
</table>
```

Output:

Separate Border

Name	Roll No
John	1234
Hamas	2345

Collapse Border

Name	Roll No
John	1234
Hamas	2345

Note: We can also set the Border-collapse value to initial or inherit from parent. Moreover, we can give the border spacing in separate borders like:

```
#TableName {
  border-collapse: separate;
  border-spacing: 5px;
}
```

Section 8.4: Border Styles

The attribute border-style is used to set the style of four border of html elements. The CSS syntax for border-style is:

border-style: none|hidden|dotted|dashed|solid|double|groove|ridge|inset|outset|initial|inherit;

These are all types of border. Border-styles can be written in one line or more than in one line (Separate line for each four borders). In case of one line, there are four possibilities:

1. ***border-style: dotted solid double dashed;***

In this case, top border will be dotted, right border will be solid, bottom border is double and left border will be dashed.

2. ***border-style: dotted solid dashed;***

top border will be dotted, left & right border will be double and bottom border will be dashed

3. ***border-style: dotted solid;***

Top & bottom borders will be dotted and remaining two borders will be solid.

4. ***border-style: dotted;***

All the four borders will be dotted.

Border Color

Border coloring is one of the major parts of border styling. We can style every border with our own color by using CSS attribute border-color. For example:

```
div {
border-left-color: transparent;
border-right-color: red  }
```

we can also give values in gba or rgba which we will discuss in next chapters.

Border Image

We can set the border style with image using CSS attribute border-image for example:

```
#htmlElement {
  border-image: url(border.png) 30 rounds;
}
```

There are many attributes which is actually extension of border-image like border-image-source, border-image-width, border-image-slice and border-image-outset. You can explore it by yourself.

Exercise

Q1: Write A CSS block code to set the dotted border around html element only on to the top side, the color should be black and 2px in width.

Ans:
```
p {
border-top-style: dotted;
border-top-width: 2px;
border-top-color: black;
}
```

Q2: Write a command to set the border to 10px solid green.

Ans: border: 10px solid green;

Chapter 09

Size Indications

In this chapter, you will learn about

Section 9.1: Size and types of Size

Size

While cascading HTML elements with CSS the most of CSS attribute require length in their value which is called size of that attribute for example border, font-size, padding, margin, width and height require the size. The size consists of two components first one is magnitude and second its unit. There would be no space between magnitude and unit and we cannot write only magnitude without unit because there is not any default unit in CSS. The only exception is in case of 0 magnitude. No unit require with zero size. There are two types of size in CSS:

1. Absolute Size
2. Relative Size

NOTE: Size and length are used interchangeably.

Absolute Size

Absolute size is fixed and does not depend on other elements size. Following units are used for absolute size:

- cm (centimeter)
- mm (millimeters)
- in (inches) ▯ ▯ 1in = 2.54cm
- px (pixels) ▯ ▯ 1px = 1/96 of 1in
- pt (points) ▯ ▯ 1pt = 1/72 of 1in
- pc (picas) ▯ ▯ 1pc = 12pt

Example:

```
p {
font-size: 1.5pc;
line-height: 2pc;   }
<p> welcome to paragraph-1 </p>

<p> welcome to paragraph-2 </p>
```

OUTPUT:

welcome to paragraph-1

welcome to paragraph-2

NOTE: Here all the size unit are independent with any other html element or any other device except px (pixels) which is dependent on-screen size. 1px is the 1 dot (1 pixel) of the screen. If we use multiple devices then then the value of 1px will implies on all devices.

Relative Size

Relative Size is not a fixed length and always dependent on other html elements depending upon where it is used. Following are units which is used for relative size:

- em (font-size relative) ❓ ❓ 3em = 3 * font-size
- rem (Relative of the parent element font-size)
- Percentage % (parent-element relative)
- vmax (1% of viewports larger dimension)
- vmin (1% of viewports smaller dimension)
- vw (1% of width of display port)
- vh (1% of height of display port)

The most using units are em, rem, percentage. Rest of these rarely used.

Example-1

```
div {
  font-size: 30px;
  border: 1px solid black;
}
span {
  font-size: 0.5em;
}
<div> division with font size 30 px. <span> Spanning with 0.5em which is actually 0.5 * 30 = 15px </span>. </div>
```

OUTPUT:

*division with font size 30 px . Spanning with 0.5em which is actually 0.5 * 30 = 15px*

Example-2

```
div {
  font-size: 1rem;
  border: 1px solid black;        }
#division {
  font-size: 2rem;
  border: 1px solid red;  }
<div id="division">
Font size with 2rem which is actually the two times larger the browser text <div> division does not inherit from parent. It is 1time larger than browser text </div> </div>
```

OUTPUT:

Font size with 2rem which is actually the two times larger the browser text. *division does not inherit from parent. It is 1time larger than browser text*

Exercise : Explore remaining units by yourself and calculate the sizes of all the units.

Section 9.2: Using Appropriate Size

Although there is various type of units of size. First of we talk about absolute and relative sizes. As absolute size is fixed and independent and does not know about the size of neighboring elements. Moreover, if we do not about the screen size whether it is mobile or tablet or laptop and if we know about the device then how would we about the exact screen size of that device. Hence, for generic programming which is fit on all screens absolute size is not good but if we know about the exact screen size of device then absolute size are good. On the other hand, relative size are screen and other element dependent sizes. So, it is good for when we do not about exact screen size. Relative size implies generic programming. We can also use pixels if we don't know about device but yet we don't know about the other elements' sizes.

To sum up, using a unit always depend upon a type of project. One unit is good to use in one case and the same unit is bad in another situation. The recommendation is to use px from absolute and rm, rem, percentage from relative. These are units which are widely used but that doesn't mean that remaining units are useless. They are good to use in specific projects which suits their requirement.

Exercise

Q1: How many pixels is equal to 3em when the px of the page is set 16px?

Ans: 3*16=48pixels

48px is equal to 3em.

Q2: Write a Command to set the font size 1/72th of 1 inch.

Ans: font-size: 1pt;

As 1pt is equal to 1/72th of an inch.

Chapter 10

Types of Color in CSS

In this chapter, you will learn about:

Section 10.1: Introduction

Colors are very important to build the web page. Colors are used to style the html elements. CSS provide full control on colors. We can provide color value to html elements in various ways which are following:

1. By Predefined colors
2. By Hex colors
3. By RGB and RGBA colors
4. By HSL and HSLA Colors

Colors can be used to set border, margin, text, headings, background and foreground colors. We can use predefined on any html elements

Section 10.2: Predefined Colors

In CSS there are some predefined colors which can accessed by using the name of the color. For example, blue, violet etc. The Syntax is following:

* Tomato *

* Orange *

* DodgerBlue *

* MediumSeaGreen *

* Gray *

* SlateBlue *

* Violet *

* LightGray *

To set text color we use following syntax:

* text color is Tomato *

To set border value:

* Border color is Tomato </h1>*

Run all these codes in browser to see the output colors.

Section 10.3: Hex Colors

We can give value to color attribute in hexadecimal numbers. The value consists of 6 hex-digits the first two digits defines the value of red color, next two digits defines the value of green color similarly last two digits defines the value blue color. Each color value ranges from 0 to f in one digit. 0 means minimum intensity of color and f maximum intensity. Following is syntax to write value in hexadecimal.

Color: #ff00ff;

See some examples:

- #ff0000
- #0000ff
- #00ff00
- #ee82ee

Try this code to see the output result:

```
<!DOCTYPE html>
<html>
<body>
<p style = "background-color: #000000;"> #000000 </p>
<p style = "background-color: #3c3c3c;"> #3c3c3c </p>
<p style = "background-color: #787878;"> #787878 </p>
<p style = "background-color: #b4b4b4;"> #b4b4b4 </p>
<p style = "background-color: #f0f0f0;"> #f0f0f0 </p>
<p style = "background-color: #ffffff;"> #ffffff </p>
</body>
</html>
```

Section 10.4: RGB and RGBA colors

RGB

RGB stand for red green blue. It is combination of three colors i.e. red, green and blue. This notation takes three value from user and set the color according to that value. Each value defines the intensity of that color. The syntax to RGB color is following:

Color: rgb (first value, second value, third value)

Here the first, second and third are red, green and blue respectively. The range of each value is from 0 to 255. Every value set the intensity of respective color. Rgb (255,255,255) this means all the value has maximum intensity and result will show white color. Similarly, rgb (0,0,0) show the black color: See some following examples:

- Rgb (0,0,0)
- Rgb (60,60,60)
- Rgb (120,120,120)
- Rgb (240,240,240)
- Rgb (255,255,255)
- Rgb (255,0,0)
- Rgb (0,255,0)
- Rgb (0,0,255)
- Rgb (255,165,0)

RGBA

RGBA color is nothing but the extension in gba coloring. GBA color takes three values as input and rgba color takes 4 values 1 value additional to gba color which is called alpha value and all the remaining three values remain same like in GBA. The 4th value (alpha) defines the opacity of the color. The value range of alpha color is from 0.0 (minimum opacity) to 1.0 (Maximum opacity). The first three values should be discrete and alpha could be discrete or both. See following examples

- Rgba (255,99,79,1)

- Rgba (255,99,79,0.5)

- Rgba (255,99,79,0)

Try the following code in browser to see output colors:

```
<!DOCTYPE html>
<html>
<body>
<p style = "background-color: rgb (255, 0, 0);"> rgb (255, 0, 0) </p>
<p style = "background-color: rgb (0, 0, 255);"> rgb (0, 0, 255) </p>
<p style = "background-color: rgb (60, 179, 113);"> rgb (60, 179, 113) </p>
<p style = "background-color: rgba (255, 99, 71, 0);"> rgba (255, 99, 71, 0) </p>
<p style = "background-color: rgba (255, 99, 71, 0.2);"> rgba (255, 99, 71, 0.2) </p>
<p style = "background-color: rgba (255, 99, 71, 0.4);"> rgba (255, 99, 71, 0.4) </p>
</body>
</html>
```

Section 10.5: HSL and HSLA Colors

HSL

HSL stands for hue, Saturation and lightness. In HSL notation we provide three values first values ranges from 0 to 360. i.e. 360 means red color, 120 means green color and 240 means the blue color. This first value called the degree of color. Second value is saturation which has domain in percentage 100 percent saturation means full intensity of specified color and 0 percent means shade of grey. Third value is Lightness which is also has a domain in percentage value. 100 percent means is white and 0 percent means black. The syntax is following:

Color: HSL (hue, saturation, lightness)

-
-
-
-

HSLA

Like rgba HSLA is also an extension of HSL color. The 4th attribute in this HSLA notation is Alpha the other three are same as in HSL notation. Alpha decide the opacity of the specified color. The syntax is:

Color: hsla (hue, saturation, lightness, alpha)

Alpha value ranges from 0 to 1.

-
-
-

Try following to see the output colors:

```
<!DOCTYPE html>
<html>
<body>
<p style = "background-color: hsl (0, 0%, 0%);"> hsl (0, 0%, 0%) </p>
<p style = "background-color: hsl (0, 0%, 24%);"> hsl (0, 0%, 24%) </p>
<p style="background-color: hsl (0, 100%, 50%);"> hsl (0, 100%, 50%) </p>
<p style="background-color: hsl (0, 80%, 50%);"> hsl (0, 80%, 50%) </p>
<p style="background-color: hsl (0, 100%, 50%);"> hsl (0, 100%, 50%) </p>
```

* hsl (0, 100%, 75%) *

* hsla (9, 100%, 64%, 0.4) *

* hsla (9, 100%, 64%, 0.6) *

</body>

</html>

Exercise

Q1: Write a command to set the color of an html element to black with full intensity by RGBA color method.

Ans: color: RGBA (0,0,0,1);

Q2: Set the color of an element white by using hex and HSL colors.

Ans: Color: #ffffff;

Color: HSL (360,360,360)

Chapter 11

Fonts in CSS

In this chapter, you will learn about:

- 11.1: Font-family
- 11.2: Font-style
- 11.3: Font-size
- 11.4: Font-Google
- 11.5: Embedded/External fonts

Section 11.1: Font-family

While designing a web page font play a key role to make the page attractive. Font face is property that allow us to add many fonts in text. Browser pick the first font and apply it on the text and if this is not compatible with the browser or IDE then it will pick next font from that specified list and so on. So, we can say font family makes our code generic. if we want to give multiple fonts in font family then we separate them by using comma (,). The syntax is following:

font-family: times New roman, Arial

Browser will give high priority to times new roman and then arial font. Example is following:

```
<head> <style>
. paragraph1 {
  font-family: "Times New Roman", Times, serif;
}
. paragraph3 {
  font-family: "Lucida Console", Courier, monospace;
}
. paragraph2 {
  font-family: Arial, Helvetica, sans-serif;
}
</style> </head>
<p class= " paragraph1" > Times New Roman Font Paragraph </p>
<p class = "paragraph2" > Arial font Paragraph </p>
<p class = " paragraph3" > The Lucida Console font Paragraph </p>
</body>
```

OUTPUT

the Times New Roman Font paragraph

The Arial Font Paragraph

The Lucida Console Font Paragraph.

Section 11.2: Font-style

We have learnt that how can make text bold italic using HTML tags but here we make text italic without using html tag but with the help of CSS attribute font-style. Font-style is a property that define the text would be italic or not. The syntax is following:

Font-style: italic

Example:

```
<head> <style>
```

```
p. paragraph2 {
  font-style: italic;
}
p. paragraph1 {
  font-style: normal;
}
</style> </head>
<body>
<p class = "paragraph1" > paragraph in normal style. </p>
<p class = " paragraph2" > paragraph in italic style. </p>
</body>
```

OUTPUT:

paragraph in normal style.

paragraph in italic style.

Section 11.3: Font-Size

We set the text size in page by using CSS attribute Font size. We have learnt absolute and relative size earlier. We can use any size with font size attribute. The syntax is:

Font-size: 10px

Example:

```
<head> <style>
h2 {
  font-size: 30px;
}
p {
  font-size: 14px;
}
</style>
<h2>heading h2 with font size 30px </h2>
<p> paragraph with font size 14px </p>
</body>
```

Output:

heading 2 with font size 30px

paragraph with a font size 14px

Following is the example to set font size with relative size

<head> <style>

h2 {

font-size: 1.875em;

}

p {

font-size: 0.875em;

}

</style> </head>

<body>

<h2> heading h2 with font size 1.87em </h2>

* paragraph with font size 0.87em *

</body>

OUTPUT:

heading 2 with font size 1.87em

paragraph with font size 0.87em

Section 11.4: Font-google

HTML provide limited fonts to make our page user attractive. Google provide us 100s of fonts which we can use in css fonts by using google font API. If the standard font in html font does not exist then we go to google fonts.

To add google font in our css code we just add style sheet link and the reference to the font which we would to use i.e.

<link rel = "stylesheet" href = https://fonts.googleapis.com/CSS? Family = Sofia >

Example

<!DOCTYPE html>

<html> <head>

<link rel = "stylesheet" href = https://fonts.googleapis.com/CSS? Family=Sofia > <style>

body {

font-family: "Sofia";

font-size: 12px;

}

</style> </head> <body>

<h1> heading in Sofia Font </h1>

* paragraph in Sofia font *

</body> </html>

OUTPUT:

Heading in Sofia Font

Paragraph in Sofia Font

NOTE: There is long list of google fonts which you can easily find from officially google chrome fonts Some examples of fonts are ABeezee, Abel, Acme and Actor etc.

Section 11.5: Embedded/External fonts

If we don't find our desired fonts in html and google font then CSS give us another option which is called external or embedded fonts. By using this method, we can use our downloaded fonts in web pages. The CSS attribute which is used for this purpose is **@font-face.** Do following steps before adding your fonts in CSS.

1. Go to browser and download specific fonts.
2. Unzip the download file if it is zipped.
3. Go to the downloaded folder and find a file with extension .ttf (Extension of font files)
4. First install this file on your computer and then copy paste that file where the CSS file is placed.

we set the file in the destination now the task is to how to add that font-file in our code? For this we have to define the name of font family (you can use any name) by using CSS attribute **font-family** followed by specify the name of and file location of that font family by using CSS attribute **src** . These two attributes must be within the definition **@font-face** The Syntax is following:

```
@font-face {
  font-family: fontExample;
  src: url(fontName.ttf);
}
```

Note: In the above case our font file is placed in the same directory of CSS file. But if the font file directory is different from CSS file directory then we have to mention the complete directory of font file.

Example:

```
<style>
@font-face {
  font-family: FontExample;
  src: url (sansation_light. woff);
}
@font-face {
font-family: FontExample;
src: url(sansation_bold.woff);
font-weight: bold;
}
div {
  font-family: FontExample;
}
</style>
<h1> heading h1 with font face </h1>
<div> Div instruction </div>
```

OUTPUT:

Heading h1 with font-face

Div instruction.

NOTE: There a many file format like ttf, otf, woff, woff2 and many older formats exists. It is strongly recommended to use ttf or woof.

Chapter Summary

In this chapter we have discussed CSS attributes related to fonts to make our web page looks better.

- Font-family is used to give a list of fonts to browser
- Font size is used to specify the size of font
- Font style is used to specify the text is italic or normal
- By using google fonts we can use our desired fonts.
- By using font-face method we can use external/embedded fonts.

Besides these there are many CSS attributes which is also used like **font-weight** which is used to highlight the text Bold or not. **Font-variant** used to specify the that text is in small caps or not.

Exercise

Q1: Write a simple code with three html elements and set the font family of the whole page to "Courier New" and for h1 set the font "Times New Roman".

Ans:
```
<style> body {
 font-family: "Courier New";
}
h1 {
 font-family: Times new Roman;
}
</style> <body>
<h1>This is a Heading</h1>
<p>paragraph1</p>
<p> paragraph2</p>
</body>
```

Q2: Write a CSS Block code to set the "p" element bold and "10px" in Times new roman.

Ans:
```
p      {
 font: bold 10px times new roman;   }
```

Chapter 12

Inline & Block elements, Multimedia

In this chapter, you will learn about:

12.1: inline vs block elements

12.2: Display features

12.3: Integrate images

12.4: Integrate audio/videos

12.4: Page structuring in HTML

Section 12.1: Inline and Block Elements

Depending upon the type of element there is default display value of each html elements. If we want to use custom display value then there are two main types displaying elements:

1. Inline Display
2. Block display

Inline Display

Inline display remains within in a line and does start from new line and does not take unnecessary width. There are various html tags which is inline. For example, span:

<body>

 This is Hello Inline a paragraph containing span inline element

</body>

OUTPUT:

This is Hello Inline a paragraph containing span inline element.

Span is CSS attribute and used to style some part of content. By using span element, we can style the many parts of content like:

* I was born in London I took admission in Oxford when I was 19 *

OUTPUT:

I was born in **England** I took admission in **Oxford** when I was 19.

Following is the list of some HTML tags which have inline display feature as a default:

<a>	
<abbr>	<image>
<button>	<b>
<code>	<q>
<select>	<label>
<time>	

Block Display

Irrespective of inline elements, Block elements always starts from new line and also take extra width which is unnecessary for that element. In simple block element take whole block. The div tag specify a block in html file. See following example:

<!DOCTYPE html>
<html> <body>
* div block *
* This is paragraph after div block *
</body>
</html>

OUTPUT

div block

This is paragraph after div block.

Like Span, Div is also used for styling a multiple section in html file.

<!DOCTYPE html>
<html> <body>

* *
<h2> Heading h2</h2>
* This is paragraph 1 *
* This is paragraph 2 *
* </body> </html>*

OUTPUT:

Section 12.2: Display Features

We have discussed two displays in above section but there are many features of display:

- Inline
- Block
- Inline-block
- None
- Content
- Flex
- Grid

And so on.

The display behavior of html elements styling by using display attribute. The syntax is:

Display: value

Example

<style>

p {color: red;}

p. inline {display: none;}

p. block {display: inline;}

p. inline_block {display: block;}

p.ex4 {display: inline-block;}

</style>

This is None Display feature.

This is Inline Display feature.

This is Block display feature.

This is Inline-Block .

OUTPUT:

This is None Display feature.

This is Inline Display feature.

This is

Block display

feature.

This is inline-block feature.

NOTE: There are list of value of display for example flex for displaying block by block, grid for inline-grid text table for put text in table and so on.

Section 12.3: Integrate Images

We have discussed to handle text in web pages, now we are going to handle images in web pages. Images increase the captivity of the page.

Image syntax

Html provide us a tag **<img>** to embed image in web page. We did not insert images in web page but we connect it to the other web pages of pictures.

The <img> tag is an empty tag and has two attributes **src** and **alt.** The syntax is following:

<img src = "url" alt = " alternate text ">

src attribute defines the location (URL) of the image.

When we open a web page. It loads images from other sites or from database and insert it on to page. In case if image not found then it will show an empty block. For this we use **alt** attribute.

The **alt** Attribute is used to provide an alternate text if the image is not found at specified position. For example:

<img src = "pakistanFlower.jpg" alt = "Pakistan Flowers">

In case the image *pakistanFlowers.jpg* not loaded then the text *Pakistan Flowers* would be shown instead of image.

Width & Height of Image

The width and height of image is set by using simple width and height attributes or by using style attribute. For example:

<img src = "iamge.jpg" alt = "AlternateText" style = "width:50px; height:60px;">

OR

<img src="image.jpg" alt="AlternateText" width="50" height="60">

Images in Other Directories

Sometime the requires image is not in the folder of HTML file. In that case we have to provide exact location of the image for example:

<img src = "/images/image.jpg" alt = "Image Not Found" style = " width :225px; height: 223px;">

Animated Images

HTML also permit us to use animated images in the page. The syntax is same for adding animated image:

<img src = "code.gif" alt = "CSS Code" style`= "width: 50px; height: 21px;">

Image Floating

We can also use floating image in web pages by using float attribute. The syntax is following:

<p><img src = "image.gif" alt = "Happy man" style = "float: right; width:40px; height:39px;">

This image.gif is floating image to the right of text.

<p><img src="image.gif" alt="Happy man" style="float: left; width:42px; height:42px;">

This image.gif is floating image to left of text.

NOTE: We can also use link of the image inside <a> and href attribute. Also, we can use images as background image to the content by using **background-image** attribute.

Section 12.4: Integrate Audio/videos

Audio

For adding audios in web page, HTML provide audio tag. The syntax is following

<audio>

<! --Attach audio -- >

</audio>

The audio tag act as a container and contain a lot of audios. It plays the first audio which supports the browser. For example:

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

In type attribute, we give the type of audio file.

Video

Same with the audio tag, HTML provide us also video tag with a little bit difference in definition. We have to set the width and height of the video. The remaining rules are same with the audio tag. It picks up the video which support the browser and play it:

```
<video width="300" height="200" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
</video>
```

Section 12.5: Page Structuring in HTML

Now we are able to structure the page by using HTML. First of all, we should know that which basic things a page need to look attractive.

- Heading section at the top with a tittle or logo
- Navigation Bar
- Body Content Consists of images, text etc
- Footer of the page

Heading Section can be built by using header element div element. We can write tittle by using tittle attribute. To insert logo on to page we can use img tag. Then comes navigation bar. We insert navigation bar by using nav tag. Then we can insert as many contents as we want and style them by using external or internal stylesheet. Then Comes footer of the page which can also be built by using footer element.

Exercise

Q1: Write a CSS block to set display inline of paragraph p

Ans:
```
p. inline {
    Display: inline;
    }
```

Q2: Write a HTML code to add audio with alternate text “Not Found”

Ans: *<audio controls>*
<source src="horse.ogg" alt=" Not found" type="audio/ogg">
<source src="horse.mp3" alt=" Not found" type="audio/mpeg">
</audio>

Q3: Which attribute is used to set the block display?

Ans: Block display is set by using division method the help of **div** attribute.

Chapter 13

Positioning of elements

In this chapter, you will learn about

Section 13.1: Introduction

In order to place the html element in correct position, CSS provide **Position** property. Position property is used to set the position of elements. Position of an element can be:

1. Absolute
2. Relative
3. Fixed
4. Static
5. Sticky

All these positions are set by using position method which we will discuss in coming sections.

Section 13.2: Absolute & Relative of elements

Relative Position

Every element has its normal position in default. When we use relative position then that specific element set itself relative to the normal position. To set element's position we set its left, right, up bottom range. In this range no other content would be adjusted. The syntax is:

Position: relative;

Example:

```
<style>
div. Relative {
  position: relative;
  left: 20px;
  border: 2px solid Black;
} </style>
<body>
<p> Paragraph before relative position </p>
<div class = "Position_Relative" >
This element has relative position
</div> </body>
```

OUTPUT:

Paragraph before relative position

This element has relative position

Absolute Position

An absolute position of an element is a position of relative to its parent or ancestor position. If there is no any ancestor or parent exist. Then the absolute position would be relative to the document content. The syntax to set position absolute is:

Position: absolute;

Example:

```
<style>
Div. Position_Relative {
  position: relative;
  height: 100px;
   width: 300px;
   border: 1px solid black;
}
div.position_absolute {
```

```
  position: absolute;
  top: 70px;
  right: 1;
   height: 100px;
    width: 150px;
    border: 1px solid blue;
}
</style> <body>
<p> Paragraph before relative and absolute position </p>
<div class = "Position_Relative">This div element has relative position;
  <div class = "position absolute">This div element has absolute position </div>
</div> </body>
```

OUTPUT:

Paragraph before relative absolute position

Section 13.3: Fixed, Sticky & static position of elements

Static Position

Static position is the default position of html elements. It is independent of top, left, right and bottom properties. Static position of an element would be the normal position according to the flow of the content of the page. The syntax is:

Position: static;

Example:

```
<style>
```

```
div. position_static {
  position: static;
  border: 1px solid red;
}
</style> <body>
<p> Following element has static position </p>
<div class = "Position_static">
  This element has static position; </div>
</body>
```

OUTPUT:

Following element has static position

NOTE: An html element whose position is anything except static position is called positioned element.

Sticky Position

Sticky stans for stuck anything. Sticky position of element is used to stuck the elements on the screen while scrolling. Normally, when there is a lot of content on the page then the browser adds scrolling view itself. In this if we want to see any element at very time during scrolling then we set the position of that element sticky. Sticky element can be stuck on the screen any where by using top and bottom properties. The syntax is following:

position: -webkit-sticky;
position: sticky;
top: 0;

In the above syntax position: sticky defines that the position would be stuck of the element on the top of the page (top: 0).

Example:

<style>

```
div. Position_sticky {
  position: -webkit-sticky;
  position: sticky;
  top: 1;
  padding: 2px;
  background-color: white;
  border: 1px solid red;
}
</style> <body>
```

* This is sticky element at top = 1 *

* </body>*

OUTPUT:

NOTE: Here we cannot add scrolling view. To view the result, try above code in browser. You will observe that above result will be stuck at top of page when we will do scroll by adding more content.

Fixed Position

We have discussed sticky position which is actually an extension of fixed position. If there is a lot of content on the page then during scrolling the content hide from user but if we want some element to be on screen always on its specified position then we use fixed position method. Except this fixed position element all the remaining elements are scrollable. The syntax is following:

Position: fixed

Beside this we can also give the width and height of element to set the position at required place.

Example:

<style>

div. position_fixed {

  position: fixed;

  bottom: 0;

  right: 1;

  width: 200px;

  border: 1px solid red;

}

</style> <body>

 Paragraph before fixed element

This div element has position: fixed;

 </body>

OUTPUT:

Paragraph before fixed element

NOTE: In above the fixed element would be shown at the bottom right corner of the page. Try this code on the browser and check its position by scrolling the content.

It is recommended that add fixed position element at the right corner because fixed elements do leave a space where it is placed as a default.

Section 13.4: Overlapping Elements

When we add multiple elements by using positioned property then there would be high probability that these multiple elements would be overlapped. To overcome on this problem, we use **z-index** attribute. Z-index is a stack property which decide which element should in front and which element at the back. The user will see both elements but one at the front and one at the back.

Example:

```
<style>
img {
  position: absolute;
  left: 0px; top: 0px; z-index: -1;}
</style> <body>
<p> First paragraph </p>
<img src="picture.png" width="200" height="240">
<p> Second paragraph </p>
</body>
```

RESULT:

As image has z-index -1 so image would be at the back of the two paragraphs. Try this code in the browser by adding any picture in **src** attribute.

NOTE: The element which has large z-index value would be at the top and vice versa.

Chapter Summary

- Absolute/ Relative position set the elements positions in the page absolute/relative.
- Fixed position is used to set the position of element fixed
- Sticky position used to stick the position of element at the top during scrolling
- Z-index is used to overcome the problem when two elements are overlapping

NOTE: By using position method discuss in the chapter we set text on the picture at any position of picture.

Exercise

Q1: write a div block code to set the position of an html fixed at bottom right corner.

Ans:
```
div. Fixed {
  position: fixed;
  bottom: 0;
  right:1;      }
```

Q2: Write a code to set the position of an element static with yellow 2px border.

Ans: div. Static {

position: static;

border: 2px yellow; }

Chapter 14

Links in HTML

In this chapter, you will learn about:

14.1: Introduction to hyperlinks

14.2: Styling Hyperlinks

14.3: Link Bookmarks

Section 14.1: Introduction to hyperlinks

HTML links are called hyperlinks. Hyperlinks connect the two pages or two parts of the document. HTML links can also use on a HTML button. When the users want to go on the other pages or any other parts of the document they simply click on that link or button. Browser will take user to that specific link by using the onclick () method. One important recognition of link that cursor move into little hand when it is moved on the click. A link can be image, HTML element, Button but it is not a text.

The html tag use for this purpose is **<a>** we write the hyperlink inside <a> tag by using CSS attribute **href** . The syntax is following:

*** text of link ***

We write the link of the page where we want to go after user click and this link is placed behind the user. User will only the text of link. User click on the text and browser will consider it as link. If text of link not exist then user can see complete link of the page.

Example:

<body>

* This is a link *

</body>

OUTPUT:

This is link

Every link has a default setting of:

- The link which is not visited will be appeared as blue in color and underlined
- The link which is active will be red and underlined.
- The link which is visited will be purple and underlined.

HTML Target Attribute

Hyperlink take the user to the desired page. As a default that page will be in same tab and same window. If we want to change this setting then HTML provide us target attribute which the target tab or window to be where link would be open. The target element has the following values:

1. **_self:** This is same with default setting of the link. The link will be open in the current window and current tab.
2. **_blank:** This _blank opens the link in new tab or new window
3. **_parent:** This will open the link in parent view.
4. **_top:** This will open the link in full window or in full view.

The syntax is following:

*** This is a link ***

*** This is a link ***

*** This is a link ***

*** This is a link ***

Absolute & Relative Links

Absolute link is a full address of website which is not in the domain of current website.

Relative link is for that page which is a part of website and instead of giving a complete url we only mention the desired page name: For example:

*** Images ***

Image as LINK

The image can be used as a link by using following syntax:

******
<img src = "iamge.gif" alt = "image not loaded" style = "width:42px;">
******

Email Address as LINK

We can use email as link. This link will be open in composing email page or in th email application. The syntax is:

*** Compose email ***

Button as Link

To use button as link we apply onclick method inside html <**button>** tag. The syntax is following:

<button onclick = "document. location ='dimages.asp'"> images </button>

Tittle of link

The tittle of link would be shown when we move the cursor on the link. To add tittle, we use tittle attribute. The syntax is:

*** learn html ***

Section 14.2 : Styling hyperlink

Hyperlinks can be styled by using any CSS styling property. But CSS provide more features to style the links depending upon the current state of the link. There are four state of link which is:

- **a: link:** The link which is not visited
- **a: visited:** The visited link
- **a: hover:** The active link (when the user cursor is on the link)
- **a: active:** the active link (when it is clicked by user)

text-decoration attribute is used to decorate the text of the link. Text decoration defines that the color would be underline or not. Background-color attribute used to set the background color of link. Color attribute is used to set the color of link. Try following code in browser:

```
<style>
a: link {
    color: red;
   text-decoration: none;
    background-color: yellow;
}
a: visited {
  color: green;
  text-decoration: none;
  background-color: cyan;}
a: hover {
  color: hotpink;
  text-decoration: underline;
  background-color: lightgreen;
}
a: active {
  color: blue;
  text-decoration: underline;
  background-color: hotpink;
}
</style> <body>
<p> <b> <a href = "default.asp" target = "_blank"> Example of link styling </a> </b> </p>
</body>
```

OUTPUT:

This is a link

NOTE: Try the above code in the browser then your concept will be clear.

Similarly, button links also styled by using the above method

Section 14.3: Link Bookmarks

Normally the content on the page in not very small. We can scroll up down to see the content. But if the page contains very large content and we want to move quickly on the bottom element when the current position is on the top of the page then it would long time to scroll. To overcome this problem, we use link markup method to move quickly to other parts of page.

There are two steps to create markup link:

1. First make create a book by using unique id
2. Add that unique to the href attribute

The syntax is:

<h2 id = "ch4"> this is ch4 </h2>

*** cursor move to ch4 ***

In the first line, we have created the bookmark with unique id ch4 and then use that id as a link. When cursor click on the link browser will show the ch4 element, no matter what the position of **ch4** is

Exercise

Q1: Write a code to define a link with the text “visit another page”

Ans: * Visit another page *

Q2: write a code to define another section of the page and then go to the that by using hyperlinks.

Ans: * This is Another section *

* You are on another section *

Chapter 15

HTML Forms

In this chapter, you will learn about:

Section 15.1: HTML Forms

HTML forms are used to take the input from user. After taking this information sent to the server. HTML provide <form> tag to create html form. The syntax is:

<form>

Other Elements

</form>

Section 15.2: Input Types

Some of the Form elements are followings:

- **Input:** This is most using form element and used to take the input in many ways depending upon the condition
- **Type:** This attribute decides the input type. The syntax is:

```
<form>
  <label for="Input_Text"> your input:</label><br>
 <input type="text" id="input_Text1" name ="Input_Text">
</form>
```

Type text is used to take the text input from the user. Label define a label of elements and **id** of input element must be same with the value of **for** attribute. Following are some important types of input:

- <Input type =" botton">
- <Input type=" Checkbox">
- <Input type=" color">
- <Input type=" date">
- <Input type=" datetime-local">
- v<Input type=" email">
- <Input type=" file">
- <Input type=" hidden">
- <Input type=" image">
- <Input type=" month">
- <Input type=" number">

- <Input type=” password”>
- <Input type=” radio”>
- <Input type=” range”>
- <Input type=” reset”>
- <Input type=” search”>
- <Input type=” submit”>
- <Input type=”tel”>
- <Input type=”text”>
- <Input type=”url”>
- <Input type=” time”>
- <Input type=” week”>

NOTE: Default value of input is text.

The **submit** attribute is used to submit the information to the use. **Reset** is used to reset the value to initial values. **Radio** is used to mention multiple choices to the use and take one input. While on the other hand **checkbox** is used take multiple input from the user. **Button** define the buttons on the page and **color** used to take the color from the user. **Range** is used to apply the restrictions to the user on the input. **Tel** is used to take the telephone number; **email** is for getting email and **password** is to get the password from the user.

Section 15.3: HTML Forms attribute

HTML attribute are written inside the form tag in the first line:

There are following HTML forms attribute:

- **Action:** Action attribute define which action is to be performed when user submit a form.

  <form action="/page.php"> </form>
- **Target:** Target attribute specify the target page/window where the action is to be performed.

  <form action="/page.php" target="_parent"> </form>
- **Method:** The method attribute is used to set the HTTP method (get/post)

  <form action="/page.php" method="post"> </form>
- **Autocomplete:** Boolean value (on/off) means the form would be autocompleted or not when to submit a form.

  <form action="/page.php" autocomplete="on"> </form>
- **Novalidate:** Specify the data of form is validate or not.

  <form action="/page.php" novalidate > </form>

Section 15.4: Form Elements

Following elements are used in form tag

<input> *<label>*

<select> *<textarea>*

<button> *<fieldset>*

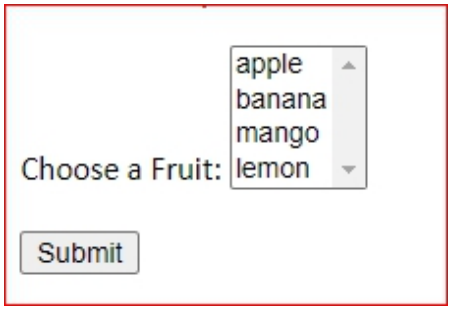
<lagend> *<datalist>*

<output> *<option>*

<optgroup>

Select is used to make a dropdown list and **option** give the choices to the user to select option and **value** is used to define the values that can be selected. **Size** specify how many option users can see at a time in dropdown list (scrolling). **Multiple** allow user to select multiple choices.

```
<label for="cars">Choose a fruit:</label>
<select id="cars" name="fruits" size="4" multiple>
  <option value ="Apple"> Apple </option>
  <option value = "Banana"> Banana </option>
  <option value = "Mango"> Banana </option>
  <option value = "Lemon"> Lemon </option>
</select>
```



Section 15.5: input Attributes

There are following Input attributes:

- **Value:** is used to set the initial value.
- **Readonly:** is used to set the element only for reading (writing restriction)

  <input type="text" id="fname" name="fname" value="John" readonly>
- **Disable:** is used to disable the any input element.
- **Size:** Specify how much width is visible to the user in characters.
- **Maxlength:** used to apply restriction how many characters user can input at a time.
- **Required:** defines that this input must be filled. Without this form cannot be submitted.
- **Datalist:** is same as dropdown list. Used to define list of data.

Exercise

Q1: write a form block code to add button input field and set the value “Hello”.

Ans: *<form>*

<input type=” button” value=” hello”>

</from>

Q2: Make an empty drop-down list with name roll number.

Ans: <form>

<select name=" roll numbers"> </select>

</form>

Q3: Make a button with value "Hello world"

Ans: <form>

<button> Hello world <button>

</form>

Chapter 16

Background and Page Layout

In this chapter you will learn about:

16.1: Set Background with CSS

16.2: Set page layout

Exercise

Section 16.1: Set Background with CSS

CSS provide full control to style the background. We can style the background of an element with color, image or any other attachment.

- **Background-color:** is used to set the background color of an element

  background-color: lightblue;

  Also, we can give values in GBA, HSL or in hex value to set the color

- **Background-image:** is used to set the background image of an html element

background-image: url("paper.gif");

- **Background-repeat:** used to repeat the image horizontal image in x-times/y-times

background-image: url("gradient_bg.png");
background-repeat: repeat-x;

- **Background-attachment:** Used to set the background position fixed or scrollable. Before we set the position of background style by using **background-position**

```
body {
  background-image: url("img_tree.png");
  background-position: right top;
  background-attachment: fixed;
}
```

Section 16.2: Set page layout

A website consists of following basic section

- Header section
- Menu Section
- Content
- Foote

C

Header & Footer

Header & footer layout can be styled using *.header* and. footer method. For example

```
. header {
  background-color: #F1F1F1;
  text-align: center;
  padding: 20px;
}
```

```
. footer {
  background-color: #F1F1F1;
  text-align: center;
  padding: 10px;
}
```

Navigation Bar

Navigation bar can be styled by topnav

```
. topnav {
  overflow : hidden ;
  background-color : #333 ;
}
.topnav a {
  float : left ;
  display : block ;
  color : #f2f2f2 ;          }
```

“Topnav a” is used to set the link style

Content

A website page can hold 1 column content, 2 column or 3 column content. This content can be styled by using .content method:

```
. column {
  float: left;
  width: 33.33%;
}
.row: after {
  content: "";
  display: table;
  clear: both;
}
```

. row: after clear the area after that column

Exercise

Q1: Write a CSS code to content column-1 width to 25 percent, column 2 width should be 50 percent.

Ans: . column. Side {

Width: 25%

}

. column. Middle {

Width: 50% }

Q2: Write a CSS code to fix the background image at right bottom corner.

Ans: *body {*
background-image: url("iamge.png");
background-position: right bottom;
background-attachment: fixed;
}

Chapter 17

Project Setup

In this chapter, you will learn about:

17.1: Responsive Web design

17.2: Link HTML & CSS Files

17.2: Folder Structure

17.3: Create a front-end website

Section 17.1: Responsive Web design

The responsive web page is that which good look in different screen size such as on computers, tablets and on smart phone. We have to make such a HTML and CSS code which can make website auto design to different scales. To make responsive website we have to add following line in the code:

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

After using above line if we add an image with **width 100 percent** then it would set itself according to the screen size but if the screen size is large and picture size is less then there is chance that our picture can be distorted So, it is recommended to use **max-width.** It would set the image width to original size if the screen size is large and make the image small size on tablets and on computers.

*<img src="img_girl.jpg" style=" **max-width:100%;** height: auto;">*

The text on responsive web design is set to relative to viewport which we have set in the first line:

*<h1 style=" **font-size:10vw** ">Hello World</h1>*

Section 17.2: Link HTML & CSS Files

We have studied that how we can style the html elements by using external style sheet. There can be multiple stylesheet you can make. After making External Stylesheets we have to link it with html file. Suppose we have 2 external stylesheets with the name paragraph.css and heading.css and one html file name as document.html, we link these files by following syntax:

<link rel = "stylesheet" href = "paragraph.css">

<link rel = "stylesheet" href = "heading.css">

These two lines must be within the **<head>** tag. The above syntax will be correct if we have all the files in same directory but if the files has different directories then we have to mention the complete physical address of folder.

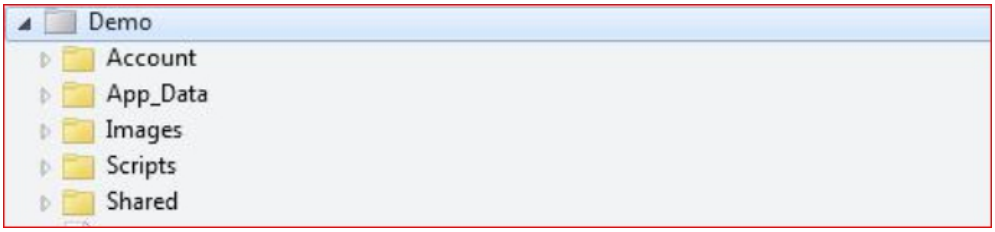
Section 17.2: Folder Structure

In a website context, there are two types of folder structures:

1. Logical Folder structure
2. Physical/virtual folder structure.

Logical folder Structure

Below is the example of logical folder structure of complete website:



- Security files are stored in Account folder.
- We have studied only the front-end of website. In case we are making the full stack web then there would be a lot data which is stored in database. App_Data contains that database
- Images which are using in the web are stored in Images folder
- Html, JavaScript and all the front-end codes are stored in script folder.
- Stylesheets are stored in Shared folder.

Physical/virtual folder structure

There would be a slight difference between physical and virtual names of folders. Physical name are the names which is use to access the folder on the computer/server. It is actually the address of the folder in hard drive of computer. While the virtual names are those which is use to access the folder from the website.

From the above example:

Suppose, there is an image named as flower.png. Then website access this picture using following virtual address:

"Images/pic31.jpg".

But the physical address of picture flower.png is:

"C:\Johnny\Documents\MyWebSites\Demo\Images\pic31.jpg"

Section 17.3: Create a front-end website

Congratulations! You are now capable to build a complete website by using the knowledge you have learnt in this book. We have explained that which IDE should we prefer to write a script code. Surely, you have a lot of practice of using Atom IDE. So, write html and css script, link them and compile these files in browser. Besides this you can use many frameworks for example bootstrap framework is the best framework of html.

After making a complete website you can run it only on your computer/local PC which contain the Coding files. To publish your site on google you have to buy domain and Hosting. There are many sites which provide hosting and domain. You have to upload your code in the hosting site. Now you can access your website with the link only. Following are some famous web store which are providing hosting and domain:

- Godady.com
- Bluehost.com
- Hostinger.com
- Siteground.com

You are now a front-end developer. You can do freelancing in by building a front-end website. Beside this you can build your own blogging website and post your own content day by day.

Exercise

Q1: Write a code to link three CSS file to HTML file named as CSS1.css, CSS2.css and CSS3.css

Ans: <link rel = "stylesheet" href = "CSS1.css">
<link rel = "stylesheet" href = "CSS2.css">
<link rel = "stylesheet" href = "CSS3.css">

Q2: Set the font size to 5 in responsive web page with the text “Kinara”.

Ans: *<h1 style=" **font-size:5vw** "> Kinara </h1>*

Extensive projects

Project-01: Building a Tribute page

As a beginner, you can make simple tribute page in which you can admire someone by using basic knowledge of HTML and CSS. For this you can use image, list, paragraph, headings, and links with CSS to style it in decent way. The template of this project is following:

Aristotle Onassis

1906 - 1975

One of the richest people in the 20th century

Aristotle Socrates Onassis (Greek: Αριστοτέλης Ωνάσης, Aristotelis Onasis; 20 January 1906 – 15 March 1975),[1] commonly called Ari or Aristo Onassis, was a Greek-Argentine shipping magnate, who amassed the world's largest privately owned shipping fleet and was one of the world's richest and most famous men.

The most well-known portrait of Onassis.

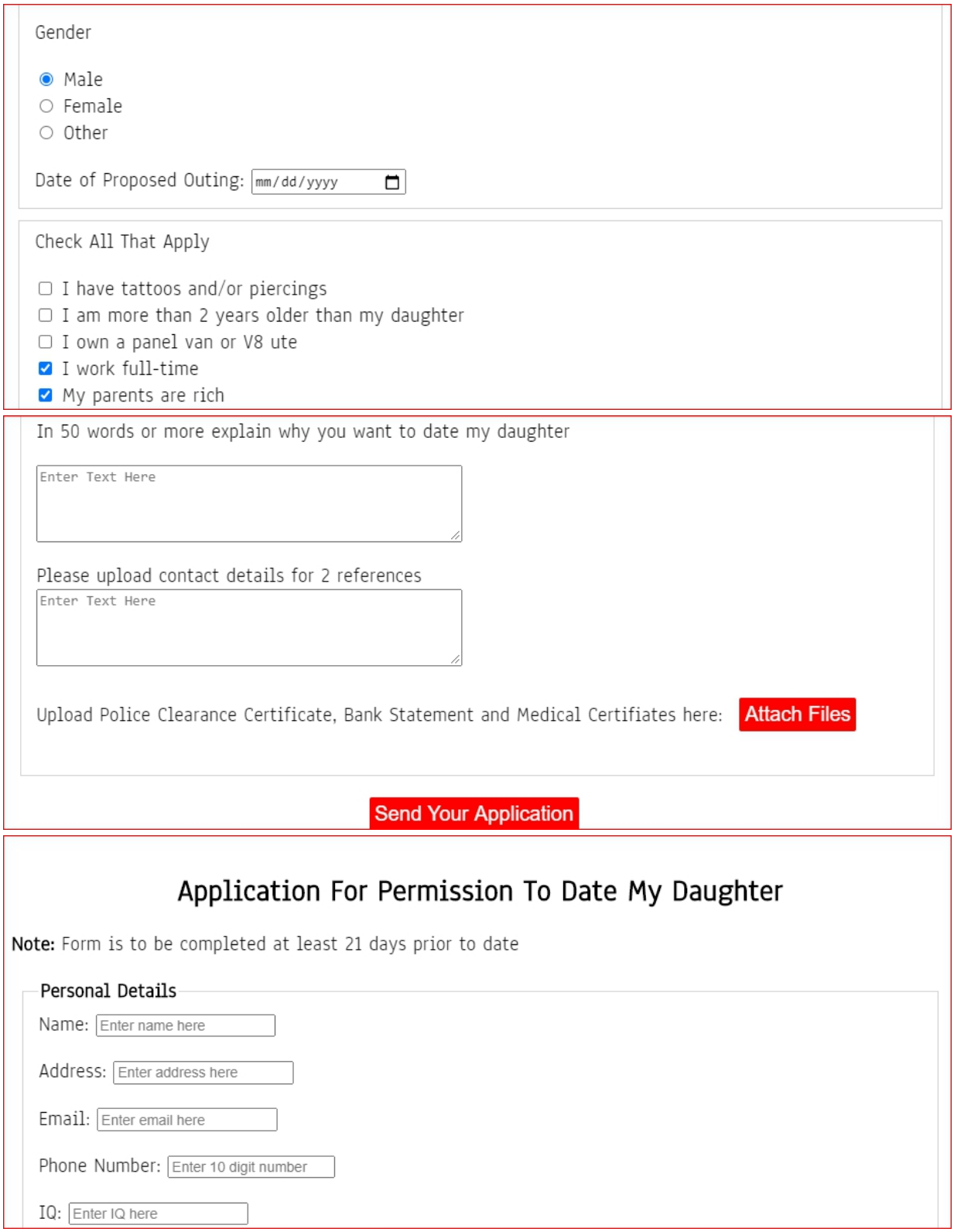
The following list is a time line of Onassis Life

- **1906** - born in Karataş, Smyrna, Ottoman Empire.
- **1922** - He left from Smyrna during the great fire of Smyrna in 1922.
- **1932** - He went to Buenos Aires, Argentina and start working as a telephone operator.
- **1929** - He relocated to New York and started his shipping businesses.
- **1946** - Onassis married Athina Livanos, daughter of shipping magnate Stavros G. Livanos and Arietta Zafrikakis, on 29 December 1946. The couple had become largely separated by the mid-1950s.
- **1950**-1956 - Onassis had success whaling off the Peruvian coast.
- **1953** - In Monaco he started to purchase the shares of Monaco's SMB using the tax haven of Panama and finally take the control of SBM.
- **1954** - Cancelation of the agreement between the Saudi government and Aristotle Onassis to transport Saudi oil on his tankers and "in any case, to make the agreement ineffective".
- **1956** - Aristotle Onassis signed a contract granting him the operational rights to the Greek air transport industry. Olympic Airways was founded.
- **1957** - Onassis and opera prima donna Maria Callas embarked on an affair despite the fact that they were both married.
- **1966** - Rainer of Monaco approved a plan to create 600.000 new shares in SBM reducing Onassis's stake to under a third.
- **1967** - Onassis court until he left Monaco.
- **1968** - Onassis announced the launch of Project Omega, a $400 million investment program that aimed to build considerable industrial infrastructure in Greece including an oil refinery and aluminum smelter.
- **1968** - Onassis ended his relationship with Callas to marry Jacqueline Kennedy, widow of U.S. President John F. Kennedy.
- **1975** - Onassis died at age 69 on 15 March 1975 at the American Hospital of Paris in Neuilly-sur-Seine, France, of respiratory failure, a complication of the myasthenia gravis from which he had been suffering during the last years of his life.

Read more about Aristotle Onassis on Wikipedia.

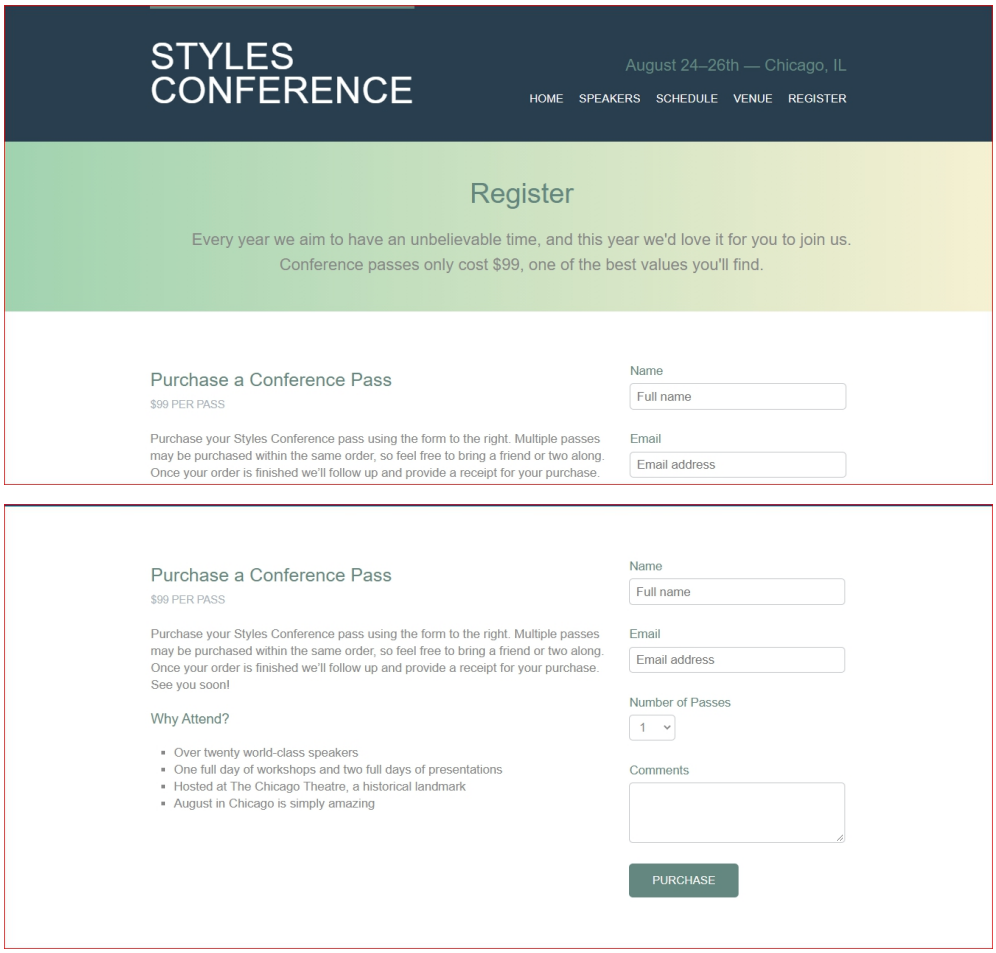
Project-02: Survey Form

Forms are extensive of any project. Once you understand the basic knowledge of Input/output, you can make any website page with using form.



Project-02: Survey Form

By using the basic knowledge given in this book you can make simple conference page in which you can invite people. Interested people purchase a pass by clicking on the register button. Following is the template page:



Besides this, you can make your blogging page for publishing blogs, or you can make personal portfolio and many other website designs.

Source Code for Download for Quick Testing

To download source code, you can use following line of code:

Download File

Following are some important tools for quick testing HTML & CSS code in which you can run various language code including html and CSS code and generate a private link of that code after compiling it. Then you can send this link to any of your friend to test and use your code.

1. JSbin
2. JSFiddle
3. Codepen
4. WebMaker
5. CSS Desk

Advantages of the Book

This book will help you in following way:

1. Simple and practical explanations:

This book is in simple language. The Person who know basic english can understand this book very well. We have explained every concept in simple way and then we give the practical and most related example to that concept which will help you to understand it. The examples and the concepts which we have explained are the most appropriate with respect to professional websites. This will be helpful when you are building front websites design like your own blogger websites, your own content containing website.

2. Exercises with solutions:

For your deeper understanding, we have given the exercises at the end of each chapter. Every chapter contain two or three question with basic concepts. You can test your understanding by thinking and practicing the solution of that question. The solution of this exercises is given. But it is strongly recommended that you first your solution and then compare your solution with the solution given in the book.

3. Extensive practical projects:

One more thing which is explained in this book and very helpful to groom your front-end skills is the extensive projects. You have learned the basic concepts and examples and then practice that concepts, but how can you build your own website which contain huge data and content. For this purpose, we have attached three extensive projects which will serve as template for your own website. Here, we have given only three projects which are Tribute page, Form page and event/conference page. By using the tribute template page, you can make any tribute to any other person or your friend. Form is the most using concept in every website page. In every website which you have built you have to take the feedback of the users or login and sign-up attributes or any other input. In every case you use form. The basic view of form is given in the second extensive project. The third project is the when you are inviting someone to your event or any other conference. You can use same template for your page.

4. Source code for download for quick testing:

At the end, we have mentioned some tools in which you can use for testing and generating link of your code. After writing large number of lines code it is difficult to send code to many friends or our teacher. So, the tools make this way easy. Another advantage of these tools we can test our code inside these tools instead of browser. These tools show the output when are writing the code.

Congratulations! You have learnt front-end design by using html & CSS if you have followed this book step by step. You can now earn by freelancing by building front end websites or you can blog by building your blogger site.

Best of luck

Glossary

HTML Elements

<a> <abbr> <acronym> <address>
<article <aside> <audio> <b>
<blockquote> <body>
 <button> <cite>
<dd> <del> <dfn>
 <dl> <dt>
 <embed> <fieldset>
<figcaption> <figure> <footer>

<form> <h > <h > <h > <h > <h > <h > <head> <header> <hgroup> <hr /> <html> <i> <iframe> <img> <input> <ins> <label> <legend> <link> <meta> <nav> <object> <option> <p> <q> <s> <script> <section> <select> <source >   <style> <sub> <sup> <table> <tbody> <td> <textarea> <tfoot> <th> <thead> <title> <tr> <u> <video>

HTML Attributes

Action align alt autoplay bgcolor border checked cite class codecs cols colspan content controls description for frameborder global attributes height href http-equiv id idattribute loo p maxlength method multiple name placeholder poster preload rel robots rows rowspan scrolling seamless selected size src style target title type type (audio/video) type (CSS) type = "date" type = "email" type = "hidden" type = "image" type = "password" type="search" type = "submit" type = "url" value width

CSS Methods

background-attachment property background-color

background-image property background-image property (gradients)

background-position property background property

background-repeat property border-bottom-color
border-bottom-style border-collapse property
border-color border-image property border-left-color border-left-style border property
border-radius property border-right-color border-right-style
border-spacing property border-style
border-top-color border-top-style
bottom property box-shadow property
clear property color cursor property display property
empty-cells property float property float property (images) @font-face
font-face font-family
font-size font-style property height (images) left property
letter-spacing line-height
list-style-image property list-style-position property
list-style property list-style-type property
margin-bottom property margin-left property
margin property margin-right property margin-top property opacity
padding-bottom property padding-left property
padding property padding property (tables) padding-right property padding-top property position property rgba
right property text-align
text-decoration text-indent
text-shadow text-transform
top property vertical-align
visibility property width (images) width property (floating elements) width property (tables) word-spacing z-index property