HTML

(Hyper Text Markup Language)

&

CSS

(Cascading Style Sheet)

Designed By **:**

Sheikh Zubair Akhtar.

HTML Introduction

HTML stands for ***Hyper Text Markup Language***. HTML is the basis for all things Web and is a necessary skill for any Web Developer. Almost every website is comprised of HTML whether that is a variation of HTML or plain old HTML. Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.*HTML is not a Programming Language, It is a Markup Language.*

HTML History

HTML was created by ***Tim Berners-Lee*** in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having **HTML-5** version which is an extension to HTML 4.01, and this version was published in 2012.

Web Page

A ***Webpage*** is a document commonly written in Hyper Text Markup Language (HTML) that is accessible through the Internet or other network using an Internet browser. A web page is accessed by entering a URL address and may contain text, graphics, and hyperlinks to other web pages and files. A Web site is a collection of pages. A Web page is an individual HTML document.

Web Site

A ***Web Site*** is a central location of various web pages that are all related and can be accessed by visiting the home page using a Web browser. A Web site is a related collection of World Wide Web (WWW) files that includes a beginning file called a home page. A company or an individual tells you how to get to their Web site by giving you the address of their home page. From the home page, you can get to all the other pages on their site.

Web Browser

A ***Web Browser*** is a piece of software that runs on your personal computer and enables you to view webpage. Web browsers, often simply called **"Browsers"**, interpret the HTML Code and provide a visual layout displayed on the screen. The primary function of a web browser is to render HTML, the code used to design or "markup" webpages. Each time a browser loads a web page, it processes the HTML, which may include text, links, and references to images and other items, such as cascading style sheets and JavaScript functions.

HTML ایک ایسی لینگوئج ہے جو کہ ہمیں ویب پیج بنانے میں مدد دیتی ہے۔اس کی مدد سے ہم کسی بھی ویب سائٹ کا ڈیزائن آسانی سے بناسکتے ہیں۔کسی بھی ویب سائٹ کو بنانے کے لیے ہمیں سب سے پہلے اس زبان کو سیکھنا ضروری ہوتا ہے اس کے بعد ہم کسی بھی ویب سائٹ کا ڈیزائن آسانی سے بناسکتے ہیں۔**HTML** عام طور پر ویب سائٹ کا بنیادی ڈھانچہ بنانے کے لیے استعمال ہوتی ہے اس لینگوئج کی سب سے بڑی خصوصیت یہ ہے کہ اس کہ لیے کسی خاص سوفٹ ویئر کی ضرورت نہیں ہوتی بلکہ آپ اس کی کوڈنگ کو کسی بھی ٹیکسٹ ایڈیٹر میں کر سکتے ہیں جیسے نوٹ پیڈ وغیرہ۔۔۔

HTML Basic Structure

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

Example Explained

<!DOCTYPE html>

This tag defines the document type and HTML version.

<html>

The <html> tag tells the browser that this is an HTML document.

<head>

The <head> element is a container for all the head elements.

<title>

The <title> element defines a title in the browser toolbar.

<body>

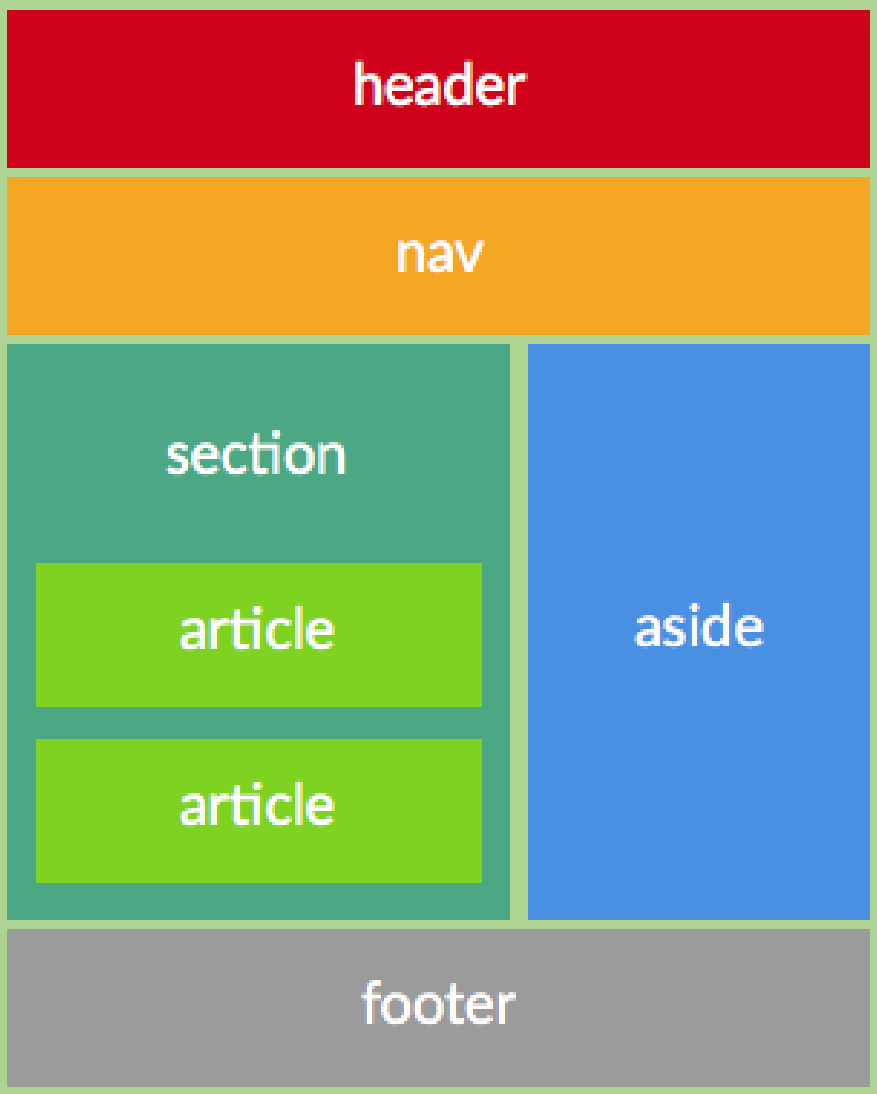
The <body> element contains all the contents of an HTML document, such as text, hyperlinks, images, tables, lists, etc.

< Tag Name> **Start Tag/ Opening Tag**

< / Tag Name> **End Tag/ Closing Tag**

***Note*: All Commands are Lower Case.**

HTML5 Layout Elements.



HTML5 offers new semantic elements that define the different parts of a web page.

<header> Defines a header for a document or a section.
<nav> Defines a container for navigation links.
<section> Defines a section in a document.
<article> Defines an independent self-contained article.
<aside> Defines content aside from the content.
<footer> Defines a footer for a document or a section.

HTML Basic Tags

Headings Tags.

Any document starts with a heading. You can use different sizes for your headings. HTML also has six levels of headings, which use the elements **<h1>, <h2>, <h3>, <h4>, <h5>,** and **<h6>.** While displaying any heading, browser adds one line before and one line after that heading.

Example

```
<h1> This is Heading 1 </h1>
<h2> This is Heading 2 </h2>
<h3> This is Heading 3 </h3>
<h4> This is Heading 4 </h4>
<h5> This is Heading 5 </h5>
<h6> This is Heading 6 </h6>
```

HTML میں H1 سب سے بڑی اور H6 سب سے چھوٹی ہیڈنگ ہوتی ہے۔

Paragraphs Tag.

The ** ** tag offers a way to structure your text into different paragraphs. Each paragraph of text should go in between an opening ** ** and a closing ** ** tag as shown below in the example.

Example

```
<p>This is a Paragraph</p>
<p>This is another Paragraph</p>
```

Line Break Tag.

Whenever you use the **
** element, anything following it starts from the next line. This tag is an example of an empty element, where you do not need opening and closing tags, as there is nothing to go in between them.

Example

```
<p>This is a Paragraph
<br>This is another Paragraph</p>
```

Centering Content.

You can use ** ** tag to put any content in the center of the page or any table cell.

Example

```
<center>
<p>This text in the Center</p>
<h1>This Heading in the Center</h1>
</center>
```

Horizontal Lines.

The **<hr>** tag creates a line from the current position in the document to the right margin and breaks the line accordingly.

Example

```
<hr>
<p>This is some text</p>
<hr>
```

Horizontalکا ٹیگ سنگل ٹیگ ہوتا ہے یہ ٹیگ ویب پیج میں لائن لگانے کے لیے استعمال ہوتا ہے اس کے علاوہ ہم اٹری بیوٹس کی مدد سے لائن کی لمبی اور موٹائی کو کم یا زیادہ کر سکتے ہیں۔

Horizontal Width and Size Attribute.

```
<hr width="50%" size="50px">
```

Text Alignment.

There are three types of alignment associated with the ALIGN tag **Left**, **Center** and **Right**. The following example HTML document details these options.

Text Alignment Property.

```
<h1 align="center or left or
right">This is a Heading</h1>
```

Non Breaking Spaces.

Normally, HTML will only display one space between words, no matter how many times you press the space bar. To force an extra space to show up, type ** ** or ** **.This displays a character called a **"Non-Breaking Space"** which will always appear.

Example

```
<body>
Something    
Really    
Cool
</body>
```

Formatting Elements.

HTML Tag vs. Element.

An HTML element is defined by a starting tag. If the element contains other content, it ends with a closing tag. For example, ** ** is starting tag of a paragraph and ** ** is closing tag of the same paragraph but ** This is paragraph ** is a paragraph element.

- **Bold Text**

```
<b>This text is Bold</b>
```

- **Italic Text**

```
<i>This text is Italic</i>
```

- **Underlined Text**

```
<u>This text is Underlined</u>
```

- **Subscript Text**

```
<sub>This text is Subscripted</sub>
```

- **Superscript Text**

```
<sup>This text is Superscripted</sup>
```

- **Deleted Text**

```
<del>This text is Deleted</del>
```

- **Larger Text**

```
<big>This text is Larger</big>
```

- **Smaller Text**

```
<small>This text is Small</small>
```

- **Emphasized Text**

```
<em>This text is emphasized</em>
```

- **Marked Text**

```
<mark>This text is marked</mark>
```

- **Important Text**

```
<strong>This text is strong</strong>
```

- **Text Abbreviation**

```
<abbr title="Pakistan">Pakistan</abbr>
```

- **Computer Code**

```
<code>This text is Computer Code</code>
```

- **Keyboard Text**

```
<kbd>This is inside kbd element</kbd>
```

- **Address Text**

```
<address>This is Address Text</address>
```

- **Marquee Text**

```
<marquee>This text is Murquee</marquee>
```

HTML Comment Tags.

Comment is a piece of code which is ignored by any web browser. You can add comments to your HTML source by using the following syntax. With comments you can place notifications and reminders in your HTML.

Example

```
<!-- This is a comment -->
<p>This is a Paragraph</p>
<!-- Remember to add more information
here -->
```

HTML Image Tag.

```
<img src="image Name.jpg">
```

HTMLمیں امیج کا ٹیگ سنگل ٹیگ ہوتا ہے اس کے علاوہ ہم امیج کے اٹری بیوٹس کی مدد سے امیج کی لمبائی اور چوڑائی کم یا زیادہ کر سکتے ہیں امیج کے ٹیگ میں Altکا اٹری بیوٹس استعمال ہوتا ہے جس کے اندر کوئی ٹیکسٹ لکھ جاتا ہے اگر کسی وجہ سے امیج شو نہ ہو تو اس کی جگہ وہ ٹیکسٹ شو ہو گا۔

Example

The Alt Attribute.

```
<img src="image Name.jpg" alt="text">
```

Image Size- Width, and Height Attribute.

```
<img src="image Name.jpg"
style="width:200px; height:100px;">
```

Image Border and Title Attribute.

```
<img src="image Name.jpg" border="Solid
2px black" title="Title Text">
```

HTML Text Links.

A webpage can contain various links that take you directly to other pages and even specific parts of a given page. These links are known as hyperlinks. Hyperlinks allow visitors to navigate between Web sites by clicking on words, phrases, and images. Thus you can create hyperlinks using text or images available on a webpage.

Anchor / Hyper Link Tag.

```
<a href="url">link text</a>
```

کسی بھی لنک کو اگلے ٹیب میں اوپن کرنے کے لیے "Target="Blank کا اٹری بیوٹس استعمال ہوتا ہے اس کے علاوہ کسی بھی لنک کا رنگ بھی تبدیل کیا جا سکتا ہے جس کی مثال نیچے دی گئی ہے

Target Blank Attribute.

```
<a href="url" target="blank">link text</a>
```

Link Color Attribute.

```
<a href="url" style="background-color:green;>link text</a>
```

Download Links.

You can create text link to make your PDF, or DOC or ZIP files downloadable. This is very simple you just need to give complete URL of the downloadable file.

```
<ahref="http://www.urdupoint.com/book.pdf ">Downlaod PDF File</a>
```

HTML Table Tags.

HTML Table.

The HTML tables allow web authors to arrange data like text, images, links, other tables, etc. into rows and columns of cells. The HTML tables are created using the **<table>** tag in which the **<tr>** tag is used to create table rows and **<td>** tag is used to create data cells. A table header is defined with the **<th>** tag.

Example

```
<table>
  <tr>
    <th>Name</th>
    <th>RollNo</th>
  </tr>
  <tr>
    <td>Zubair</td>
    <td>445356</td>
  </tr>
</table>
```

Adding a border in HTML Table.

```
<table border="1px solid black">
<tr>
<td>Zubair Akhtar</td>
</tr>
<table>
```

Border Style Types.

- **None** -Defines no border
- **Solid** -Defines a solid border.
- **Collapse** -Defines a Single border
- **Dotted** -Defines a dotted border.
- **Hidden** -Defines a hidden border
- **Dashed** -Defines a dashed border.
- **Double** -Defines a double border.

Table Width and Height.

```
<table width="200px" height="300px">
```

Table Background Color.

```
<table width="200px" height="300px"
bgcolor="green" border="1px">
```

Table Text Alignment.

```
<table width="200px" height="300px"
bgcolor="green" text-align="Center">
```

Table Rows and Columns Span Attributes.

```
<td rowspan="2">Name</td>
<td colspan="2">Roll No</td>
```

Table Cellpadding and Cellspacing Attributes.

```
<table border="1">
<td cellpadding="5">Name</td>
<td cellspacing="5">Roll No</td>
<table>
```

HTML List Tags.

HTML Unordered List.

An unordered list is a collection of related items that have no special order or sequence. This list is created by using HTML **<ul>** tag. Each item in the list is marked with a bullet.

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

- **Coffee**
- **Tea**
- **Milk**

The List Type Attribute

You can use type attribute for **<ul>** tag to specify the type of bullet you like. By default it is a disc. Following are the possible options.

```
<ul type="none">
<ul type="disc">
<ul type="square">
<ul type="circle">
```

- **Disc Example**

```
<ul type="disc">
<li>Lahore</li>
<li>Multan</li>
<li>Layyah</li>
</ul>
```

- **Square Example**

```
<ul type="square">
<li>Lahore</li>
<li>Multan</li>
<li>Layyah</li>
</ul>
```

- **Circle Example**

```
<ul type="circle">
<li>Lahore</li>
<li>Multan</li>
<li>Layyah</li>
</ul>
```

HTML Ordered List.

If you are required to put your items in a numbered list instead of bulleted, then HTML ordered list will be used. This list is created by using **<ol>** tag. The numbering starts at one and is incremented by one for each successive ordered list element tagged with **<li>**.

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

1. Coffee
2. Tea
3. Milk

The List Type Attribute

You can use type attribute for **<ol>** tag to specify the type of numbering you like. By default, it is a number. Following are the possible options.

```
<ol type="1">
```
Default-Case Numerals.

```
<ol type="I">
```
Upper-Case Numerals.

```
<ol type="i">
```
Lower-Case Numerals.

```
<ol type="a">
```
Lower-Case Letters.

```
<ol type="A">
```
Upper-Case Letters.

Type”1”Example.

```
<ol type="1">
<li>Bangladesh</li>
<li>Greenland</li>
<li>Pakistan</li>
<li>Kazakhstan</li>
</ol>
```

Type”I” Example.

```
<ol type="I">
<li>Bangladesh</li>
<li>Greenland</li>
<li>Pakistan</li>
<li>Kazakhstan</li>
</ol>
```

Type”i”Example.

```
<ol type="i">
<li>Bangladesh</li>
<li>Greenland</li>
```

```
<li>Pakistan</li>
<li>Kazakhstan</li>
</ol>
```

Type”a”Example.

```
<ol type="a">
<li>Bangladesh</li>
<li>Greenland</li>
<li>Pakistan</li>
<li>Kazakhstan</li>
</ol>
```

Type”A”Example.

```
<ol type="A">
<li>Bangladesh</li>
<li>Greenland</li>
<li>Pakistan</li>
<li>Kazakhstan</li>
</ol>
```

The Start Attribute.

You can use start attribute for **<ol>** tag to specify the starting point of numbering you need. Following are the possible options.

```
<ol type="1" start="4">
```
Numerals starts with 4.

```
<ol type="I" start="4">
```
Numerals starts with IV.

```
<ol type="i" start="4">
```
Numerals starts with iv.

```
<ol type="a" start="4">
```
Letters starts with d.

```
<ol type="A" start="4">
```
Letters starts with D.

HTML Forms.

HTML Forms are required, when you want to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc. There are various form elements available like text fields, text area fields, drop-down menus, radio buttons, checkboxes, etc.

Text Input Controls

There are three types of text input used on forms.

- **Single Line Text Input Controls.**

This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML **<input>** tag.

Example

```
<body>
<form>
First name:<input type="text"
name="first_name"/>
<br>
Last name:<input type="text"
name="last_name"/>
</form>
</body>
```

- **Multi Line Text Input Controls.**

This is used when the user is required to give details that may be longer than a single sentence.Multi-line input controls are created using HTML **<textarea>** tag.

Example

```
<body>
<form>
Description:<br>
<textarea rows="5" cols="50"
name="description">
Enter description here...
</textarea>
</form>
</body>
```

- **Password Input Controls.**

This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML **<input>** tag but type attribute is set to **password**.

Example

```
<body>
<form>
User ID: <input type="text"
name="user_id"/>
<br>
Password: <input type="password"
name="password"/>
</form>
</body>
```

Attribute:

type ,name ,value ,size , maxlength.

Checkbox Control

Checkboxes are used when more than one option is required to be selected. They are also created using HTML **<input>** tag but type attribute is set to **checkbox.**

Example

```
<body>
<form>
<input type="checkbox" name="maths"
value="on"> Maths
<input type="checkbox" name="physics"
value="on"> Physics
</form>
</body>
```

Attributes

Type, name, value, checked

Radio Button Control

Radio buttons are used when out of many options, just one option is required to be selected. They are also created using HTML **<input>** tag but type attribute is set to **radio.**

Example

```
<body>
<form>
<input type="radio" name="subject1"
value="maths"> Maths
<input type="radio" name="subject2"
value="physics"> Physics
</form>
</body>
```

Select Box Control

A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

Example

```
<body>
<form>
<select name="dropdown">
<option value="Maths">Maths</option>
<option value="Physics">Physics</option>
</select>
</form>
</body>
```

File Upload Box

If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the **<input>** element but type attribute is set to **file.**

Example

```
<body>
<form>
<input type="file" name="fileupload"
accept="image/*" />
</form>
</body>
```

Button Controls

There are various ways in HTML to create clickable buttons. You can also create a clickable button using **<input>** tag by setting its type attribute to **button**. The type attribute can take the following values:

Type

Submit, reset, button, image

Example

```
<body>
<form>
<input type="submit" name="submit"
value="Submit" />
<input type="reset" name="reset"
value="Reset" />
<input type="button" name="ok" value="OK"
/>
<input type="image" name="imagebutton"
src="/html/images/logo.png" />
</form>
</body>
```

HTML Embed Multimedia

Sometimes you need to add music or video into your web page. The easiest way to add video or sound to your web site is to include the special HTML tag called **<embed>**. This tag causes the browser itself to include controls for the multimedia automatically provided browser supports <embed> tag and given media type. You can also include a **<noembed>** tag for the browsers which don't recognize the **<embed>** tag. You could, for example, use **<embed>** to display movie of your choice, and **<noembed>** to display a single JPG image if browser does not support **<embed>** tag.

Example

```
<html>
<body>
<embed src="/html/yourfile.mid"
width="100%" height="60">
<noembed><img src="yourimage.gif"
alt="Alternative Media" ></noembed>
</embed>
</body>
</html>
```

Supported Video Types

Format	File	Format	File
MPEG	.mpg .mpeg	**Flash**	.swf .flv
AVI	.avi	**OGG**	.ogg
WMV	.wmv	**WebM**	.webm
QuickTime	.mov	**MP4**	.mp4

Background-Audio

You can use HTML **<bgsound>** tag to play a soundtrack in the background of your webpage. This tag is supported by Internet Explorer only and most of the other browsers ignore this tag. It downloads and plays an audio file when the host document is first downloaded by the user and displayed. The background sound file also will replay whenever the user refreshes the browser. This tag is having only two attributes loop and src. Both these attributes have same meaning as explained above.

Example

```
<html>
<body>
<bgsound src="/html/yourfile.mid">
<noembed><img src="yourimage.gif"
></noembed>
</bgsound>
</body>
</html>
```

HTML Iframe Tag.

An **iframe** is used to display a web page within a web page .An HTML iframe is defined with the **<iframe>** tag.

Example

```
<iframe src="URL"></iframe>
```

Iframe Size and Border.

Size :

```
<iframe src="URL" height="200" wid
th="300"></iframe>
```

Border :

```
<iframe src="URL" style="border
:2px solid grey;"></iframe>
```

Note: **All HTML Document is Completed.**

(Cascading Style Sheet)

CSS Introduction

What is CSS?

Cascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, as well as a variety of other effects. CSS is easy to learn and understand but it provides a powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

CSS History

Cascading Style Sheets (CSS) describe how documents are presented on screens, in print, or perhaps how they are pronounced. W3C has actively promoted the use of style sheets on the Web since the consortium was founded in 1994. Cascading Style Sheets (CSS) provide easy and effective alternatives to specify various attributes for the HTML tags. Using CSS, you can specify a number of style properties for a given HTML element. Each property has a name and a value, separated by a colon (:). Each property declaration is separated by a semi-colon (;).

Advantages of CSS

- **CSS Saves Time** - You can write CSS once and then reuse the same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many web pages as you want.

- **Pages Load Faster** - If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So, less code means faster download times.
- **Easy Maintenance** - To make a global change, simply change the style, and all the elements in all the web pages will be updated automatically.

Three Ways to Insert CSS

There are three ways of inserting a style sheet**:**

- **External Style Sheet**
- **Internal Style Sheet**
- **Inline Style**

1: External Style Sheet

With an **External Style Sheet**, you can change the look of an entire Website by changing just one file. If you need to use your style sheet to various pages, then its always recommended to define a common style sheet in a separate file. A cascading style sheet file will have extension as **.css** and Each page must include a reference to the **External Style sheet** file inside the **<link>** element. The **<link>** element goes inside the **<head>** section.

Example

```
<head>
<link rel="stylesheet" type="text/css"
href="mystyle.css">
</head>
```

2: Internal Style Sheet.

An **Internal Style Sheet** is used if one single page has a unique style. **Internal Styles** are defined within the **<style>** element, inside the **<head>** section of an HTML page. If you want to apply Style Sheet rules to a single document only, then you can include those rules in header section of the HTML document using **<style>** tag. Rules defined in **Internal Style Sheet** overrides the rules defined in an external CSS file.

Example

```
<head>
<style>
body {background-color: linen;}
h1 {color: maroon; margin-left: 40px;}
</style>
</head>
```

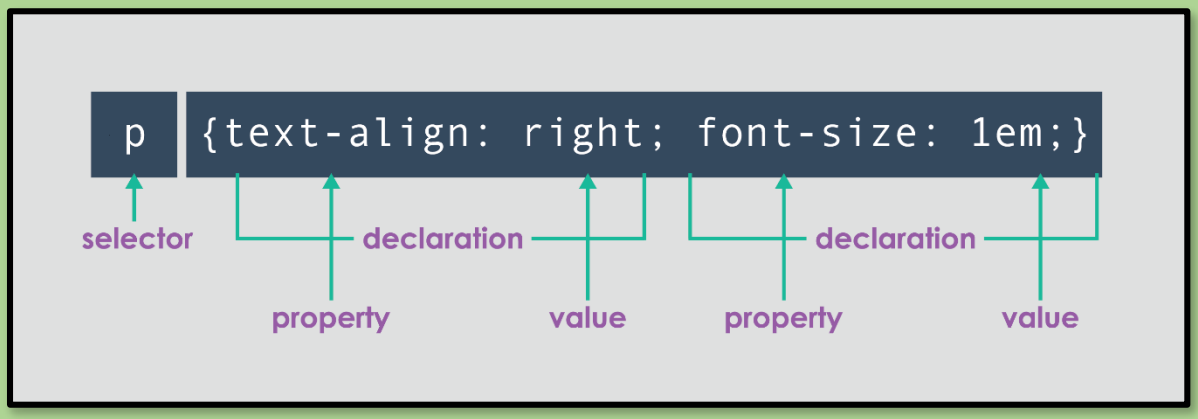
3: Inline Style Sheet

You can apply style sheet rules directly to any HTML element using style attribute of the relevant tag. This should be done only when you are interested to make a particular change in any HTML element only. Rules defined **Inline** with the element overrides the rules defined in an external CSS file as well as the rules defined in **<style>** element. An inline style is used to apply a unique style for a single element. To use **Inline Styles**, add the style attribute to the relevant element. The style attribute can contain any CSS property. The example below shows how to change the color and the left margin of a **<h1>** element:

Example

```
<h1 style="color:green;margin-left:30px;"
>This is a heading</h1>
```

CSS Syntax



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

•**Selector:** A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.

•**Property:** A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be color, border, etc.

•**Value:** Values are assigned to properties. For example, color property can have the value either red or #F1F1F1 etc. You can put CSS Style Rule Syntax as follows:

```
selector { property: value }
```

Example

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

CSS Selectors

1: The Element Selector

The element selector selects elements based on the element name. You can select all **<p>** elements on a page like this in this case, all **<p>** elements will be center-aligned, with a red text color.

Example

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

2: The ID Selector

The **ID selector** uses the id attribute of an HTML element to select a specific element. The id of an element should be unique within a page, so the id selector is used to select one unique element. To select an element with a specific id, write a hash (**#**) character, followed by the id of the element. The style rule below will be applied to the HTML element with **id="para1"**.

Example

```
#para1 {
    color: red;
    text-align: center;}
#para2 {
    font-size: 10px;
    text-decoration: none;}
```

Note: An **ID** and **Class** name cannot start with a Number!

3: The Class Selector

The **Class Selector** selects elements with a specific class attribute. To select elements with a specific class, write a period **(.)** character, followed by the name of the class. In the example below, all HTML elements with **class="center"** will be red and center-aligned.

Example

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

4: Grouping Selectors

It will be better to group the selectors, to minimize the code. To group selectors, separate each selector with a comma. In the example below we have grouped the selectors from the code above.

Example

```
p,h1,h2,h3{
    color: red;
    text-align: center;}
```

CSS Comments

Comments are used to explain the code, and may help when you edit the source code at a later date. Comments are ignored by browsers. A CSS comment starts with /* and ends with */. Comments can also span multiple lines.

Example

```
p {
    color: red;
    /* This is a single-line comment */
    text-align: center;}
```

CSS Backgrounds

1: Background Color

The `background-color` property specifies the background color of an element. The background color of a page is set like this.

Example

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

2: Background Image

The `background-image` property specifies an image to use as the background of an element. By default, the image is repeated so it covers the entire element. The background image for a page can be set like this.

Example

```
body {
    background-image: url("image.png");}
```

Repeat Horizontally or Vertically.

By default, the background-image property repeats an image both horizontally and vertically. If the images above is repeated horizontally `(background-repeat: repeat-x;)`, To repeat an image vertically, set `background-repeat: repeat-y;` the background will look better.

Example

```
body {
    background-image: url("image.png");
    background-repeat: repeat-x;
    background-repeat: repeat-y;}
```

Set Position and no-repeat.

Showing the background image only once is also specified by the background-repeat property. We want to change the position of the image, so that it does not disturb the text too much. The position of the image is specified by the background-position property.

Example

```
body {
    background-image: url("image.png");
    background-repeat: no-repeat;
    background-position: right top;}
```

CSS Border Properties

The CSS border properties allow you to specify the style, width, and color of an element's border.

Border Style

The border-style property specifies what kind of border to display.

- **None** -Defines no border
- **Solid** -Defines a solid border.
- **Collapse** -Defines a Single border
- **Dotted** -Defines a dotted border.
- **Hidden** -Defines a hidden border
- **Dashed** -Defines a dashed border.
- **Double** -Defines a double border.

The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).

Example

```
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid  {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge  {border-style: ridge;}
p.inset  {border-style: inset;}
p.outset {border-style: outset;}
p.none   {border-style: none;}
p.hidden {border-style: hidden;}
```

Border Width

The `border-width` property specifies the width of the four borders. The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: **thin, medium, or thick**. The `border-width` property can have from one to four values for the top border, right border, bottom border, and the left border.

Example

```
p.one {
    border-style: solid;
    border-color: black;
    border-width: 5px;}
p.two {
    border-style: solid;
    border-color: red;
    border-width: medium;}
    border-width: 2px 10px 4px 20px;}
```

Rounded Borders

The `border-radius` property is used to add rounded borders to an element.

Example

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

CSS Margins

The CSS `margin` properties are used to generate space around elements. The `margin` properties set the size of the white space outside the border. With CSS, you have full control over the margins. There are CSS properties for setting the margin for each side of an element (**top**, **right**, **bottom**, and **left**).

Margin - Individual Sides

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

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

Example

```
p {
    margin-top: 100px;
    margin-bottom: 100px;
    margin-right: 150px;
    margin-left: 80px;}
```

CSS Padding

The CSS Padding properties are used to generate space around content. The padding clears an area around the content (**inside the border**) of an element. With CSS, you have full control over the padding. There are CSS properties for setting the padding for each side of an element (top, right, bottom, and left).

Padding - Individual Sides

CSS has properties for specifying the padding for each side of an element.

- padding-top
- padding-right
- padding-bottom
- padding-left

Example

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

Padding - Shorthand Property

To shorten the code, it is possible to specify all the padding properties in one property. The padding property is a shorthand property for the following individual padding properties.

- padding-top
- padding-right
- padding-bottom
- padding-left

CSS Height and Width

Setting height and width.

The height and width properties are used to set the height and width of an element. The height and width can be set to auto (this is default. Means that the browser calculates the height and width), or be specified in length values, like px, cm, etc., or in percent (%) of the containing block. This element has a height of 200 pixels and a width of 50%.

Example

```
div {
    height:200px;
    width:50%;
    background-color: powderblue;}
```

Note: The height and width properties do not include padding, borders, or margins; they set the height/width of the area inside the padding, border, and margin of the element.

Setting max-width

The max-width property is used to set the maximum width of an element .The max-width can be specified in length values, like px, cm, etc., or in percent (%) of the containing block, or set to none (this is default. Means that there is no maximum width). The following example shows a element with a height of 100 pixels and a max-width of 500 pixels.

Example

```
div {
    max-width: 500px;
    height: 100px;
   background-color: powderblue;}
```

CSS Box Model

All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout. The CSS box model is essentially a box that wraps around every HTML element. It consists of margins, borders, padding, and the actual content.

Explanation of the different part:

- **Content:** The content of the box, where text and images appear.
- **Padding:** Clears an area around the content. The padding is transparent.
- **Border:** A border that goes around the padding and content.
- **Margin:** Clears an area outside the border. The margin is transparent.

Example

```
div {
    width: 300px;
    border: 25px solid green;
    padding: 25px;
    margin: 25px;}
```

Width and Height of an Element

In order to set the width and height of an element correctly in all browsers, you need to know how the box model works .When you set the width and height properties of an element with CSS, you just set the width and height of the content area. To calculate the full size of an element, you must also add padding, borders and margins. **Assume we want to style a element to have a total width of 350px:**

Example

```
div {
    width: 320px;
    padding: 10px;
    border: 5px solid gray;
    margin: 0;}
```

Here is the math:
320px (width)
+ 20px (left + right padding)
+ 10px (left + right border)
+ 0px (left + right margin)
= 350px

CSS Outline

The CSS `outline` properties specify the style, color, and width of an outline. An outline is a line that is drawn around elements (outside the borders) to make the element "stand out". However, the outline property is different from the border property. The outline is NOT a part of an element's dimensions; the element's total width and height is not affected by the width of the outline. This element has a thin black border and a double outline that is 10px wide and green.

Example.

```
p {
    border: 1px solid black;
    outline-color: red;}
```

Outline Style

The `outline-style` property specifies the style of the outline. The `outline-style` property can have one of the following values.

- **None** -Defines no border
- **Solid** -Defines a solid border.
- **Collapse** -Defines a Single border
- **Dotted** -Defines a dotted border.
- **Hidden** -Defines a hidden border
- **Dashed** -Defines a dashed border.
- **Double** -Defines a double border.

The following example first sets a thin black border around each <p> element, then it shows the different outline-style values.

Example.

```
p {
    border: 1px solid black;
    outline-color: red;}

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

Outline Color

The `outline-color` property is used to set the color of the outline. The color can be set by:

- **Name** - specify a color name, like "red"
- **RGB** - specify a RGB value, like "rgb(255,0,0)"
- **Hex** - specify a hex value, like "#ff0000"

Example.

```
p {
    border: 1px solid black;
    outline-style: double;
    outline-color: red;}
```

Outline Width

The `outline-width` property specifies the width of the outline. The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick.

Example.

```
p {border: 1px solid black;}

p.one {
    outline-style: double;
    outline-color: red;
    outline-width: thick;}

p.two {
    outline-style: double;
    outline-color: green;
    outline-width: 3px;}
```

CSS Text

Text Color

The `color property` is used to set the color of the text. With CSS, a color is most often specified by.

- a **Color name** - like "red".
- a **HEX value** - like "#ff0000".
- an **RGB value** - like "rgb(255,0,0)".

Example

```
body {
    color: blue;}
h1 {
    color: green;}
```

Text Alignment

The `text-align` property is used to set the horizontal alignment of a text. A text can be left or right aligned, centered, or justified. The following example shows center aligned, and left and right aligned text (left alignment is default if text direction is left-to-right, and right alignment is default if text direction is right-to-left).

Example

```
h1 {
    text-align: center;}
h2 {
    text-align: left;}
h3 {
    text-align: right;}
h4 {
    text-align: justify;}
```

Text Decoration

The `text-decoration` property is used to set or remove decorations from text. The value `text-decoration:none;` is often used to remove underlines from links.

Example

```
a {
    text-decoration: none;}
h1 {
    text-decoration: overline;}
h2 {
    text-decoration: line-through;}
h3 {
    text-decoration: underline;}
```

Text Transformation

The `text-transform` property is used to specify uppercase and lowercase letters in a text. It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word.

Example

```
p.uppercase {
    text-transform: uppercase;}
p.lowercase {
    text-transform: lowercase;}
p.capitalize {
    text-transform: capitalize;}
```

Text Indentation

The `text-indent` property is used to specify the indentation of the first line of a text.

Example

```
p {
    text-indent: 50px;}
```

Letter Spacing

The `letter-spacing` property is used to specify the space between the characters in a text. The following example demonstrates how to increase or decrease the space between characters.

Example

```
h1 {
    letter-spacing: 3px;}
h2 {
    letter-spacing: -3px;}
```

Line Height

The `line-height` property is used to specify the space between lines.

Example

```
p.small {
    line-height: 0.8;}
p.big {
    line-height: 1.8;}
```

Text Shadow

The `text-shadow` property adds shadow to text. The following example specifies the position of the horizontal shadow (3px), the position of the vertical shadow (2px) and the color of the shadow (red).

Example

```
h1 {
    text-shadow: 3px 2px red;}
```

CSS Fonts

Font Family

The `font family` of a text is set with the font-family property. The font-family property should hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font, and so on. Start with the font you want, and end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available.

Note: If the name of a font family is more than one word, it must be in quotation marks, like: "Times New Roman". More than one font family is specified in a comma-separated list.

Example

```
p {
    font-family: "Times New Roman", Times,
serif;}
```

Font Style

The `font-style` property is mostly used to specify italic text. This property has three values.

- **Normal** - The text is shown normally
- **Italic** - The text is shown in italics
- **Oblique** - The text is "leaning" (oblique is very similar to italic, but less supported)

Example

```
p.normal {
    font-style: normal;}
p.italic {
    font-style: italic;}
p.oblique {
    font-style: oblique;}
```

Font Size

The `font-size` property sets the size of the text. Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs. Always use the proper HTML tags, like <h1> - <h6> for headings and <p> for paragraphs. The font-size value can be an absolute, or relative size.

Example

```
h1 {
    font-size: 40px;}
h2 {
    font-size: 30px;}
```

Set Font Size With Pixels

Setting the `text-size` with pixels gives you full control over the text size.

Example

```
h1 {
    font-size: 40px;}
h2 {
    font-size: 30px;}
p {
    font-size: 14px;}
```

Set Font Size With Em

To allow users to resize the text (in the browser menu), many developers use em instead of pixels. The em size unit is recommended by the W3C. 1em is equal to the current font size. The default text size in browsers is 16px. So, the default size of 1em is 16px. The size can be calculated from pixels to em using this formula: pixels/16=em .

Example

```
body {
    font-size: 100%;}
h1 {
   font-size: 2.5em;} /* 40px/16=2.5em */
h2 {
  font-size: 1.875em;} /*30px/16=1.875em*/
p {
  font-size: 0.875em;}/*14px/16=0.875em */
```

Font Weight

The `font-weight` property specifies the weight of a font.

Example

```
p.normal {
    font-weight: normal;}
p.thick {
    font-weight: bold;}
```

Font Variant

The `font-variant` property specifies whether or not a text should be displayed in a small-caps font. In a small-caps font, all lowercase letters are converted to uppercase letters. However, the converted uppercase letters appears in a smaller font size than the original uppercase letters in the text.

Example

```
p.normal {
    font-variant: normal;}
p.small {
    font-variant: small-caps;}
```

CSS Icons

How To Add Icons

The simplest way to add an icon to your HTML page, is with an icon library, such as Font Awesome. Add the name of the specified icon class to any inline HTML element (like <i> or). All the icons in the icon libraries below, are scalable vectors that can be customized with CSS (size, color, shadow, etc.).

Font Awesome Icons

To use the Font Awesome icons, add the following line inside the **<head>** section of your HTML page.

Example

```
<html>
<head>
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
</head>
<body>
<i class="fa fa-cloud"></i>
<i class="fa fa-heart"></i>
<i class="fa fa-car"></i>
<i class="fa fa-file"></i>
<i class="fa fa-bars"></i>
</body>
</html>
```

Note: No downloading or installation is required!

CSS Links

Styling Links

Links can be styled with any CSS property (e.g. color, font-family, background, etc.).

Example

```
a { color: hotpink;}
```

The four links states are:

- **a:link** - a normal, unvisited link.
- **a:visited** - a link the user has visited.
- **a:hover** - a link when the user mouses over it.
- **a:active** - a link the moment it is clicked.

Example

```
/* unvisited link */
a:link {
    color: red;}
/* visited link */
a:visited {
    color: green;}
/* mouse over link */
a:hover {
    color: hotpink;}
/* selected link */
a:active {
    color: blue;}
```

- a:hover MUST come after a:link and a:visited
- a:active MUST come after a:hover

Link Text Decoration

The `text-decoration` property is mostly used to remove underlines from links.

Example

```
a:link {
    text-decoration: none;}
a:visited {
    text-decoration: none;}
a:hover {
    text-decoration: underline;}
a:active {
    text-decoration: underline;}
```

Link Background Color

The `background-color` property can be used to specify a background color for links.

Example

```
a:link {
    background-color: yellow;}
a:visited {
    background-color: cyan;}
a:hover {
    background-color: lightgreen;}
a:active {
    background-color: hotpink;}
```

Advanced - Link Buttons

This example demonstrates a more advanced example where we combine several CSS properties to display links as boxes/buttons.

Example

```
a:link, a:visited {
    background-color: #f44336;
    color: white;
    padding: 14px 25px;
    text-align: center;
    text-decoration: none;
    display: inline-block;}

a:hover, a:active {
    background-color: red;}
```

CSS Lists

HTML Lists and CSS List Properties

In HTML, there are two main types of lists:

- **Unordered Lists** (<ul>) marked with bullets.
- **Ordered Lists** (<ol>) marked with numbers or letters

The CSS list properties allow you to.

- Set different list item markers for ordered lists.
- Set different list item markers for unordered lists.
- Set an image as the list item marker.
- Add background colors to lists and list items.

Different List Item Markers

The `list-style-type` property specifies the type of list item marker. The following example shows some of the available list item markers.

Example

```
ul.a {
    list-style-type: circle;}
ul.b {
    list-style-type: square;}
ol.c {
    list-style-type: upper-roman;}
ol.d {
    list-style-type: lower-alpha;}
```

An Image as The List Item Marker

The `list-style-image` property specifies an image as the list item marker.

Example

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

Styling List With Colors

We can also `style lists` with colors, to make them look a little more interesting. Anything added to the <ol> or <ul> tag, affects the entire list, while properties added to the <li> tag will affect the individual list items.

Example

```
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;}
```

CSS Tables

The look of an HTML table can be greatly improved with CSS.

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Berglunds snabbköp	Christina Berglund	Sweden
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Königlich Essen	Philip Cramer	Germany
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

Table Borders

To specify table borders in CSS, use the border property. The example below specifies a black border for **<table>, <th>,** and **<td>** elements.

Example

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

Collapse Table Borders

The `border-collapse` property sets whether the table borders should be collapsed into a single border.

Example

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

Table Width and Height

`Width` and `height` of a table are defined by the width and height properties. The example below sets the width of the table to 100%, and the height of the **<th>** elements to 50px.

Example

```
    table {
        width: 100%;}
    th {
        height: 50px;}
```

Horizontal Alignment

The `text-align` property sets the horizontal alignment (like left, right, or center) of the content in <th> or <td>. By default, the content of <th> elements are center-aligned and the content of <td> elements are left-aligned. The following example left-aligns the text in <th> elements.

Example

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

Vertical Alignment

The `vertical-align` property sets the vertical alignment (like top, bottom, or middle) of the content in <th> or <td>. By default, the vertical alignment of the content in a table is middle (for both <th> and <td> elements). The following example sets the vertical text alignment to bottom for <td> elements.

Example

```
td {
    height: 50px;
    vertical-align: bottom;}
```

Table Padding

To control the space between the border and the content in a table, use the padding property on <td> and <th> elements.

Example

```
th, td {
    padding: 15px;
    text-align: left;}
```

Responsive Table

A responsive table will display a horizontal scroll bar if the screen is too small to display the full content.

First Name	Last Name	Points	Points	Points	Points	Points
Jill	Smith	50	50	50	50	50
Eve	Jackson	94	94	94	94	94
Adam	Johnson	67	67	67	67	67

CSS Layout - The Display Property

The display property is the most important CSS property for controlling layout.

Inline Elements

An inline element does not start on a new line and only takes up as much width as necessary. This is an inline **** element inside a paragraph.

Examples of inline elements.

- ****
- **<a>**
- **<img>**

Display: none;

display: none; is commonly used with JavaScript to hide and show elements without deleting and recreating them. Take a look at our last example on this page if you want to know how this can be achieved. The <script> element use display: none; as its default.

Override The Default Display Value

As mentioned, every element has a default display value. However, you can override this. Changing an inline element to a block element, or vice versa, can be useful for making the page look a specific way, and still follow the web standards. A common example is making inline <li> elements for horizontal menus:

Example

```
li {
    display: inline;}
a {
    display: block;}
```

CSS Layout - The Position Property

The position property specifies the type of positioning method used for an element (**static, relative, fixed** or **absolute**).

The Position Property

The position property specifies the type of positioning method used for an element.

There are four different position values:

- **Static**
- **Relative**
- **Fixed**
- **Absolute**

Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the position value.

Position: static;

HTML elements are positioned static by default. Static positioned elements are not affected by the top, bottom, left, and right properties. An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page: This element has `position: static;`

Here is the CSS that is used:

Example

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

Position: relative;

An element with `position:relative;` is positioned relative to its normal position. Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element. This element has `position: relative;`

Here is the CSS that is used:

Example

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

Position: fixed;

An element with `position:fixed;` is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element. A fixed element does not leave a gap in the page where it would normally have been located. Notice the fixed element in the lower-right corner of the page.

Here is the CSS that is used:

Example

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

Position: absolute;

An element with `position:absolute;` is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed). However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

Here is a simple example:

Example

```
div.relative {
    position: relative;
    width: 400px;
    height: 200px;
    border: 3px solid #73AD21;}
div.absolute {
    position: absolute;
    top: 80px;
    right: 0;
    width: 200px;
    height: 100px;
    border: 3px solid #73AD21;}
```

CSS Layout – Overflow

CSS Overflow

The **CSS overflow** property specifies whether to clip content or to add scrollbars when the content of an element is too big to fit in a specified area.

Visible

By default, the overflow is visible, meaning that it is not clipped and it renders outside the element's box. You can use the overflow property when you want to have better control of the layout. The overflow property specifies what happens if content overflows an element's box.

Example

```
div {
    width: 200px;
    height: 50px;
    background-color: #eee;
    overflow: visible;}
```

Hidden

With the **hidden** value, the overflow is clipped, and the rest of the content is hidden.

Example

```
div {
    overflow: hidden;}
```

Scroll

Setting the value to scroll, the overflow is clipped and a scrollbar is added to scroll inside the box. Note that this will add a scrollbar both horizontally and vertically (even if you do not need it).

Example

```
div {
    overflow: scroll;}
```

Auto

The auto value is similar to scroll, only it add scrollbars when necessary.

Example

```
div {
    overflow: auto;}
```

Overflow-x and Overflow-y

The `overflow-x` and overflow-y properties specifies whether to change the overflow of content just horizontally or vertically (or both).

- **overflow-x** specifies what to do with the left/right edges of the content.
- **overflow-y** specifies what to do with the top/bottom edges of the content.
- **Example**

```
div {
    overflow-x: hidden; /* Hide horizontal scrollbar */
    overflow-y: scroll; /* Add vertical scrollbar */}
```

CSS Layout - Float and Clear

The float property specifies whether or not an element should float. The clear property is used to control the behavior of floating elements.

The Float Property

In its simplest use, the float property can be used to wrap text around images. The following example specifies that an image should float to the right in a text.

Example

```
img {
    float: right;
    margin: 0 0 10px 10px;}
```

The Clear Property

The **clear property** is used to control the behavior of floating elements. Elements after a floating element will flow around it. To avoid this, use the clear property. The clear property specifies on which sides of an element floating elements are not allowed to float.

Example

```
div {
    clear: left;}
```

CSS Opacity / Transparency

The **opacity property** specifies the opacity/transparency of an element.

Transparent Image

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

Example

```
img {
    opacity: 0.5;
    filter: alpha(opacity=50); /* For IE8 and earlier */}
```

Transparent Hover Effect

The opacity property is often used together with the :hover selector to change the opacity on mouse-over.

Example

```
img {
    opacity: 0.5;
    filter: alpha(opacity=50); /* For IE8 and earlier */}
img:hover {
    opacity: 1.0;
    filter: alpha(opacity=100); /* For IE8 and earlier */}
img:hover { opacity: 0.5;
    filter: alpha(opacity=50); /* For IE8 and earlier */}
```

Transparent Box

When using the opacity property to add transparency to the background of an element, all of its child elements become transparent as well. This can make the text inside a fully transparent element hard to read.

Example

```
div {
    opacity: 0.3;
    filter: alpha(opacity=30); /* For IE8 and earlier */}
```

CSS Navigation Bar

Navigation Bars

Having easy-to-use navigation is important for any web site. With CSS you can transform boring HTML menus into good-looking navigation bars.

Navigation Bar = List of Links

A navigation bar needs standard HTML as a base. In our examples we will build the navigation bar from a standard HTML list. A navigation bar is basically a list of links, so using the <ul> and <li> elements makes perfect sense.

Example

```
<ul>
 <li><a href="news.htm">Home</a></li>
 <li><a href="about.htm">About</a></li>
 <li><a href="default.htm">News</a></li>
 <li><a href="contact.htm">Login</a></li>
</ul>
```

Now let's remove the bullets and the margins and padding from the list:

Example

```
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;}
```

Example explained:

- **list-style-type: none;** - Removes the bullets. A navigation bar does not need list markers.
- Set **margin: 0;** and **padding: 0;** to remove browser default settings.

Vertical Navigation Bar

To build a vertical navigation bar, you can style the <a> elements inside the list, in addition to the code above.

Example

```
li a {
    display: block;
    width: 60px;}
```

Example explained:

- **display: block;** - Displaying the links as block elements makes the whole link area clickable (not just the text), and it allows us to specify the width (and padding, margin, height, etc. if you want)
- **width: 60px;** - Block elements take up the full width available by default. We want to specify a 60 pixels width

Example

```
ul {list-style-type: none;
    margin: 0;
    padding: 0;
    width: 60px;}
li a { display: block;}
```

Horizontal Navigation Bar

There are two ways to create a horizontal navigation bar. Using inline or floating list items.

Inline List Items

One way to build a horizontal navigation bar is to specify the <li> elements as inline, in addition to the "standard" code above.

Example

```
li { display: inline;}
```

Fixed Navigation Bar

Make the navigation bar stay at the top or the bottom of the page, even when the user scrolls the page.

Fixed Top

```
ul {position: fixed;
    top: 0;
    width: 100%;}
```

Fixed Bottom

```
ul {position: fixed;
    bottom: 0;
    width: 100%;}
```

CSS Dropdowns

Basic Dropdown

Create a dropdown box that appears when the user moves the mouse over an element.

Example

```
<style>
.dropdown {
    position: relative;
    display: inline-block;}
.dropdown-content {
    display: none;
    position: absolute;
    background-color: #f9f9f9;
    min-width: 160px;
    box-shadow: 0px 8px 16px 0px
rgba(0,0,0,0.2);
    padding: 12px 16px;
    z-index: 1;}
.dropdown:hover .dropdown-content {
    display: block;}
</style>
<div class="dropdown">
  <span>Mouse over me</span>
  <div class="dropdown-content">
    <p>Hello World!</p>
  </div> </div>
```

Dropdown Menu

Create a dropdown menu that allows the user to choose an option from a list. This example is similar to the previous one, except that we add links inside the dropdown box and style them to fit a styled dropdown button.

Example

```
<style>
/* Style The Dropdown Button */
.dropbtn {
    background-color: #4CAF50;
    color: white;
    padding: 16px;
    font-size: 16px;
    border: none;
    cursor: pointer;}
.dropdown {
    position: relative;
    display: inline-block;}
/* Dropdown Content Hidden by Default */
.dropdown-content {
    display: none;
    position: absolute;
    background-color: #f9f9f9;
    min-width: 160px;
    box-shadow: 0px 8px 16px 0px
rgba(0,0,0,0.2);}
/* Links inside the dropdown */
```

```
.dropdown-content a {
    color: black;
    padding: 12px 16px;
    text-decoration: none;
    display: block;}
/* Change color of dropdown links on hover */
.dropdown-content a:hover {background-
color: #f1f1f1}
/* Show the dropdown menu on hover */
.dropdown:hover .dropdown-content {
    display: block;}
/* Change the background color of the dropdown
button when the dropdown content is shown */
.dropdown:hover .dropbtn {
    background-color: #3e8e41;}
</style>
<div class="dropdown">
  <button
class="dropbtn">Dropdown</button>
  <div class="dropdown-content">
    <a href="#">Link 1</a>
    <a href="#">Link 2</a>
    <a href="#">Link 3</a>
  </div>
</div>
```

CSS Tooltip

Create a Tooltip that appears when the user moves the mouse over an element.

Example

```
<style>
/* Tooltip container */
.tooltip {
    position: relative;
    display: inline-block;
    border-bottom: 1px dotted black; }
/* Tooltip text */
.tooltip .tooltiptext {
    visibility: hidden;
    width: 120px;
    background-color: black;
    color: #fff;
    text-align: center;
    padding: 5px 0;
    border-radius: 6px;
    position: absolute;
    z-index: 1;}
/* Show the tooltip text when you mouse over
the tooltip container */
.tooltip:hover .tooltiptext {
    visibility: visible;}
</style> <div class="tooltip">Hover over me
  <span class="tooltiptext">Tooltip
text</span></div>
```

Positioning Tooltips

In this example, the tooltip is placed to the right (left:105%) of the "hoverable" text ( ). Also note that top:-5px is used to place it in the middle of its container element. We use the number **5** because the tooltip text has a top and bottom padding of 5px.

Right Tooltip

```
.tooltip .tooltiptext {
    top: -5px;
    left: 105%;}
```

Left Tooltip

```
.tooltip .tooltiptext {
    top: -5px;
    right: 105%;}
```

Top Tooltip

```
.tooltip .tooltiptext {
    width: 120px;
    bottom: 100%;
    left: 50%;
    margin-left: -60px;}
```

Bottom Tooltip

```
.tooltip .tooltiptext {
    width: 120px;
    top: 100%;
    left: 50%;
    margin-left: -60px;}
```

Tooltip Arrows

To create an arrow that should appear from a specific side of the tooltip, add "empty" content after tooltip, with the pseudo-element class ::after together with the content property. The arrow itself is created using borders. This will make the tooltip look like a speech bubble. This example demonstrates how to add an arrow to the bottom of the tooltip.

Bottom Arrow

```
.tooltip .tooltiptext::after {
    content: " ";
    position: absolute;
    top: 100%; /* At the bottom of the
tooltip */
    left: 50%;
    margin-left: -5px;
    border-width: 5px;
    border-style: solid;
    border-color: black;}
```

Example Explained

Position the arrow inside the tooltip: top: 100% will place the arrow at the bottom of the tooltip. left: 50%will center the arrow.

CSS Forms

The look of an HTML form can be greatly improved with CSS.

Styling Input Fields

Use the width property to determine the width of the input field.

Example

```
input {
    width: 100%;}
```

The example above applies to all <input> elements. If you only want to style a specific input type, you can use attribute selectors.

- **Input[type=text]** will only select text fields.
- **Input[type=password]** will only select password fields.
- **Input[type=number]** will only select number fields.

Bordered Inputs

Use the border property to change the border size and color, and use the border-radius property to add rounded corners.

Example

```
input[type=text] {
    border: 2px solid red;
    border-radius: 4px;}
```

Colored Inputs

Use the background-color property to add a background color to the input, and the color property to change the text color.

Example

```
input[type=text] {
  background-color: #3CBC8D;
  color: white;}
```

Animated Search Input

In this example we use the CSS3 transition property to animate the width of the search input when it gets focus. You will learn more about the transition property later, in our CSS3 Transitions chapter.

Example

```
input[type=text] {
    -webkit-transition: width 0.4s ease-in-out;
    transition: width 0.4s ease-in-out;}
input[type=text]:focus {
    width: 100%;}
```

Styling Textareas

Example

```
textarea {
    width: 100%;
    height: 150px;
    padding: 12px 20px;
    box-sizing: border-box;
    border: 2px solid #ccc;
    border-radius: 4px;
    background-color: #f8f8f8;
    resize: none;}
```

Styling Select Menus

Example

```
select {
    width: 100%;
    padding: 16px 20px;
    border: none;
    border-radius: 4px;
    background-color: #f1f1f1;}
```

Styling Input Buttons

Example

```
input[type=button],
input[type=submit],
input[type=reset] {
    background-color: #4CAF50;
    border: none;
    color: white;
    padding: 16px 32px;
    text-decoration: none;
    margin: 4px 2px;
    cursor: pointer;}

/* Tip: use width: 100% for full-width buttons */
```

CSS3 border-radius Property

With CSS3, you can give any element "rounded corners", by using the border-radius property.

Here are three examples:

Example

```
#rcorners1 {
    border-radius: 25px;
    background: #73AD21;
    padding: 20px;
    width: 200px;
    height: 150px;}
#rcorners2 {
    border-radius: 25px;
    border: 2px solid #73AD21;
    padding: 20px;
    width: 200px;
    height: 150px;}
#rcorners3 {
    border-radius: 25px;
    background: url(paper.gif);
    background-position: left top;
    background-repeat: repeat;
    padding: 20px;
    width: 200px;
    height: 150px;}
```

CSS3 border-radius - Specify Each Corner

If you specify only one value for the `border-radius` property, this radius will be applied to all 4 corners. However, you can specify each corner separately if you wish. Here are the rules:

• **Four values**: first value applies to top-left, second value applies to top-right, third value applies to bottom-right, and fourth value applies to bottom-left corner.

• **Three values:** first value applies to top-left, second value applies to top-right and bottom-left, and third value applies to bottom-right

• **Two values:** first value applies to top-left and bottom-right corner, and the second value applies to top-right and bottom-left corner.

• **One value:** all four corners are rounded equally.

Here are three examples:

1. Four values - border-radius: 15px 50px 30px 5px:
2. Three values - border-radius: 15px 50px 30px:
3. Two values - border-radius: 15px 50px:

Here is the code:

Example

```
#rcorners4 {
    border-radius: 15px 50px 30px 5px;
    background: #73AD21;
    padding: 20px;
    width: 200px;
    height: 150px;}
```

```
#rcorners5 {
    border-radius: 15px 50px 30px;
    background: #73AD21;
    padding: 20px;
    width: 200px;
    height: 150px;}
#rcorners6 {
    border-radius: 15px 50px;
    background: #73AD21;
    padding: 20px;
    width: 200px;
    height: 150px;}
```

CSS3 Shadow Effects

With CSS3 you can add shadow to text and to elements. In this chapter you will learn about the following properties.

- **text-shadow**
- **box-shadow**

CSS3 Text Shadow

The CSS3 `text-shadow` property applies shadow to text. In its simplest use, you only specify the horizontal shadow (2px) and the vertical shadow (2px):

Text shadow effect!

Example

```
h1 { text-shadow: 2px 2px;}
h2 { text-shadow: 2px 2px red;}
h3 { color: white;
    text-shadow: 2px 2px 4px #000000;}
```

CSS Buttons

Basic Button Styling

Example

```
.button {
    background-color: #4CAF50; /* Green */
    border: none;
    color: white;
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;}
```

Button Colors

Use the `background-color` property to change the background color of a button.

Example

```
/* Green */
.button1 {background-color: #4CAF50;}
/* Blue */
.button2 {background-color: #008CBA;}
/* Red */
.button3 {background-color: #f44336;}
/* Gray */
.button4 {background-color: #e7e7e7;}
/* Black */
.button5 {background-color: #555555;}
```

Button Sizes

Use the `font-size` property to change the font size of a button.

Example

```
.button1 {font-size: 10px;}
.button2 {font-size: 12px;}
.button3 {font-size: 16px;}
.button4 {font-size: 20px;}
.button5 {font-size: 24px;}
```

Button Padding

Use the `padding` property to change the padding of a button.

Example

```
.button1 {padding: 10px 24px;}
.button2 {padding: 12px 28px;}
.button3 {padding: 14px 40px;}
.button4 {padding: 32px 16px;}
.button5 {padding: 16px;}
```

Rounded Buttons

Use the `border-radius` property to add rounded corners to a button.

Example

```
.button1 {border-radius: 2px;}
.button2 {border-radius: 4px;}
.button3 {border-radius: 8px;}
.button4 {border-radius: 12px;}
.button5 {border-radius: 50%;}
```

Colored Button Borders

Use the border property to add a colored border to a button.

Example

```
.button1 {
    background-color: white;
    color: black;
    border: 2px solid #4CAF50;}/*Green*/
```

Hoverable Buttons

Use the :hover selector to change the style of a button when you move the mouse over it.

Example

```
.button {
    transition-duration: 0.4s;}
.button:hover {
    background-color: #4CAF50;
    color: white;} /* Green */
```

Disabled Buttons

Use the opacity property to add transparency to a button (creates a "disabled" look).

Tip: You can also add the cursor property with a value of "not-allowed", which will display a "no parking sign" when you mouse over the button.

Example

```
.disabled {
    opacity: 0.6;
    cursor: not-allowed;}
```

Button Width

By default, the size of the button is determined by its text content (as wide as its content). Use the width property to change the width of a button.

Example

```
.button1 {width: 250px;}
.button2 {width: 50%;}
.button3 {width: 100%;}
```

Button Groups

Remove margins and add `float:left` to each button to create a button group.

Example

```
.button {
    float: left;}
```

Bordered Button Groups

Use the border property to create a bordered button group.

Example

```
.button {
    float: left;
    border: 1px solid green;}
```

CSS Pagination

Simple Pagination

If you have a website with lots of pages, you may wish to add some sort of pagination to each page. To create a pagination, style a HTML list.

Example

```
ul.pagination {
    display: inline-block;
    padding: 0;
    margin: 0;}
ul.pagination li {display: inline;}
ul.pagination li a {
    color: black;
    float: left;
    padding: 8px 16px;
    text-decoration: none;}
```

Active and Hoverable Pagination

Example

```
ul.pagination li a.active {
    background-color: #4CAF50;
    color: white;}
ul.pagination li a:hover:not(.active)
{background-color: #ddd;}
```

Rounded Active and Hoverable Buttons

Add the border-radius property if you want a rounded "**active**" and "**hover**" button.

Example

```
ul.pagination li a {
    border-radius: 5px;}
ul.pagination li a.active {
    border-radius: 5px;}
```

Hoverable Transition Effect

Add the transition property to the page links to create a transition effect on hover.

Example

```
ul.pagination li a {
    transition: background-color .3s;}
```

Bordered Pagination

Use the border property to add borders to the pagination.

Example

```
ul.pagination li a {
    border: 1px solid #ddd;} /* Gray */
```