

HTML



Contents

- Introduction, Structure, DOM
- Headings and Paragraphs
- Styles, Formatting and Colors
- Images
- Links
- Tables
- Lists
- Blocks and Spans
- Classes and IDs
- Forms

Introduction

- HTML is the standard markup language for creating Web pages.
- HTML stands for Hyper Text Markup Language
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements are represented by tags
- Browsers use tags to render the content of the page

Structure

```
<!DOCTYPE html>
<html lang="en">

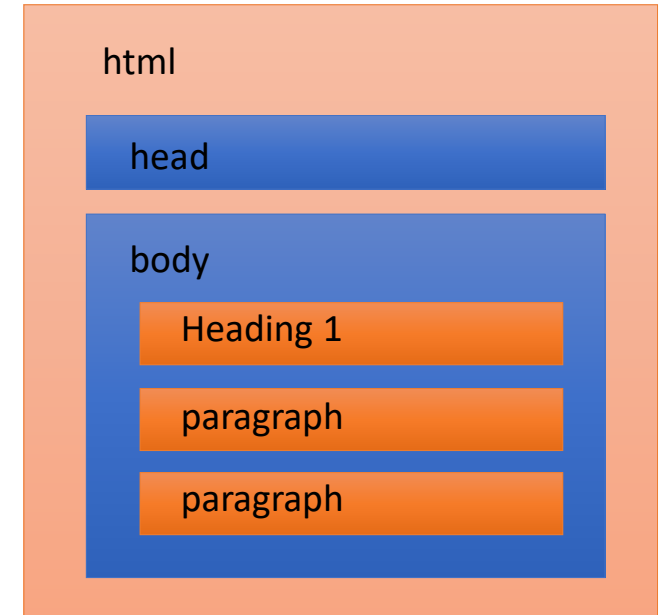
<head>
  <title>Hyper Text Mark-up Language</title>
</head>
```

```
<body>
  <h1>Learning HTML</h1>
  <p style="color:red">Build your website using HTML</p>
  <p style="color:green">It's the standard mark-up language for web pages</p>
  <p style="color:blue">You will enjoy it!</p>
</body>
</html>
```

Tags

Attribute

Value



<h1> </h1>

Opening and Closing Tags

DOM

- The Document Object Model (DOM) is a cross-platform and language-independent interface that treats an XML or HTML document as a tree structure wherein each node is an object representing a part of the document.
- The DOM represents a document with a logical tree
- It's important to be aware of this representation while we study the HTML

Views

Learning HTML

Build your website using HTML

It's the standard mark-up language for web pages

You will enjoy it!

Rendered View

<https://software.hixie.ch/utilities/js/live-dom-viewer/>
Or install DOM inspection plug-ins for your browser

```
└ DOCTYPE: html
  └ HTML lang="en"
    └ HEAD
      └ #text:
        └ TITLE
          └ #text: Hyper Text Mark-up Language
        └ #text:
      └ #text:
    └ BODY
      └ #text:
      └ H1
        └ #text: Learning HTML
      └ #text:
      └ P style="color:red"
        └ #text: Build your website using HTML
      └ #text:
      └ P style="color:green"
        └ #text: It's the standard mark-up language for web pages
      └ #text:
      └ P style="color:blue"
        └ #text: You will enjoy it!
      └ #text:
```

DOM View

Headings

- HTML defines six levels of headings using tags **<h1>** to **<h6>**
- Each level has a default font size

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



Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

- The default size can be overridden using the style attribute

```
<h1 style="font-size:60px;">Heading 1</h1>
```

Paragraphs

- The **<p>** tag define a paragraph
- A paragraph always starts from a new line
- It is usually a block of text

```
<p>This is a quote from Starwars</p>
```

```
<p>
```

```
"Do.
```

```
Or do not.
```

```
There is no try!"
```

```
- Yoda
```

```
</p>
```


Breaks

- Use `
` if a line break is needed
- The `<pre>` element define preformatted text
 - Preserves both line breaks and font
 - Font is usually fixed width Courier
 - Try printing a poem using `
` and `<pre>`
- The `<hr>` tag defines a thematic break in an HTML page
 - Often displayed as a horizontal rule
 - Can be used to separate content in an HTML page
 - Try using `<hr>` between two paragraphs

Comments

- HTML comments are specified using the `<!-- comment here -->` tags

Styles

- Setting the style of an HTML element can be done with style attribute
- Most of the styles are CSS attributes
- Basic syntax is:

```
<tagname style="property:value;">
```

- Some basic properties are:
 - background-color (eg: powderblue, #CFCFCF)
 - color (eg: red, green, blue)
 - font-family (eg: verdana, courier)
 - font-size (units are %, em or px)
 - text-align (eg: center)

Colors

- HTML colors can be specified using the color names
 - Supports nearly 140 color names:
https://www.w3schools.com/colors/colors_names.asp
- Colors can also be specified using RGB values, HEX values, HSL values, RGBA values, and HSLA values

```
<h1 style="background-color:rgb(255, 99, 71);">...</h1>  
<h1 style="background-color:#ff6347;">...</h1>  
<h1 style="background-color:hsl(9, 100%, 64%);">...</h1>  
<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>  
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">...</h1>  
<h1 style="border:2px solid Violet;">Hello World</h1>
```

HTML Images

- Images can be added to the HTML content using the **** tag
- The HTML attributes accepted by the tag are:
 - **src** attribute to define the URL of the image
 - **alt** attribute to define an alternate text for an image, if it cannot be displayed
 - **width** and **height** defines the size of the image (use CSS style instead)
- Also, CSS tags such as **width** and **height** properties to define the size of the image, specified against **style** attribute
- Use the CSS **float** property to let the image float

```

```

Links

- Links allow users to click their way from page to page
- HTML links are hyperlinks defined with a **<a>** tag
- The tag generally takes two attributes:
 - **href** that specifies the destination address
 - **target** attribute specifies where to open the linked document
 - The target attribute can take the values: `_blank`, `_self`, `_parent`, `_top`
 - **title** specifies a little more information on the link

```
<a href="https://www.google.com/" target="_blank">Just Google!</a>
```

Using images and buttons as links

- The following examples show how images and buttons can be used as links:

```
<a href="default.asp">  
    
</a>
```

```
<button onclick="document.location = 'default.asp'">Click Me!</button>
```

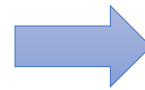
JS allows you to specify what should happen when a button is clicked.

Tables

- A table in HTML can be specified using the **<table>** tag
- Row is defined using the **<tr>** tag
- Table header is defined using the **<th>** tag
 - Table heading are bold and centered by default
- Table data is specified using the **<td>** tag
- **<caption>** tag can be used to specify captions to the table
 - Should be inserted right after **<table>**
- **colspan** and **rowspan** attributes can be specified for rows and columns spanning multiple cells
- Several CSS styling elements can be applied to format the tables

Tables

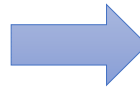
```
<table>
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>INR 100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>INR 50</td>
  </tr>
</table>
```



Month	Savings
January	INR 100
February	INR 50

Tables

```
<table style="width:50%">
  <caption>Telephone Index</caption>
  <tr>
    <th>Name:</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan="2">Telephone:</th>
    <td>55577854</td>
  </tr>
  <tr>
    <td>55577855</td>
  </tr>
</table>
```



Name:	Bill Gates
Telephone:	55577854
	55577855

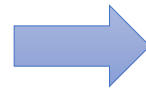
Lists

- HTML supports three types of lists
 - Ordered Lists – marked by numbers
 - Un-ordered Lists – marked by bullets
 - Description lists – list of terms with description of each item

Ordered Lists

- Ordered lists start with the **** tag
- Each list item starts with **** tag
- **type** attribute of **** define the list item marker
 - Values for **type** are: "1", "A", "a", "I", "i"

```
<h2>Ordered List with Numbers</h2>
<ol type="1">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```



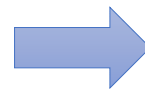
Ordered List with Numbers

1. Coffee
2. Tea
3. Milk

Unordered Lists

- Un - ordered lists start with the **** tag
- Each list item starts with **** tag
- Items will be marked by bullets by default
- The CSS **list-style-type** property is used to define the style of the list item marker
 - Values for **list-style-type** are: disc, circle, square, none

```
<h2>Unordered List with Disc Bullets</h2>
<ul style="list-style-type:disc;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```



Unordered List with Disc Bullets

- Coffee
- Tea
- Milk

Description Lists

- Description lists are defined in the **<dl>** tag
- The **<dt>** defines the term
- The **<dd>** tag describes each term

```
<h2>A Description List</h2>
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```



A Description List

Coffee
- black hot drink
Milk
- white cold drink

Nesting of Lists

- Lists can be nested one inside another

```
<h2>A Nested List</h2>
<p>List can be nested (lists inside lists):</p>
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
```



A Nested List

List can be nested (lists inside lists):

- Coffee
- Tea
 - Black tea
 - Green tea
- Milk

Block and Inline Elements

- A block level element always starts in a newline and takes up the full width available on the page
 - Example: The `<div>` element
- The inline element does not start in a newline and only takes up as much width as necessary
 - Example: The `` element

The <div> element

- The **<div>** element is often used as a container for other HTML elements
- The **style**, **class** and **id** attributes are commonly used
- In the CSS, **<div>** is used to style the blocks

```
<div style="background-color:coral;color:white;padding:20px">  
  <h2>Bengaluru</h2>  
  <p>Bengaluru is the capital city of Karnataka</p>  
</div>
```

Bengaluru

Bengaluru is the capital city of Karnataka

The element

- The **** is used as a container for some text
- The **style**, **class** and **id** attributes can be used

```
<h1>My <span style="color:red">Important</span> Heading</h1>
```

My Important Heading

Other Block Level Elements

<code><address></code>	<code><article></code>	<code><aside></code>	<code><blockquote></code>	<code><canvas></code>
<code><figcaption></code>	<code><figure></code>	<code><footer></code>	<code><form></code>	<code><h1>-<h6></code>
<code><noscript></code>	<code></code>	<code><p></code>	<code><pre></code>	<code><section></code>
<code><dd></code>	<code><div></code>	<code><dl></code>	<code><dt></code>	<code><fieldset></code>
<code><header></code>	<code><hr></code>	<code></code>	<code><main></code>	<code><nav></code>
<code><table></code>	<code><tfoot></code>	<code></code>	<code><video></code>	

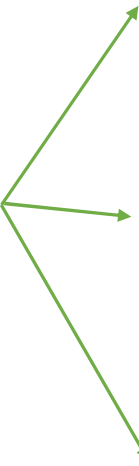
Other Inline Elements

<a>	<abbr>	<acronym>		<bdo>
<dfn>		<i>		<input>
<q>	<samp>	<script>	<select>	<small>
<time>	<tt>	<var>		
<big>	 	<button>	<cite>	<code>
<kbd>	<label>	<map>	<object>	<output>
		<sub>	<sup>	<textarea>

Classes

- The HTML **class** attribute is used to define equal styles for elements with same class name

```
<style>
.cities {
  background-color:coral;
  color: white;
  margin: 20px;
  padding: 20px;
}
</style>
```



```
<div class="cities">
  <h2>London</h2>
  <p>London is the capital of England.</p>
</div>
```

```
<div class="cities main">
  <h2>Paris</h2>
  <p>Paris is the capital of France.</p>
</div>
```

```
<div class="cities">
  <h2>Tokyo</h2>
  <p>Tokyo is the capital of Japan.</p>
</div>
```

Multiple Classes

- The **class** attribute can be specified for both block and inline elements
- Multiple classes can also be specified

```
<h2 class="city main">London</h2>
```

Different Tags with Same Class

- Different tags, like `<h2>` and `<p>`, can have the same class name and thereby share the same style

```
<h2 class="city">Paris</h2>
```

```
<p class="city">Paris is the capital of France</p>
```

IDs

- The id attribute specifies a unique id for an HTML element
- It should be unique within the HTML document
- It can be used by CSS and JS to perform certain task on the element

```
<style>
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}
</style>
```



```
<h1 id="myHeader">My Header</h1>
```


Difference between Class and ID

- An HTML element can only have one unique id that belongs to that single element, while a class name can be used by multiple elements

IDs as Bookmarks

```
<h2 id="C4">Chapter 4</h2>  
<a href="#C4">Jump to Chapter 4</a>
```

```
<h2 id="C4">Chapter 4</h2>  
<a href="book.html#C4">Jump to Chapter 4</a>
```

FORMS

- An HTML form is used to collect user input
- Input collected from the user is sent to a server for processing
- The **<form>** element defines a form which can contain form elements
- The **<input>** element is the most important element
- There are several input types such as text, url, date, checkbox, etc
- Among form attributes:
 - **action** defines the action to be performed when form is submitted
 - **target** defines where the submitted result will open
 - **method** defines whether to use HTTP Get or Post method
 - **novalidate** specifies that form-data should not be validated on submit

GET and POST

- The default HTTP method when submitting is GET
 - Appends form-data into the URL in name/value pairs
 - The length of a URL is limited (2048 characters)
 - Never use GET to send sensitive data! (will be visible in the URL)
 - Useful for form submissions where a user wants to bookmark the result
 - GET is better for non-secure data, like query strings in Google
- Use POST method to send sensitive and large amounts of information
 - POST has no size limitations and can be used to send large amounts of data.
 - Form submissions with POST cannot be bookmarked

Form Elements

- Most common is the **<input>** element and it can be displayed in several ways depending on its **type** attribute
- **<select>** defines a drop down list
 - **<option>** element defines options that can be selected
 - **size** attribute is used to specify the number of visible values
 - **multiple** attribute allows the user to select multiple values
- **<textarea>** defines a multi-line text input field
 - **rows** and **cols** can be specified to define the size of the text area
- **<button>** defines a clickable button
 - **onclick** specifies what should happen when the button is clicked

Form Elements

- **<fieldset>** is used to group related data in a form
 - **<legend>** defines a caption for the **<fieldset>**
- **<datalist>** element specifies a list of pre-defined options for **<input>**
 - Users will see a drop-down list of pre-defined options as they input data
 - Should be used with **<input>**
 - The list attribute of the **<input>** element, must refer to the id attribute of the **<datalist>** element
- The **<output>** element represents the result of a calculation

Form Input Types

```
<input type="button">  
<input type="checkbox">  
<input type="color">  
<input type="date">  
<input type="datetime-local">  
<input type="email">  
<input type="file">  
<input type="hidden">  
<input type="image">  
<input type="month">  
<input type="number">  
<input type="password">
```

```
<input type="radio">  
<input type="range">  
<input type="reset">  
<input type="search">  
<input type="submit">  
<input type="tel">  
<input type="text">  
<input type="time">  
<input type="url">  
<input type="week">
```

Form Input Attributes

Attribute	Description
checked	Specifies that an input field should be pre-selected when the page loads (for type="checkbox" or type="radio")
disabled	Specifies that an input field should be disabled
max	Specifies the maximum value for an input field
maxlength	Specifies the maximum number of character for an input field
min	Specifies the minimum value for an input field
pattern	Specifies a regular expression to check the input value against
readonly	Specifies that an input field is read only (cannot be changed)
required	Specifies that an input field is required (must be filled out)
size	Specifies the width (in characters) of an input field
step	Specifies the legal number intervals for an input field
value	Specifies the default value for an input field

Form Input Attributes

Attribute	Description
multiple	Specifies that user is allowed to specify more than one value
placeholder	Specifies a short hint that describes the expected value
autofocus	Specifies that an input field should be automatically highlighted when the page loads
height	Specifies height
width	Specifies width

Form Attributes for `<input>`

Attribute	Description
formaction	Specifies the URL of the file that will process the input when the form is submitted
formenctype	Specifies how the form data should be encoded when submitted (only with POST)
formmethod	Defines the HTTP method for sending the form-data to the action URL (override form attribute's)
formtarget	Specifies a name or a keyword that indicates where to display the response that is received after submitting the form
formnovalidate	Specifies that an <code><input></code> element should not be validated when submitted.

Constraint Validation – HTML and CSS

Attribute	Description
disabled	Specifies that the input element should be disabled
max	Specifies the maximum value of an input element
min	Specifies the minimum value of an input element
pattern	Specifies the value pattern of an input element
required	Specifies that the input field requires an element
type	Specifies the type of an input element

Selector	Description
:disabled	Selects input elements with the "disabled" attribute specified
:invalid	Selects input elements with invalid values
:optional	Selects input elements with no "required" attribute specified
:required	Selects input elements with the "required" attribute specified
:valid	Selects input elements with valid values

Automatic Form Validation

```
<form action="/action_page.php" method="post">  
  <input type="text" name="fname" required>  
  <input type="submit" value="Submit">  
</form>
```