

XHTML - Extensible HyperText Markup Language

HTML vs XHTML

Difficult to write elements in XHTML

HTML → ← difficulties of didT and

HTML is much easier to write Consistency is more

(strict syntactic rules)

huge no. of HTML documents

available on the web, thus almost Syntactic correctness

can be checked, either

by a XML browser or

All browsers support it

by an XML validation tool.

difficult to maintain - the code gets

Convert legacy HTML documents → XHTML documents using

SLW tools.

Tidy available at <http://tidy.sourceforge.net>

is one such tool.

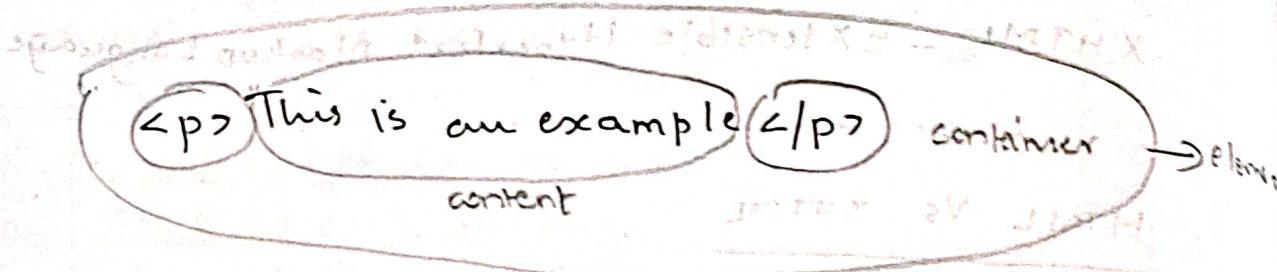
Basic Syntax

→ The fundamental syntactic units of HTML are

called tags. < > → opening tag

< / > → closing tag

`< >` together specify container
content



Comments → increase readability

`<!-- This is a comment -->`

Standard XHTML Document Structure

standard defined

Q&A

xml declaration element

xml declaration element - defines what do all the characters

no need of attribute - version number - 1.0

encoding - encoding used for the document - unicode utf-8 encoding

First line of every XHTML document

`<?xml version = "1.0" encoding = "utf-8"?>`

DOCTYPE command

`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//en"`

`"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">`

→ public identifier specifying the version & type.

→ system identifier which points to the DTD file on the W3C website.

The `<html>` element includes an attribute, `xmlns` that specifies the XHTML namespace.

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

- An XHTML document consists of 2 parts

head and the body

```
<head> </head> The title element is included in the head part
```

```
<body> </body> The body provides the content of the document
```

```
<head>
  <title> My first webpage </title>
```

The content of the title element is displayed by the browser at the top of its display window usually in the browsers window title bar.

Basic text mark up

* Paragraph

```
<p> No pains no gains. This proverb gives a very important lesson in hardwork. If we want
```

to gain anything in our life we have to

work hard. we know "work is worship". Lazy

people cannot gain anything in their lives. so

* Line breaks

Example: *Homework* *Unit 1*

</p>

Johny Johny,
 here I am!

Yes Pappa

Eating sugar?
 smile

No pappa </p>

Preserving whitespace

New

Horizon

College

Engineering

</p></pre>

New

Horizon

College

⇒ pre tag

Engineering

</pre>

</p>

* Headings.

<h1> Indigo Airlines </h1>

<h2> Headquartered in Gurugram </h2>

<h3> Popular airlines in India </h3>

<h4> Founded in 2005 </h4>

<h5> International flight routes </h5>

<h6> Gained reputation for its customer

service </h6>

* Content based tags

- The emphasis tag, `` → specifies that its textual content is ~~padding~~ special and should be displayed in "some way" that indicates this.

Most browsers use italics for such content.

- The strong tag, `` → browsers often set the content of strong elements in bold.

* Horizontal Rules

- The `<hr />` tag causes a line break (ending the current line) and draws a line across the screen.

<hr />

Images

• jpeg

The `` tag of images

↓ 2006 in broadband cards

2 attributes

src → specifies the file containing the image.

alt → specifies text to be displayed when it is not possible to display the image.

alt part contains text part depends on src

src → "stars.jpeg"

"images/stars.jpeg" will be displayed

if no drive not available default forward font

``

→ To Scale the size of the image 2 optional

attributes of img ie, width and height can be included

Hypertext Links

- A link that points to a different document

specifies the address of that document.

- Links are specified in an attribute of an anchor tag `<a>` which is an inline tag.

- The anchor tag that specifies a link is called the source of that link.

- The document whose address is specified in a link is called the target of that link.

- The attribute required for creating links is `href` (hypertext reference).

- The value assigned to `href` specifies the target of the link.

- Links are usually implicitly rendered in a different color than the surrounding text. Sometimes, they are also underlined.

- When the mouse cursor is placed over the

anchor tag content and the left mouse button is pressed, the link is taken by the browser.

If the target is a different document, that document is loaded and displayed, replacing the currently

displayed document. If the target is self in the current document, the document is scrolled by the browser to display the target of the link.

html file → example.html

→ current file → link.html

```
<body>
<a href = "example.html"> Information on Ford </a>
</body>
```

when the link → Information on Ford is clicked, the browser

displays the screen of example.html file

Targets within Documents:

→ The target element can include an id attribute,

which can be used to identify it in an href

attribute. Consider the following example:-

```
<h2 id = "education"> Educational Details </h2>
```

- The value of an id attribute must be unique

within the document.

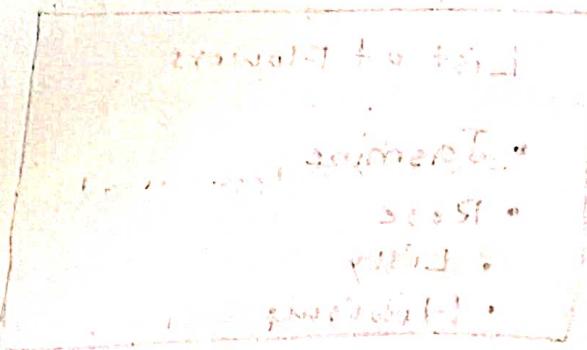
If the target is in the same document as the link, the target is specified in the href attribute value by preceding the id value with a pound sign (#), as in the following example:

```
<a href="#education">Educational Details</a>
```

When the Educational Details link is taken, the browser moves the display so that the `h2` element whose id is education is at the top:

When the target is a part or fragment of another document, the name of the part is specified at the end of the URL, separated by a pound sign (#) as follows:-

```
<a href="example.html #education">Education</a>
```



target of the link

Lists

Unordered Lists

Ordered Lists

Unordered Lists

The `` tag creates an unordered list.

Each item in a list is specified with `` tag

`<head>`

`<title>` Unordered list `</title>`

`</head>`

`<body>`

`<h3>` List of Flowers `</h3>`

``

`` Jasmine ``

`` Rose ``

`` Lilly ``

`` Hibiscus ``

``

`</body>`

`</html>`

Output

List of Flowers

• Jasmine

• Rose

• Lilly

• Hibiscus

<ul type="circle">

 Jasmine

 Rose

 Lilly

• Jasmine

• Rose

• Lilly

Nested Lists

 Solid

 Liquid

 Gas

<ul type="square">

 Wood

 Water

 Oxygen

Ordered Lists

- The ordered list created within the tag

<body>

<h3> List of Fruits </h3>

<ol style="list-style-type: none;">

 Orange

 Apple

 Banana

list of Fruits

1. Orange

2. Apple

3. Banana

I. Solid

- a. Glass
- b. Wax
- c. Copper

II. Liquid

- a. Water
- b. Oil
- c. Blood

III. Gas

- a. Helium
- b. Hydrogen
- c. Nitrogen

<body>

<ol type="I">Solid

<ol type="a">Glass

 Wax

 Copper

first pointLiquid direct to solid

<ol type="a">Water

 Oil

 Blood

 Gas

<ol type="a"> Helium

 Hydrogen

 Nitrogen

explore > Helium

</body>

Tables

Employee Details

Name	Emp-Id	Job Role	Company
Ajith	123	HTML Expert	Thoughtworks
Sonia	313	Programmer	In5nit Technologies
Shubam	202	Android Developer	DxMinds
Kishore	115	Analyst	Honeywell

- A table is specified as the content of the `<table>` tag.
- The most common attribute for the `<table>` tag is `border`.
- A displayed table is preceded by a title, which is given as the content of a `<caption>` tag.
- Each row of a table is specified with a `row` tag `<tr>`.
- `<th>` \Rightarrow table head & `<td>` \Rightarrow table data.

```
<html>
<head>
    <title>Tables </title>
</head>
<body>
    <table border="border">
        <caption style="font-size:large;><strong>
            Employee Details </strong></caption>
```

```
<thead>
```

```
<tr>
```

```
<th> Name </th>
```

```
<th> Emp-Id </th>
```

```
<th> Job Role </th>
```

```
<th> Company </th>
```

Name	Emp-Id	Job Role	Company
Ajith	123	Software Dev.	Thoughtworks
Sonia	313	Programmer	Init Technologies
Shubham	002	Analyst	Wipro

```
<td> Ajith </td>
```

```
<td> 123 </td>
```

```
<td> Html Expert </td>
```

```
<td> Thoughtworks </td>
```

```
</tr>
```

```
<tr>
```

```
<td> Sonia </td>
```

```
<td> 313 </td>
```

```
<td> Programmer </td>
```

```
<td> Init Technologies </td>
```

```
</tr>
```

```
<tr> <td> Shubham </td>
```

```
<td> 002 </td>
```

<td> ~~Roo~~ Android Developer </td>

<td> Domains </td>

</tr>

<tr>

<td> kishore </td>

<td> 115 </td>

<td> Analyst </td>

<td> Honeywell </td>

</tr>

<tbody>

<table>

<body>

<head>

<html>

Levels of css

- Inline css
- Document level css
- External css

Inline css → css applied to single line of element

Document level → css applied to body of the whole document

External → css applied to multiple documents

level

Inline css: It is used to apply css on a single line of element.

eg:- <p style="color:pink"> Welcome css </p>

Document level css / Internal css

It is used to apply css on a single document of page. It can effect all the elements of the page.

It is written inside the style tag within head section of html.

eg:- <html>
 <head>
 <style>
 body { background-color: yellow; }
 </style>
 </head>

 <body>

 <h1> Welcome to css </h1>

</body>
</html>

External CSS:- It is used to apply CSS on multiple pages or all pages. It can be written in any text editor.

The file should not contain any HTML tags. The file must be saved with .css extension.

eg:- style.css

body

{

background-color: blue;

h1 { color: red; font-size: 2em; }

h1

{

color: pink;

background-color: yellow; font-size: 1.5em; width: 50%; height: 50%;

<html>

<head>

<link rel="stylesheet"

type="text/css"

href="style.css">

</head>

<body>

<h1> This is heading </h1>

</body>

</html>

Parameters	HTML	XHTML	Syntactic Differences between HTML and XHTML
Case sensitivity	Tags and attribute names are case insensitive	Tags and attributes names must be in lowercase	
Closing tags	Closing tags may be omitted	All elements must have closing tag	
Quoted attribute values	Special characters are quoted. Numerical values are rarely quoted.	All attribute values must be quoted including numbers	
Explicit attribute values	Some attribute values are implicit for eg: table borders. A default value for border is assumed.	All attribute values must be explicitly stated	
id and name attributes	Both id and name attributes are encouraged.	Use of id is encouraged and use of name is discouraged	
Element nesting	Rules against improper nesting of elements are not enforced.	All nesting rules are strictly enforced.	

Selector Forms

→ Simple Selector forms

→ Class Selectors

→ Generic Selectors (E.g. `h1, h2, h3, h4, h5, h6`)

→ id Selectors

→ Universal Selectors

• Simple selector → Selects the HTML element by name.

```
p {
```

```
    text-align:center;  
    color:blue;
```

```
}
```

All `<p>` elements on the page will be center-aligned

with a blue text color.

• id selector → The id selector selects the HTML element with specific id attribute.

It is written with the (#) character, followed by the id of the element.

```
#para1 {  
    text-align:center;
```

```
    color:blue;  
}
```

The rule will be applied to the HTML element

with `id='para1'`

```
<p id="para1">welcome to less</p>
```

Class Selector - The class selector selects the HTML element with a specific class attribute.

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

```
.class1{  
    text-align:center;  
    color:blue;  
}
```

The rule will be applied to the HTML element with

`class="class1"`

`<p class="class1">This is class selector in css</p>`

Universal Selector :- The universal selector (*)

Selects all the HTML elements on the page

```
* {  
    text-align:center;  
    color:blue;  
}
```

- The rule will be applied to all the HTML elements on the page.

`<p>Welcome to css</p>`

`<p class="class">This is class selector in`

`<p>css Selector forms </p>`

Generic Selectors

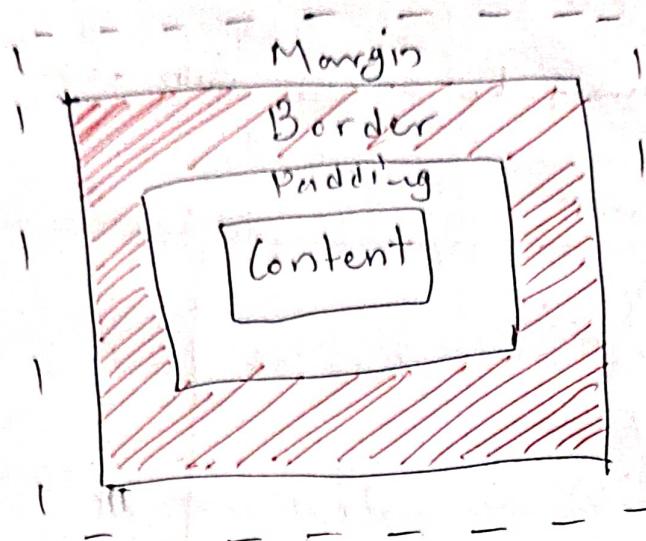
```
• sale {  
    text-align: center;  
    color: blue  
}
```

<h3 class="sale">Weekend Sale </h3>

<p class="sale">25% discount </p>

- If we want to apply styles to the content of more than one kind of tag. This is done by using a generic class which is defined without a tag name in its name. In place of the tag name, you use the name of the generic class, which must begin with a period.

The box Model



Demonstration of the box model.

HTML file

```
<html> <head>
  <title> CSS Box Model </title>
  <link rel="stylesheet" type="text/css" href="abc.css">
<body>
  <h2 class="title"> The CSS Box Model </h2>
  <p> Styling </p>
```

```
</body>
</html>
```

abc.css - CSS file

```
p {
```

```
background-color: skyblue;
display: inline-block;
padding: 50px;
border: 10px solid limegreen;
margin-top: 100px;
```

The Box Model

- The CSS box model is essentially a box that wraps around every HTML element.
- It consists of : content, padding, borders and margins.
- The amount of space between the content of the element and its border is known as padding.
- The space between border and adjacent element is known as margin.
- Content:- The content of the box, where text and images appear.
- Border:- A border that goes around the padding and content.

The `` tag:

We want to apply special font properties to part of a paragraph. The `` tag is designed just for this purpose.

```
<style type="text/css">
  .spanred {
    font-size: 14pt;
    font-family: Arial;
    color: red
  }
<p> You are braver than you think, more
  talented than you know. </p>
```

Background Images:

The `background-image` property is used to place an image in the background of an element.

```
<head><title>Background Images</title>
<style type="text/css">
  body {
    background-image: ;
  }
  p {
    margin-left: 30px;
    margin-right: 30px;
    margin-top: 50px;
    font-size: 14pt;
  }
</style></head>
<body>
  <p> I am a
  </p>
</body>
```

Background Images

- Replication of background image to fill an area is called tiling.
- Tiling can be controlled with the `background-repeat` property, which can take the value repeat (the default), no-repeat, repeat-x or repeat-y.
- The no-repeat value specifies that just one copy of the image is to be displayed. The repeat-x value means the image is to be ~~displayed~~ repeated horizontally.

repeat-y means that is the image is to be repeated vertically.

The <div> tag

→ The <div> tag defines a division or a section in an HTML document. The <div> tag is used as a container for HTML elements which is then styled with CSS. The <div> tag is easily styled by using the class or id attribute.

```
<html>
<head> <style>
    .myDiv {
        border: 5px outset red;
        background-color: lightblue;
        text-align: center;
    }
</style>
</head>
```

```
<body>
    <div class="myDiv">
        <h2> This is a heading in a
            div element </h2>
        <p> This is some text in a
            div element. </p>
    </div>
</body>
</html>
```

Canvas

- The <canvas> element is created with the ID "myCanvas" and a width and height.
- The <script> section contains JavaScript code that gets the 2D rendering context of the canvas using getContext('2d').
- Two shapes are drawn on the canvas: a filled rectangle and a filled circle.

- fillStyle property sets the fill color, and strokeStyle sets the stroke color.

(Write example program)

Video

- The <video> element is used to embed the video.
- The width and height attributes set the dimensions of the video player.
- The controls attribute adds video controls (play, pause, volume, etc.).
- The <source> element inside <video> specifies the video file (name of the file.mpt) and its type

(video/mp4).

→ The text "Your browser does not support the video tag", will be displayed if the browser doesn't support the <video> tag.

Video Attributes

- src: Specifies the URL of the video file.
- type: Specifies the MIME type of the video file.
- controls: Enables video controls.
- width and height: Set the dimensions of the video player.

(Write example program)

Forms

- The HTML `<form>` element is used to create an HTML form for user input.
- The `<form>` element is a container for different elements such as text fields, checkboxes, & radio buttons, submit buttons etc.
- * The `<input>` Element
- The HTML `<input>` element is the most used form element.
- An `<input>` element can be displayed in many ways, depending on the `type` attribute.

Here are some examples:-

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

Write laboratory pgm as example → Forms

The `<label>` element defines a label for many form elements.