



## HTML5 Essentials

HTML is the language that powers the web. With HTML you can create your web pages/website. This course provides an in-depth look at the essentials: the syntax of HTML and best practices for writing and editing your code. Will learn HTML, Document structure, block and inline-level tags, floating images, controlling white space, phrase and font markup, list, links, and tables.

### Topics include

1. [Course Introduction](#)
2. [Introducing HTML](#)
  - 2b. [Introducing HTML5](#)
3. [Basic page structure](#)
4. [Formatting page content](#)
5. [Structuring content](#)
6. [List - Creating Lists](#)
7. [Link - Creating Links](#)
8. [Table - Working with Table](#)
9. [Form - Dealing with Forms](#)
10. [HTML Layout](#)
11. [What's the next step?](#)

## Section 1. Course Introduction

## 1.1. Welcome

Hi All, I'm **Dinanath Jayaswal**, **Senior UI/Web Developer and Adobe Certified Expert Professional**, I wanna welcome you to HTML5 Essentials. This course will take an in-depth look at HTML/HTML5, will learn Syntax and Semantics of HTML. Will also focus on best practices to implement while writing and editing HTML. Here will start by learning what is HTML, Basic HTML Document/Page structure, How to write and format page content, creating a list, link, table, and many more things.

# Section 2. Introducing HTML

## 2.1. Importance

### 3 pillars / core Languages/Technologies used for Front End Web Design/Development:

1. **HTML (HyperText Markup Language)** - Markup language, essential page structure/content, readable and convey structure to the user, text layout model, page mark-up, text, tags, data, details for pages, images, tables, anchor links, forms
  - HTML is an acronym that stands for HyperText Markup Language which is used for creating web pages and web applications
  - HTML is not a programming language; it is a markup language that defines the structure of your content ie. document
2. **CSS (Cascading Style Sheet)** - Style Sheet language, page design / presentation / layouts / styling / formattings, look and feel, creative part of web pages
  - CSS is a stylesheet language that allows you to control the appearance (look and feel) of your webpages
3. **JavaScript (JS)** - Scripting language, dynamic page behavior, logics, conditions and validations, events (mouse click, hover), interactivity with the user/events, dynamic updates in a web page
  - JavaScript is a scripting language of the web that allows you to do interactivity with the user, implement conditions and validations, dynamic updates in a web page, etc.

### Syntax & Example:

```
<html>

<head>
  <title>02.01.intro.html</title>
</head>
```

```
<body>
  <h1>This is Intro HTML file</h1>
  This line will displayed on browser / web page.
</body>

</html>
```

## 2.2. HTML (HyperText Markup Language)

- HTML stands for HyperText Markup Language
- The Standard/foundation/gateway/most widely used language used for developing/creating/structuring content on the web
- We can create a static website by HTML only
- Technically, HTML is a Markup language rather than a programming language
- HTML describes the structure of Web pages using markup
- HTML elements/tags are the building blocks of HTML pages, which represented by <tag>content</tag>
- NO HTML = NO WEB PAGE :)
- HyperText Markup Language, used to MARK the content what exactly it is ie <p> = Paragraph, <header> = Header, <table> = table etc..

## 2.3. What is HTML?

- HTML is an acronym that stands for HyperText Markup Language which is used for creating web pages and web applications.
- HTML is not a programming language; it is a markup language that defines the structure of your content
  - **Hyper Text:**
    - HyperText simply means "Text within Text." A text that has a link within it, is a hypertext. HyperText is a way to link two or more web pages (HTML documents) with each other.
    - Hypertext refers to the way in which/how Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.
  - **Markup language:**
    - A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes the text more interactive and dynamic. It can turn text into images, tables, links, lists, headings, etc.
    - HTML is a Markup Language which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.
  - **Web Page:**

- A web page is a document that is commonly written in HTML and translated by a web browser. A web page can be identified by entering an URL. A Web page can be of the static or dynamic type. With the help of HTML only, we can create static web pages.

## 2.4. Prerequisites of using HTML (HyperText Markup Language)

### Tools for building web sites-web pages/writing HTML

1. **Text Editor/HTML Editor** - NotePad, NotePad++, SublimeText, Atom, Brackets, Coda, Visual Studio Code, DreamWeaver etc.
  - An HTML file (.html / .htm) is a text file itself consists of HTML tags/elements, so to create/modify an HTML file we can use any text editors.
2. **Browsers To view output of .html pages** - Google Chrome, Mozilla Firefox, Internet Explorer, Safari etc.
  - Once the .html/.htm file is created and saved, we can see its output in any latest web browser.



Image - Widely used modern browsers

## 2.5. Basic HTML Document

### Syntax & Example:

```
<!DOCTYPE html>

<html>

  <head>
    <title>02.05.basic.html</title>
  </head>

  <body>
    <h1>Heading: Welcome Welcome to HTML</h1>
    <p>Paragraph: This is my first HTML page</p>
```

```

        <li>Bullet List Item</li> <br />
        <button>Click me</button>
    </body>

</html>

```

## 2.6 Building blocks of HTML (Tag Attribute Element)

An HTML document consist of its basic building blocks:

Name / Item	Description
Tags	An HTML tag surrounds the content and applies meaning to it. It is written between < and > brackets.
Attribute	An attribute in HTML provides extra information about the element, and it is applied within the start tag. An HTML attribute contains two fields: name & value / key & value
Elements	An HTML element is an individual component of an HTML file. In an HTML file, everything written within tags is termed as HTML elements

### Syntax & Example:

```
<tag attribute="attribute value"> content: whole together is an element </tag >
```

### Syntax & Example:

```
<h1 align="right">Heading 1 aligned to right </h1>
```

## 2.7 Tag

- HTML tags are like keywords which defines that how web browser will format and display the content <p>, <h1>, <table>, <strong>
- HTML is a markup language and makes use of various tags to format the content
- Tags are enclosed within angle braces <Tag>
- Except few tags, most of the tags have their corresponding closing tags. <Tag> </Tag> , <Tag />, <p> </p>, <h1> </h1>, <br />, <hr /> <link />
- When a web browser reads an HTML document, browser reads it from top to bottom and left to right
- HTML tags are used to create HTML documents and render their properties

### Syntax & Example:

```
<tag> content </tag >
```

### Syntax & Example:

```

<h1>HTML Tags: Various tags used to develop web page.</h1>
<h3>Heading: Welcome Welcome to HTML</h3>
<p>Paragraph: This is my first HTML page</p>
<hr /> Horizontal Rules
<br /> Line Break
<strong> Bold Strong Text </strong>
<li> list item bulleted list </li>

```

## 2.8 Attribute

- HTML attributes are special words that provide additional information about the elements or attributes are the modifier of the HTML element
- Each element or tag can have attributes, which defines the characteristics/behavior of that element
- Attributes should always be applied with start-tag
- All attributes are made up of two parts – a name and a value, Attributes usually come in name/value pairs like name="value"
- Attribute names and attribute values are case-insensitive

### Syntax & Example:

```
<tag attribute="attribute value"> content </tag >
```

### Syntax & Example:

```

<h1 class="headings" id="heading1"> Heading 1 </h1 >
<p align="right"> Paragraphic text </p>
<hr size="20" width="500"/> Horizontal Rules
<br /> Line Break
<strong style="color:red;"> Bold Strong Text </strong>
<center><h2 title="Title attribute: Heading 2">Heading 2: Hover Me</h2></center>

```

## 2.9 Element

- An HTML file is made of elements
- Elements are responsible for creating web pages and define content in that webpage
- An element in HTML usually consists of a start tag <tagName>, close tag </tagName> and content inserted between them

### Syntax & Example:

```
<tag attribute="attribute value"> content: whole together is an element </tag >
```

### Syntax & Example:

```
<h1 align="right">Heading 1 aligned to right </h1>
```

## 2.10. HTML Syntax

- HTML Syntax is very simple easy to learn
- HTML Contents/Documents ie files are nothing but a simple text files only with .htm or .html extension
- HTML page consists of DTD & TAGS for different visual aspects
- HTML uses TAGS to markup and identify page content
- `<start_tag> Some content here </end_tag>` **Example:** `<p> p is paragraph tag used to create paragraphic text.</p>`
- An HTML Element = `<tag attribute="attribute value"> content: whole together is an element </tag >`
- We can create contents with nested elements like `<tag1><tag2>Some Content </tag2></tag1>` **Example:** `<p><strong>I am bold paragraphic text</strong></p>`

### Types of Attributes

1. **Informative Attribute** - gives extra information about an element. **Example:** `lang="en", class="header", id="container1"`
2. **Functional Attribute** - very important who perform some activities. **Example:** `href="index.html", src="script.js"`

## 2.11. State (Evolution of HTML)

The Web is changing very fast and so HTML

### History of HTML

Year	History
1990	Specification Drafted in Mid 90s by Tim Berners Lee
1991	Tim Berners Lee published document called "HTML Tags"
1995	HTML 2.0 ('standard' HTML)
1997	HTML 3.2 (proprietary browser companies developed many features)
1999	HTML 4.0 W3C Web standards support
2000	XHTML 1.0 (HTML move towards XML)-Extensible Markup Language
2001	2009 - XHTML 2.0

Year	History
2004	WHATWG formed (Web HyperText Application Technology Working Group, they continued with development of HTML 5.0)
2007	2008 - W3C adopts WHATWG's HTML5 Web Application 1.0 specification and drafts
2009	XHTML 2.0 drops/stops
2014	W3C Recommendation: HTML5

## The Current State of HTML

**There are two specifications of HTML: (*Not much changes with syntax*)**

1. W3C (World Wide Web Consortium)
2. WHATWG (Web Hypertext Application Technology Working Group)

## What should you focus on?

There is no good reason not to learn the HTML5 syntax. Focus on learning the elements and syntax first. Gradually learn the new capabilities in HTML5, once you become comfortable with writing clean HTML.

## 2.12. Reference

Some important and valuable references / resources:

- <https://www.w3.org/TR/html53/>
- <https://www.w3.org/TR/2010/WD-html5-20100624/>
- <https://html.spec.whatwg.org/multipage/>
- <https://webplatform.github.io/docs/html/>
- <https://developer.mozilla.org/en-US/>

## 2.13. Editor

### Choosing a Code/HTML/Text Editor

- Any modern Text Editor can work (Not a word editor)



Note: A good code/HTML/text editor can speed up the process of writing and editing code, also highlights syntax errors.

**Code/HTML/Text Editor Features:** The latest text editors must-have features like:

- auto-text
- auto type of code
- Indentation style
- syntax highlighting,
- auto code completion (Intelligent code completion)
- Spell checker
- autosave
- live preview etc.

**text editor VS visual editor (WYSIWYG - (wizzy-wig or What You See Is What You Get))**

The web development world has two types of HTML editing programs:

1. **Text-based HTML editor**

- Text editors are more often used for writing code
- Developer Prefers
- It is better to have little programming to knowledge to type code
- Initially, more coding needed
- **Example:** Adobe Brackets, SublimeText, Atom, Notepad, Notepad++

2. **Visual WYSIWYG, wizzy-wig OR What You See Is What You Get editor**

- Visual editors tend to be used for quick content creation, editing, and styling
- Designer Prefers
- Programming knowledge is not required, anyone can create web content/web pages
- Faster so helps to publish content at high speed
- **Example:** Adobe DreamWeaver, WordPress editor

## Section 2b. Introducing HTML5

HTML5 is the next generation of HTML.

**Prerequisite**

Before learning HTML5, you must have a basic knowledge of HTML, XHTML, CSS.

## 2b.1. What is HTML5?

HTML5 is the latest version of Hypertext Markup Language, the code that describes web pages. HTML5 is a co-operation of W3C (World Wide Web Consortium) and WHATWG (Web Hypertext Application Technology Working Group).

## 2b.2. Why HTML5?

There are many features related to DTD, Tags, Elements, attributes, APIs introduced in HTML:

- It allows you to play a video and audio file
- It facilitates you to design better forms (form 2.0)
- Reduce the usage of the external plugin (Features added in the browser so that no need to use external plugins like SWF player, java applets or so)

### New HTML5 Tags/Elements

- HTML5 Semantic elements: <header>, <footer>, <nav>, <section>, <article>, <aside>
- Graphic elements: <svg>, <canvas>
- Multimedia/Interactive elements: <audio>, <video>

### New HTML5 Form attributes

- New Form 2.0 attributes: number, date, time, calendar, color, file, range

### New HTML5 API's (Application Programming Interfaces)

- HTML Geolocation
- HTML Local Storage
- HTML Application Cache
- HTML Drag and Drop
- HTML Web Workers

## 2b.3. HTML5

### Syntax & Example:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
```

```
<title>HTML5 document</title>
</head>

<body>
  some HTML5 content
</body>

</html>
```

## Section 3. Basic Page Structure

### 3.1. Explore

Let's have a look and observe an HTML source code in the browser:

1. Open any existing .html file in the browser, in our case open `structure.html`
2. Check the page source or Inspect page with Debugger utilities like: FireBug, Inspect/Inspect Element (in browser -> Right Click on page -> choose Inspect Element -> Check/observe actual HTML code/tags).

#### Doctype DTD (Document Type Declaration OR Document Type Definition)

- First line of an HTML document - `!DOCTYPE html`
- It instructs Browsers/User Agents which version of html used for rendering

#### HEAD Section

- All external references in the form of styles, scripts, and third-party plug-ins themes, etc.
- Also consists of additional instructions on how the document will render on different devices
- HEAD section contains non-visible instructions, non-visual elements

#### BODY Section

- All page contents which are visible on the page in the browser
- Body section consists of visible instructions, visual elements
- Example: paragraph, images, list, tables, links, etc.

### 3.2. Doctype DTD (Document Type Declaration OR Document Type Definition)

- Usually the first line of an HTML page, its an instruction to the web browser about what version of HTML the page is written and rules to parse/render.

- Browsers use a DOCTYPE at the beginning of the document to decide whether to handle it in quirks mode (non-standard behavior in the browser) or standards mode (supports all behavior).
- The <!DOCTYPE> declaration is *not case sensitive*.
- **Different Doctype Declarations:**

HTML Version	DTD
HTML5	Document Type Declaration (<!DOCTYPE html>)
XHTML	Document Type Definition (.DTD file) <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
HTML4	Document Type Definition (.DTD file) <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

### 3.3. Head

- <HEAD> section of document contains functional information which is not visible on document.
- The <head> element contains meta information about the document

#### Syntax & Example:

```
<head>

  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>page title here</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" type="text/css" media="screen" href="main.css" />
  <script src="main.js"></script>

</head>
```

### 3.4. Body

<BODY> section/element contains all contents visible in browser window/ on web page.

#### Syntax & Example:

```
<body>
  <h1>Basic HTML Page Structure</h1>
```

Lets have a look and observe different section/parts of HTML page:

```
<ul>
```

```
<li>1. Doctype</li>
<li>2. HEAD</li>
<li>3. BODY</li>
</ul>

</body>
```

## 3.5. Model

### Different Tag/Element category

Tags are divided into different categories based on:

#### 1. Output

##### i. **Block Level elements -**

- Block level element own next line, always shows output on next line
- Block level element starts on a new line, and create a line break before and after the element
- Example Tags: <p>, <div>, <form>, <header>, <nav>, <ul>, <li>, and <h1>

##### ii. **In-line elements -**

- Inline elements are comes in flow of document, content shown in same line
- Inline elements displays output in same line, it only take up the amount of space the content occupies
- Example Tags: <a>, <span>, <strong>, <em>, <u>, <b>, <i> <font>, <center>

#### 2. Closing

##### i. **Container Tags / Paired Tags -**

- Container Tags usually travel in pairs
- A container tag is one which has to be closed
- Example Tags: <html>, <head>, <title>, <body>, <p>, <h1> to <h6>, <strong>

##### ii. **Stand-alone / Unclosed Tags / Empty Tags / Non-Container Tags -**

- A few tags are don't contains any content - they are stand alone.
- Some HTML tags are not closed with end tag, they start and close at the same place
- Example Tags: <br />, <hr />, <link /> <img />

### Understanding Content Models / Tag Content Model

- Content Models / Tag Content Model defines the type of content expected inside an element, and also control syntax rules such as element structure/nesting (DOM)
- Website:
  - <https://www.w3.org/TR/2011/WD-html5-20110525/content-models.html>,
  - [https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Content\\_categories](https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Content_categories)

- **other tag types are:**
  - i. phrasing content : <canvas>, <code>, <data>, <label>, <mark>
  - ii. interactive content : <a>, <button>, <select>, <textarea>, <audio>, <video>, <menu>
  - iii. metadata content : <title>, <link>, <script>, <noscript>, <meta>
  - iv. sectioning content : <header>, <footer>, <nav>, <section>, <article>, <aside>
  - v. heading content : <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <hgroup>
  - vi. embedded content : <audio>, <canvas>, <embed>, <img>, <video>, <math>
  - vii. flow content : <address>, <article>, <aside>, <table>, <template>

## Section 4. Formatting Page Content

### 4.1. Format

- HTML Formatting is a process of formatting text for better look and feel, visibility
- Formatting elements were designed to display special types of text without using CSS

Tag	Use
<strong>	Bold Important text
<b>	Bold text
<em>	Emphasized text
<i>	Italic text
<u>	Underline text
<mark>	marked or highlighted text:
<sub>	Subscript text
<sup>	Superscript text
<big>	increase font size by one conventional unit
<small>	Small text
<del>	Deleted text
<tt>	monospaced font

### 4.2. Headings

- Headings are used to structure pages and determine content hierarchy.

- Headings/Heading Sectioning Elements are defined with the <h1> to <h6> tags.
- <h1> defines the most important/largest/biggest heading. <h6> defines the least important/smallest heading.

### 4.3. Paragraphs

- The HTML <p> element represent/defines a paragraph
- Paragraphs are block-level elements (always goes to next line), and will automatically close if another block-level element is parsed before the closing </p> tag.
- Paragraphs are usually represented in visual media as blocks of text separated from adjacent blocks by blank lines and/or the first-line indentation

### 4.4. Breaks

- We use <br /> in case we want a line break (a new line) without starting a new paragraph
- Sometimes we need to have Hard-Return in our content (poems, quotes) without starting a new paragraph, there line break <br /> is useful.
- <br /> is a standalone tag, which means we need to start and close at the same place like: <br />

### 4.5. Emphasizing

- <b> bold tag simply means bold text without any extra importance just a presentation
- <strong> tag have meaning like strongly emphasizing Bold Important text
- <i> element defines the italic text, without any extra importance
- <em> element defines the emphasized text, with added semantic importance
- For Screen Readers or Accessibility purpose <strong> & <em> semantic tags are playing an important role

### 4.6. Special Characters / HTML Entity (Named character entity)

- HTML Entity represents special characters like &, copyright sign, a registered sign
- Entities are frequently used to display reserved characters
- Also, we can use Entities in place of other characters that are not available on the keyboard/difficult to type with a standard keyboard
- Entity always start with & ampersand sign and ends with; semicolon (**Example: <**)
- <https://dev.w3.org/html5/html-author/charref>

Note: Browsers may not support all entity names

String/Character	Description / Note	Entity Name	Entity Number
------------------	--------------------	-------------	---------------

String/Character	Description / Note	Entity Name	Entity Number
&	and OR ampersign OR ampersand	&amp;	&#38;
	non-breaking space	&nbsp;	&#160;
<	less than	&lt;	&#60;
>	greater than	&gt;	&#62;
<	left-pointing angle bracket	&lang;	&#9001;
>	right-pointing angle bracket	&rang;	&#9002;
€	euro	&euro;	&#8364;
£	pound	&pound;	&#163;
©	copyright	&copy;	&#169;
®	registered trademark	&reg;	&#174;
"	double quotation mark	&quot;	&#34;
TM	Trade Mark sign	&trade;	&#8482;

## 4.6. White spaces

- By default, browsers ignore the white spaces inserted in content after 1st space
- &nbsp; non-breaking space entity is used to insert extra white space

Note: Default only 1 space is identified/rendered by the browser **Note:** Many browsers blocks &nbsp; entity considering junk/spam, advisable to use CSS and apply margin/padding accordingly

## 4.8. Images

**There are many formats of photos/pictures/images/Web Graphics used on the web**

- BMP (Bitmap)
- GIF (Graphics Interchange Format)
- JPEG / JPG (Joint Photographic Experts Group)
- PNG (Portable Network Graphic)
- SVG (Scalable Vector Graphics)

**img tag**



- Images are one of the most important and integral parts of a web page used to improve the design and the appearance of a web page
- In HTML, images are embedded/defined/imported/placed with the `<img>` tag.
- The `src` attribute is required, it specifies/contains the path (the URL/web address) of the image you want to embed
- The `alt` attribute contains/provides/describe the image a textual description (alternate text) of the image, isn't mandatory but is incredibly useful for accessibility (screen readers)
- The `<img>` tag is a standalone/empty tag, it contains attributes only, and does not have a closing tag.

#### 4.9. Font (Not Supported in HTML5)

- Fonts play a very important role in making a website more user-friendly and increasing content readability
- HTML `<font>` tag is used to define the font style for the text contained within it
- The font tag is having three attributes called `face`, `size`, and `color`
- The `<font>` tag defines/specifies the font face, font size, and color of the text

## Section 5. Structuring Content

### 5.1. Structure

- HTML allows us to structure the document as per the meaning of the content
- Div, Table tags are used to create structure, layout but its not semantic
- HTML Semantic Tags - Structure of the webpage to show meaning, not just layout
- HTML Semantic Structural/Sectioning Elements helps to create pages more readable, accessible, better search engine results-oriented, easy to modify/update
- A semantic element clearly describes its meaning to both the browser and the developer

Note: HTML5 semantic elements are supported in all modern browsers.

#### HTML Heading Sectioning Elements

h1 to h6 (total 6 headings are available)

#### HTML5 Semantic Structural/Sectioning Elements

Tag	Use
<code>&lt;main&gt;</code>	Specifies the main content of a document
<code>&lt;header&gt;</code>	Specifies a header for a document or section

Tag	Use
<code>&lt;nav&gt;</code>	Defines navigation links
<code>&lt;section&gt;</code>	Defines a section in a document
<code>&lt;article&gt;</code>	Defines an article
<code>&lt;aside&gt;</code>	Defines content aside from the page content
<code>&lt;footer&gt;</code>	Defines a footer for a document or section

### Why HTML5 Semantic Structural/Sectioning Elements introduced?

- Earlier with HTML, developers used/created their own id/class names to style elements like header, footer, top nav, bottom nav, main menu, nav - navigation, main left right top bottom container, content, left article, right-sidebar, etc.
- This made it difficult and impossible for search engines to identify the correct web page content also a developer to the browser and edit content
- With the new HTML5 semantic/meaningful elements like (`<header>` `<footer>` `<nav>` `<section>` `<article>` `<aside>`), this will become easier
- HTML5 semantic/meaningful elements make pages more readable for machines as well users with accessibility, search engine results, more consistent and easier to use, and style
- According to the W3C, a Semantic Web: "Allows data to be shared and reused across applications, enterprises, and communities."

### Implementing structural hierarchy

It's really up to you what exactly the elements involved represent, as long as the hierarchy makes sense. You just need to bear in mind a few best practices as you create such structures:

- Preferably you should just use a single `<h1>` per page — this is the top-level heading, and all others sit below this in the hierarchy.
- Make sure you use the headings in the correct order in the hierarchy. Don't use `<h3>`s to represent subheadings, followed by `<h2>`s to represent subheadings — that doesn't make sense and will lead to weird results.

### Why do we need structure?

- Users looking at a web page tend to scan quickly to find relevant content. If they can't see anything useful within a few seconds, they'll likely get frustrated and go somewhere else.
- Search engines indexing your page consider the contents of headings as important keywords for influencing the page's search rankings. Without headings, your page will perform poorly in terms of SEO (Search Engine Optimization).

- Visually impaired people often don't read web pages; they listen to them instead. If headings are not available, they will be forced to listen to the whole document read out loud.

## 5.2. Div

- The div tag is known as Division, Divider tag
- The HTML Content Division element (<div>) is the generic container for flow content
- Before HTML5 Semantic Tags, the Div tag is used in HTML to make divisions of content in the web page like (text, images, header, footer, navigation bar, etc)
- The <div> tag is an empty container tag, which defines a division or a section by specifying child elements
- With HTML5 Semantic Tags, Div a powerful tag also used for structuring, dividing, Layout, Sectioning page/content with CSS or manipulated with scripts
- The <div> tag is a block-level element, so a line break is placed before and after it
- The Div is the most usable tag in web development because it helps us to separate data in the web page and we can create a particular section
- It is used to the group of various tags of HTML so that sections can be created and style can be applied to them

## 5.3. Outlines

- HTML uses Semantic Elements, Headings, Semantic Tags to generate/describe the Document Outline of page content (Document Outline = Topics, Table of Contents, Index)
- Document Outline is used by devices/browsers to scan & search content, navigate to a particular/specific section, also to determine how contents relate to each other
- HTML5 Outliner is used to understand & know the exact outline ie page structure of the HTML page/document
- Use outliner utility: <https://gsnedders.html5.org/outliner/>
- Text Editor like Brackets uses a Document Outliner Plugin/extension to show proper document outline/table of contents/topics/Indexing of page
- Brackets -> View menu -> Show Document Outline

# Section 6. Creating Lists

- HTML lists are used to specify lists of information (bulleted/unordered or numbered/ordered content)
- HTML list is an incredible/useful way to represent organized content

**HTML offers three types/ways for specifying lists of information. All lists must contain one or more list elements**

<b>Tag</b>	<b>Meaning / Use / Description</b>
<code>&lt;ul&gt;</code>	An Unordered List or bulleted list. List items using plain bullets
<code>&lt;ol&gt;</code>	An Ordered List or numbered list. Uses different schemes of numbers to list your items
<code>&lt;dl&gt;</code>	A Description List or Definition List. This arranges your items in the same way as they are arranged in a dictionary, header item, and description

## **HTML List Tags**

<b>Tag</b>	<b>Meaning / Use / Description</b>
<code>&lt;ul&gt;</code>	Defines an Unordered List or bulleted list
<code>&lt;ol&gt;</code>	Ordered List or numbered list
<code>&lt;li&gt;</code>	Defines List Item
<code>&lt;dl&gt;</code>	Defines Description List or Definition List
<code>&lt;dt&gt;</code>	Defines a term/item/heading in a description list
<code>&lt;dd&gt;</code>	Describes the term in a description list

### **6.1. Unordered List or Bulleted List**

- An Unordered List is a collection of related items that have no special order or sequence
- Unordered List is created by using HTML `<ul>` tag. Each item in the list `<li>` is marked with a bullet/rendered as a bulleted list
- `<li>` List Item, type attribute helps to specify the type of bullet `disc/circle/square/none`

### **6.2. Ordered List or Numbered List**

- The HTML `<ol>` element represents an ordered list of items, typically rendered as a numbered list
- Ordered List is created by using HTML `<ol>` tag. Each item in the list `<li>` is marked with a number or an order in ascending/descending
- `<li>` List Item, type attribute helps to specify the type of order `1/A/a/I/i`

### **6.3. Description List or Definition List**

- Description List or Definition List represents entries list like in a dictionary or encyclopedia

- Description List or Definition List is the ideal way to present a glossary, list of terms, or another name/value list
- A description list is a list of terms, with a description of each term
- The <dl> tag defines the description list, the <dt> tag defines the term (name), and the <dd> tag describes each term

## Section 7. Creating Links

- A webpage can contain various links/hyperlinks that take you directly to other pages and even specific parts of a given page
- Hyperlinks allow visitors to navigate between Web sites by clicking on words, phrases, and images.

### 7.1. Anchor <a>

- A link is specified using HTML tag <a>, also is called anchor tag
- The HTML <a> (anchor element) creates a hyperlink to other web pages, files, locations within the same page, email addresses, or any other URL.
- The href attribute is the most important attribute which specifies links to destination page or URL href="url path"
- target attribute Specifies where to display the linked URL. target="\_blank"
- title attribute makes the anchor links more accessible, semantic, meaningful, title="Click to visit"
- rel attribute describes relationship of current linked object and target object, rel="next/download/help/author"

#### Syntax & Example:

```
<a href="url-path" title="accessible" target="_blank"
rel="next/prev/help/download">Content</a>
```

#### Syntax & Example:

```
<a href="http://www.google.com" title="Click to visit Google">Visit Google.com</a>
```

### 7.2 internal external links

#### Syntax & Example:

```
<a href="../07.01.link.anchor.html" title="visit links" >Links example 1</a> <br/>
<br/>
```

```
<a href="../08. Working with Table/08.01.table.basic.html" title="Click to visit
Table" >Table example 1</a> <br/>
```

### 7.3 download links

- Ability to download some file or resource

#### Syntax & Example:

```
<a href="help.pdf" title="download help pdf" download>Download Help PDF</a>
```

replace the name of the file while downloading

```
<a href="_help123.pdf" title="download help pdf" download="html_help">text</a>
```

in this scenario, "\_help123.pdf" will be renamed with "html\_help.pdf"

## 7.4 fragment identifiers / anchor #links

- #Anchors helps to jump to a specific section of current (within)/other linked URL/file
- Need to Define and linked to the ID of the specific element

# Section 8. Working with Table

- The HTML tables allow arranging data like text, images, links, into rows and columns of cells
- In HTML <table> tag is used structure data in tabular form
- HTML table tag display data in tabular form (row \* column). There can be many columns in a row
- We can create a table to display data in tabular form, using <table> element, with the help of <tr> , <td>, and <th> elements
- Table row is defined by <tr> tag, the table header is defined by <th>, and table data cells is defined by <td>
- HTML tables are used to manage the layout of the page e.g. header section, navigation bar, body content, footer section, etc.

Note: But it is recommended to use a div tag over the table to manage the layout of the page

#### HTML Table Tags

Tag	Meaning / Use / Description
<table>	Defines a table
<tr>	Defines a table row
<td>	Defines a table cell
<th>	Defines a header in table
<tbody>	Group the body content in a table

Tag	Meaning / Use / Description
<thead>	Group the header content in a table
<tfoot>	Group the footer content in a table

## 8.1. Creating Basic Table

### Syntax & Example:

```
<table>

  <tr>
    <td>Sr.</td>
    <td>Name</td>
    <td>Address</td>
    <td>Contact Number</td>
  </tr>

  <tr>
    <td>1.</td>
    <td>Dinanath</td>
    <td>India</td>
    <td>91-1234567890</td>
  </tr>

  <tr>
    <td>2.</td>
    <td>Dino</td>
    <td>USA</td>
    <td>1-0987654321</td>
  </tr>

  <tr>
    <td>3.</td>
    <td>Ambar</td>
    <td>UK</td>
    <td>44-01234567890</td>
  </tr>

  <tr>
    <td>4.</td>
    <td>Anthony</td>
    <td>India</td>
    <td>91-0223344556</td>
  </tr>

</table>
```

## Section 9. Dealing with Forms

- HTML forms are a very powerful tool for interacting with user/user data
- HTML Forms are required, when you want to collect some data from the site visitor
- HTML Forms are one of the main points of interaction between a user and a website or application. They allow users to send data to the website
- An HTML form is a section of a document that contains controls such as text fields, password fields, checkboxes, radio buttons, submit button, etc.
- An HTML form facilitates the user to enter data that is to be sent to the server for processing such as name, email address, password, phone number, etc.
- The <form> tag is used to create an HTML form to take input from a user
- The <input> element is a fundamental form element that creates different form components with the type attribute

## 9.1. Creating Basic Form

### Syntax & Example:

```
<form name="enrollmentForm" action="#" method="post">

    <table border="1" width="800" cellspacing="5" cellpadding="5">

        <th colspan="2"><h1>Enrollment Form </h1></th>

        <tr>
            <td>First Name:</td>
            <td><Input type="text" name="firstNameText" id="firstNameText1" /> </td>
        </tr>

        <tr>
            <td>Last Name:</td>
            <td><Input type="text" name="lastNameText" id="lastNameText1" /></td>
        </tr>

        <tr>
            <td>Password:</td>
            <td><Input type="password" name="password" id="password1" /></td>
        </tr>

        <tr>
            <td>Gender:</td>
            <td>
                <input type="radio" name="gender" id="male" value="male" /> Male
                <input type="radio" name="gender" id="female" value="female" />
                <input type="radio" name="gender" id="other" value="other" /> Other
            </td>
        </tr>

        <tr>
            <td>Hobby:</td>
            <td>
```

Female



```

        <input type="checkbox" name="reading" id="reading" value="reading" />
Reading
        <input type="checkbox" name="music" id="music" value="music" /> Music
        <input type="checkbox" name="dance" id="dance" value="dance" /> Dance
        <input type="checkbox" name="sports" id="sports" value="sports" />
Sports
    </td>
</tr>

<tr>
    <td>Country:</td>
    <td>
        <select name="country" id="country">
            <option value="india">India</option>
            <option value="usa">USA</option>
            <option value="uk">UK</option>
            <option value="nz">NZ</option>
        </select>
    </td>
</tr>

<tr>
    <td>Comments:</td>
    <td>
        <textarea cols="40" rows="5">Enter Comments</textarea>
    </td>
</tr>

<tr>
    <td colspan="2">
        <center>
            <input type="submit" name="submit" value="Submit">
            <input type="reset" name="reset" value="Reset" />
        </center>
    </td>
</tr>

</table>

</form>

```