

Introduction

HTML

What is HTML?

HTML is a **markup** language for **describing** web documents (web pages).

HTML stands for **H**yper **T**ext **M**arkup **L**anguage

A markup language is a set of **markup tags**

HTML documents are described by **HTML tags**

Each HTML tag **describes** different document content

HTML Example

```
► <!DOCTYPE html>
  <html>
    <head>
      <title>Page Title</title>
    </head>
    <body>

      <h1>My First Heading</h1>
      <p>My first paragraph.</p>

    </body>
  </html>
```

- The **<!DOCTYPE html>** declaration defines this document to be HTML5
- The text between **<html>** and **</html>** describes an HTML document
- The text between **<head>** and **</head>** provides information about the document
- The text between **<title>** and **</title>** provides a title for the document
- The text between **<body>** and **</body>** describes the visible page content
- The text between **<h1>** and **</h1>** describes a heading
- The text between **<p>** and **</p>** describes a paragraph

HTML Elements

- ▶ HTML Tags Starts With “<” and Ends with “>” . The Text between this two Brackets called As HTML Tags.
- ▶ We have Two types of Tags in HTML . That is Starting Tag and Closing Tag.
- ▶ Closing tag will starts with “</” and ends with “>”
- ▶ Example: `<h1>Hello World</h1>`

HTML Page Structure

```
<html>  
  
  <head>  
    <title>Page title</title>  
  </head>  
  
  <body>  
    <h1>This is a heading</h1>  
    <p>This is a paragraph.</p>  
    <p>This is another paragraph.</p>  
  </body>  
  
</html>
```

The <!DOCTYPE> Declaration

Common Declarations

HTML5

```
<!DOCTYPE html>
```

HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

XHTML 1.0

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

HTML Basics

- ▶ HTML Headings .. `<h1>`,`<h2>`....`<h6>`
- ▶ HTML Paragraphs: `<p></p>`
- ▶ HTML Links : ``
- ▶ HTML Images ``
- ▶ HTML Line Break `
`

HTML Attributes

- ▶ Attributes provide additional information about HTML elements.
- ▶ HTML elements can have **attributes**
- ▶ Attributes are always specified in **the start tag**
- ▶ Attributes come in name/value pairs like: **name="value"**

HTML Attributes

- ▶ The title Attribute
- ▶ The href Attribute
- ▶ Size Attributes
- ▶ The alt Attribute

The HTML Style Attribute

- ▶ `<h1 style="color:blue">This is a heading</h1>`
`<p style="color:red">This is a paragraph.</p>`
- ▶ **HTML Text Size**
- ▶ `<h1 style="font-size:300%">This is a heading</h1>`
`<p style="font-size:160%">This is a paragraph.</p>`
- ▶ Use the **style** attribute for styling HTML elements
- ▶ Use **background-color** for background color
- ▶ Use **color** for text colors
- ▶ Use **font-family** for text fonts
- ▶ Use **font-size** for text sizes
- ▶ Use **text-align** for text alignment

HTML Text Formatting Elements

HTML Text Formatting Elements

Tag	Description
<u></u>	Defines bold text
<u></u>	Defines emphasized text
<u><i></u>	Defines italic text
<u><small></u>	Defines smaller text
<u></u>	Defines important text
<u><sub></u>	Defines subscripted text
<u><sup></u>	Defines superscripted text
<u><ins></u>	Defines inserted text
<u></u>	Defines deleted text
<u><mark></u>	Defines marked/highlighted text

HTML Comments

- ▶ Comment tags `<!--` and `-->` are used to insert comments in HTML.
- ▶ `<!-- Write your comments here -->`

HTML Styles - CSS

- ▶ Styling HTML with CSS
- ▶ CSS stands for Cascading Style Sheets
- ▶ Styling can be added to HTML elements in 3 ways:
- ▶ Inline - using a **style attribute** in HTML elements
- ▶ Internal - using a **<style> element** in the HTML <head> section
- ▶ External - using one or more **external CSS files**

CSS Syntax

- ▶ CSS styling has the following syntax:
- ▶ *element { property:value; property:value }*
- ▶ Use the HTML **style** attribute for inline styling
- ▶ Use the HTML **<style>** element to define internal CSS
- ▶ Use the HTML **<link>** element to refer to an external CSS file
- ▶ Use the HTML **<head>** element to store **<style>** and **<link>** elements
- ▶ Use the CSS **color** property for text colors
- ▶ Use the CSS **font-family** property for text fonts
- ▶ Use the CSS **font-size** property for text sizes
- ▶ Use the CSS **border** property for visible element borders
- ▶ Use the CSS **padding** property for space inside the border
- ▶ Use the CSS **margin** property for space outside the border

HTML Links

- ▶ HTML links are hyperlinks.
- ▶ You can click on a link and jump to another document.
- ▶ **Note:** A link does not have to be text. It can be an image or any other HTML element.
- ▶ **HTML Links - Syntax**
- ▶ In HTML, links are defined with the `<a>` tag:
- ▶ `link text`
- ▶ `Visit our HTML tutorial`
- ▶ The **href** attribute specifies the destination address (`http://www.w3schools.com/html/`) of the link.
- ▶ The **link text** is the visible part (Visit our HTML tutorial).
- ▶ Clicking on the link text will send you to the specified address.

HTML Links - The target Attribute

- ▶ The **target** attribute specifies where to open the linked document.
- ▶ The target attribute can have one of the following values:
- ▶ `_blank` - Opens the linked document in a new window or tab
- ▶ `_self` - Opens the linked document in the same window/tab as it was clicked (this is default)
- ▶ `_parent` - Opens the linked document in the parent frame
- ▶ `_top` - Opens the linked document in the full body of the window
- ▶ `framename` - Opens the linked document in a named frame
- ▶ This example will open the linked document in a new browser window/tab:
- ▶ Example:
- ▶ `Visit W3Schools!`

HTML Tables

- ▶ Tables are defined with the `<table>` tag.
- ▶ Tables are divided into **table rows** with the `<tr>` tag.
- ▶ Table rows are divided into **table data** with the `<td>` tag.
- ▶ A table row can also be divided into **table headings** with the `<th>` tag.
- ▶ `<table border="1" style="width:100%">`

- ▶ Example:

```
<tr>
  <td>Jill</td>
  <td>Smith</td>
  <td>50</td>
</tr>
<tr>
  <td>Eve</td>
  <td>Jackson</td>
  <td>94</td>
</tr>
</table>
```

- ▶ Use the HTML **<table>** element to define a table
- ▶ Use the HTML **<tr>** element to define a table row
- ▶ Use the HTML **<td>** element to define a table data
- ▶ Use the HTML **<th>** element to define a table heading
- ▶ Use the HTML **<caption>** element to define a table caption
- ▶ Use the CSS **border** property to define a border
- ▶ Use the CSS **border-collapse** property to collapse cell borders
- ▶ Use the CSS **padding** property to add padding to cells
- ▶ Use the CSS **text-align** property to align cell text
- ▶ Use the CSS **border-spacing** property to set the spacing between cells
- ▶ Use the **colspan** attribute to make a cell span many columns
- ▶ Use the **rowspan** attribute to make a cell span many rows
- ▶ Use the **id** attribute to uniquely define one table

HTML Lists

HTML can have Unordered lists, Ordered lists, or Description lists:

Unordered List

- The first item
- The second item
- The third item
- The fourth item

Ordered List

1. The first item
2. The second item
3. The third item
4. The fourth item

Description List

The first item
Description of item

The second item
Description of item

Unordered HTML Lists - The Style Attribute

A **style** attribute can be added to an **unordered list**, to define the style of the marker:

Style	Description
list-style-type:disc	The list items will be marked with bullets (default)
list-style-type:circle	The list items will be marked with circles
list-style-type:square	The list items will be marked with squares
list-style-type:none	The list items will not be marked

Unordered HTML List

- ▶ **Unordered HTML List - Choose List Item Marker**
- ▶ The CSS **list-style-type** property is used to define the style of the list item marker:

Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

Ordered HTML Lists - The Type Attribute

A **type** attribute can be added to an **ordered list**, to define the type of the marker:

Type	Description
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

HTML Description Lists & Nested HTML Lists

- ▶ HTML also supports description lists.
- ▶ 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
- ▶ **Nested HTML Lists**
- ▶ List can be nested (lists inside lists):

Horizontal Lists

- ▶ HTML lists can be styled in many different ways with CSS.
- ▶ One popular way is to style a list horizontally, to create a menu:
- ▶ Example : [Horizontal menu](#)

HTML Block Elements and Inline Elements

- ▶ Most HTML elements are defined as **block level** elements or **inline** elements.
 - ▶ Block level elements normally start (and end) with a new line, when displayed in a browser.
 - ▶ Examples: `<h1>`, `<p>`, ``, `<table>`
-
- ▶ An inline element does not start on a new line and only takes up as much width as necessary.
 - ▶ Examples: ``, `<td>`, `<a>`, ``

The `` Element

- ▶ The `` element is often used as a container for some text.
- ▶ The `` element has no required attributes, but both **style** and **class** are common.
- ▶ When used together with CSS, the `` element can be used to style parts of the text:

The <div> Element

- ▶ The <div> element is often used as a container for other HTML elements.
- ▶ The <div> element has no required attributes, but both **style** and **class** are common.
- ▶ When used together with CSS, the <div> element can be used to style blocks of content:

HTML The class Attribute

- ▶ The HTML class attribute makes it possible to define equal styles for elements with the same class name.
- ▶ Let us see an Example with three `<div>` elements that points to the same class name:
- ▶ **Using The class Attribute on Inline Elements**
- ▶ The HTML class attribute can also be used for inline elements:

HTML Layouts



HTML Layouts

Website Layout Using HTML5

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

<code><header></code>	header	Defines a header for a document or a section
<code><nav></code>	nav	Defines a container for navigation links
<code><section></code>	section	Defines a section in a document
<code><article></code>	article	Defines an independent self-contained article
<code><aside></code>	aside	Defines content aside from the content (like a sidebar)
<code><footer></code>	footer	Defines a footer for a document or a section
	details	Defines additional details
	summary	Defines a heading for the details element

This example uses `<header>`, `<nav>`, `<section>`, and `<footer>` to create a multiple column layout:

HTML Responsive Web Design

- ▶ Responsive Web Design makes your web page look good on all devices (desktops, tablets, and phones).
- ▶ Responsive Web Design is about using CSS and HTML to resize, hide, shrink, enlarge, or move the content to make it look good on any screen:

HTML Forms

- ▶ The **<form>** Element
- ▶ HTML forms are used to collect user input.
- ▶ The **<form>** element defines an HTML form:
- ▶ `<form>`

form elements

`</form>`

- ▶ HTML forms contain **form elements**.
- ▶ Form elements are different types of input elements, checkboxes, radio buttons, submit buttons, and more.

The `<input>` Element

- ▶ The `<input>` element is the most important **form element**.
- ▶ The `<input>` element has many variations, depending on the **type** attribute.
- ▶ `<input type="text">` defines a one-line input field for **text input**:
- ▶ `<input type="radio">` defines a **radio button**.
- ▶ `<input type="submit">` defines a button for **submitting** a form to a **form-handler**.
- ▶ The form-handler is typically a server page with a script for processing input data.

Action attribute

- ▶ The **action attribute** defines the action to be performed when the form is submitted.
- ▶ The common way to submit a form to a server, is by using a submit button.
- ▶ Normally, the form is submitted to a web page on a web server.
- ▶ `<form action="action_page.php">`
- ▶ If the action attribute is omitted, the action is set to the current page.

The Method Attribute

- ▶ The **method attribute** specifies the HTTP method (**GET** or **POST**) to be used when submitting the forms:
- ▶ `<form action="action_page.php" method="get">`
- ▶ `<form action="action_page.php" method="post">`

When to Use GET?

- ▶ You can use GET (the default method):
- ▶ If the form submission is passive (like a search engine query), and without sensitive information.
- ▶ When you use GET, the form data will be visible in the page address:
- ▶ `action_page.php?firstname=Mickey&lastname=Mouse`
- ▶ GET is best suited to short amounts of data. Size limitations are set in your browser.

When to Use POST?

- ▶ You should use POST:
- ▶ If the form is updating data, or includes sensitive information (password).
- ▶ POST offers better security because the submitted data is not visible in the page address.

The Name Attribute

- ▶ To be submitted correctly, each input field must have a name attribute.
- ▶ This example will only submit the "Last name" input field:
- ▶

```
<form action="action_page.php">  
  First name:<br>  
  <input type="text" value="Mickey"><br>  
  Last name:<br>  
  <input type="text" name="lastname" value="Mouse"><br><br>  
  <input type="submit" value="Submit">  
</form>
```

Grouping Form Data with <fieldset>

- ▶ The <fieldset> element groups related data in a form.
- ▶ The <legend> element defines a caption for the <fieldset> element.
- ▶

```
<form action="action_page.php">  
  <fieldset>  
    <legend>Personal information:</legend>  
    First name:<br>  
    <input type="text" name="firstname" value="Mickey"><br>  
    Last name:<br>  
    <input type="text" name="lastname" value="Mouse"><br><br>  
    <input type="submit" value="Submit">  
  </fieldset>  
</form>
```
- ▶ This is how the HTML code above will be displayed in a browser:

Personal information:

First name:

Mickey

Last name:

Mouse

Submit

Here is the list of <form> attributes:

Attribute	Description
accept-charset	Specifies the charset used in the submitted form (default: the page charset).
action	Specifies an address (url) where to submit the form (default: the submitting page).
autocomplete	Specifies if the browser should autocomplete the form (default: on).
enctype	Specifies the encoding of the submitted data (default: is url-encoded).
method	Specifies the HTTP method used when submitting the form (default: GET).
name	Specifies a name used to identify the form (for DOM usage: document.forms.name).
novalidate	Specifies that the browser should not validate the form.
target	Specifies the target of the address in the action attribute (default: _self).

You will learn more about attributes in the next chapters.

HTML Form Elements

- ▶ The `<select>` Element (Drop-Down List)

- ▶

```
<select name="cars">  
  <option value="volvo">Volvo</option>  
  <option value="saab">Saab</option>  
  <option value="fiat">Fiat</option>  
  <option value="audi">Audi</option>  
</select>
```

- ▶ The `<textarea>` Element

- ▶ The `<textarea>` element defines a multi-line input field (a text area):

- ▶

```
<textarea name="message" rows="10" cols="30">  
The cat was playing in the garden.  
</textarea>
```

HTML Input Attributes

- ▶ **The value Attribute**

- ▶ The **value** attribute specifies the initial value for an input field

- ▶ **The readonly Attribute**

- ▶ The **readonly** attribute specifies that the input field is read only (cannot be changed)

- ▶ **The disabled Attribute**

- ▶ The **disabled** attribute specifies that the input field is disabled.
- ▶ A disabled element is un-usable and un-clickable.
- ▶ Disabled elements will not be submitted

- ▶ **The size Attribute**

- ▶ The **size** attribute specifies the size (in characters) for the input field:

- ▶ **The maxlength Attribute**

- ▶ The **maxlength** attribute specifies the maximum allowed length for the input field:

HTML5 Introduction

- ▶ **New HTML5 Elements**
- ▶ The most interesting new elements are:
- ▶ New **semantic** elements like `<header>`, `<footer>`, `<article>`, and `<section>`.
- ▶ New form **control attributes** like `number`, `date`, `time`, `calendar`, and `range`.
- ▶ New **graphic** elements: `<svg>` and `<canvas>`.
- ▶ New **multimedia** elements: `<audio>` and `<video>`.

What are Semantic Elements?

- ▶ A semantic element clearly describes its meaning to both the browser and the developer.
- ▶ Examples of **non-semantic** elements: `<div>` and `` - Tells nothing about its content.
- ▶ Examples of **semantic** elements: `<form>`, `<table>`, and `` - Clearly defines its content.

New Semantic Elements in HTML5

- ▶ `<article>`
- ▶ `<aside>`
- ▶ `<details>`
- ▶ `<figcaption>`
- ▶ `<figure>`
- ▶ `<footer>`
- ▶ `<header>`
- ▶ `<main>`
- ▶ `<mark>`
- ▶ `<nav>`
- ▶ `<section>`
- ▶ `<summary>`
- ▶ `<time>`

New HTML5 API's (Application Programming Interfaces)

- ▶ The most interesting new API's are:
- ▶ HTML Geolocation
- ▶ HTML Drag and Drop
- ▶ HTML Local Storage
- ▶ HTML Application Cache
- ▶ HTML Web Workers
- ▶ HTML SSE

HTML5 <section> Element

- ▶ The <section> element defines a section in a document.
- ▶ According to W3C's HTML5 documentation: "A section is a thematic grouping of content, typically with a heading."
- ▶ A Web site's home page could be split into sections for introduction, content, and contact information.
- ▶ `<section>`
 `<h1>WWF</h1>`
 `<p>The World Wide Fund for Nature (WWF) is....</p>`
 `</section>`

HTML5 <article> Element

- ▶ The <article> element specifies independent, self-contained content.
- ▶ An article should make sense on its own, and it should be possible to read it independently from the rest of the web site.
- ▶ Examples of where an <article> element can be used:
 - ▶ Forum post
 - ▶ Blog post
 - ▶ Newspaper article
- ▶ `<article>`
 - `<h1>What Does WWF Do?</h1>`
 - `<p>WWF's mission is to stop the degradation of our planet's natural environment, and build a future in which humans live in harmony with nature.</p>`
 - `</article>`

HTML5 <header> Element

- ▶ The <header> element specifies a header for a document or section.
- ▶ The <header> element should be used as a container for introductory content.
- ▶ You can have several <header> elements in one document.
- ▶ The following example defines a header for an article:
- ▶

```
<article>  
  <header>  
    <h1>What Does WWF Do?</h1>  
    <p>WWF's mission:</p>  
  </header>  
  <p>WWF's mission is to stop the degradation of our planet's natural environment,  
  and build a future in which humans live in harmony with nature.</p>  
</article>
```

Tag	Description
<article>	Defines an article in the document
<aside>	Defines content aside from the page content
<bdi>	Defines a part of text that might be formatted in a different direction from other text
<details>	Defines additional details that the user can view or hide
<dialog>	Defines a dialog box or window
<figcaption>	Defines a caption for a <figure> element
<figure>	Defines self-contained content, like illustrations, diagrams, photos, code listings, etc.
<footer>	Defines a footer for the document or a section
<header>	Defines a header for the document or a section
<main>	Defines the main content of a document
<mark>	Defines marked or highlighted text
<menuitem>	Defines a command/menu item that the user can invoke from a popup menu
<meter>	Defines a scalar measurement within a known range (a gauge)
<nav>	Defines navigation links in the document

HTML Multimedia

► The HTML <video> Element

► `<video width="320" height="240" controls>`
 `<source src="movie.mp4" type="video/mp4">`
 `<source src="movie.ogg" type="video/ogg">`
 Your browser does not support the video tag.
`</video>`

► Audio on the Web

- Before HTML5, there was no standard for playing audio files on a web page.
- Before HTML5, audio files could only be played with a plug-in (like flash).
- The HTML5 <audio> element specifies a standard way to embed audio in a web page.
- `<audio controls>`
 `<source src="horse.ogg" type="audio/ogg">`
 `<source src="horse.mp3" type="audio/mpeg">`
 Your browser does not support the audio element.
`</audio>`

HTML5 Drag and Drop

```
▶ <script>
function allowDrop(ev) {
    ev.preventDefault();
}

function drag(ev) {
    ev.dataTransfer.setData("text", ev.target.id);
}

function drop(ev) {
    ev.preventDefault();
    var data = ev.dataTransfer.getData("text");
    ev.target.appendChild(document.getElementById(data));
}
</script>
</head>
<body>

<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>



</body>
```

HTML5 Local Storage

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

► HTML Local Storage Objects

- HTML local storage provides two objects for storing data on the client:
- `window.localStorage` - stores data with no expiration date
- `window.sessionStorage` - stores data for one session (data is lost when the browser tab is closed)
- Before using local storage, check browser support for `localStorage` and `sessionStorage`:
- ```
if(typeof(Storage) !== "undefined") {
 // Code for localStorage/sessionStorage.
} else {
 // Sorry! No Web Storage support..
}
```