HTML

```
<!DOCTYPE html>

<html>

<head>    </head>

<body>    </body>

</html>
```

1. !Doctype html : specify the html version (we use version 5) .
2. html : the beginning and the end of the code .
3. head : Set page title, charset, SEO tags, and link styles/scripts .
4. body : Structure and display text, images, links, forms, etc.
   NOTES :

- There is an important tool called "inspect" (show the source code for the website)
  To get it , right click in the site .
- `<html lang="en">` : specify the language in the search engines .

Head Components

`<h1>` to< `h6>` vary in head sizes .
`<title></title>` :Defines the page title shown in the browser tab and in search engine result .
`<meta charset="UTF-8">` : Sets the character encoding for the document, usually to UTF-8 to support most languages and symbols.
`<meta name="description" content="Description of the page">` :Provides a short description for search engines, improving SEO by giving a summary for search results.
`<meta name="keywords" content="keyword1, keyword2, keyword3">` :Lists keywords related to the page's content .
`<meta name="author" content="John Doe">` :Define the author of a page.
`<meta http-equiv="refresh" content="30">` :Refresh document every 30 seconds.
**NOTE : if you want to change the user from user to another using html (`<meta http-equiv="refresh" content="3 url=https://www.google.com/">`).

`<meta name="viewport" content="width=device-width, initial-scale=1.0">` :Setting the viewport to make your website look good on all devices.

BODY

`<p></p>` to make a paragraph in a separated line .
`<br>` to make a line break(can be used in paragraph too).

```
<pre>
hello

   my

     name

     is mohanad
</pre>
```

- pre is to make a paragraph (like p) however, it shows the block of code like how I write it.
  `<hr>` to make a line to separation between two sentences .
  `<p> Lorem (n)</p>` make a paragraph contains (n) words.
  NOTES :
  - to make a ready website code write <! then select !.
  - `<!-- -->` to make a comment .

Formatting Text

- All formatting text are in line.
- `<b>` - Bold text
- `<strong>` - Important text
- `<i>` - Italic text
- `<em>` - Emphasized text
- `<mark>` - Marked text
- `<small>` - Smaller text
- `<del>` - Deleted text
- `<ins>` - Inserted text
- `<sub>` - Subscript text (the word is down)
- `<sup>` - Superscript text (the word is up)

HTML Image

`<img src="(path of the image)" alt="the description of the image">`

- it is prefer to put the image in the same folder with html file.
- The `title` attribute provides additional information about an HTML element.
- `height` and `width` attributes control the height and width in pixels.
- `border` attribute makes a border around the element.

HTML Audio

`<audio src=""></audio>` supports only one extension.

```
    <audio controls loop muted autoplay>
     <source src="music.mp3" type="audio/mpeg">
     <source src="music.wav" type="audio/wav">
     <source src="music.ogg" type="audio/ogg">
   </audio>
```

Attributes :

- `controls` attribute displays playback controls like play, pause, and volume for audio or video elements.
- `muted` attribute ensures the media starts playing without sound.
- `loop` attribute makes the media automatically repeat once it reaches the end.
- `autoplay` attribute starts the media playback as soon as it loads, without user interaction.
  NOTES:
- The usage of `<source>` tag is to make sure that the browser handles all types (mpeg/wav/ogg).
- The complier read the code my sequence so it reads mp3 if it isn't exist it reads wav at the same way.

HTML Video

we can use the same attributes of audio .

```
<video  controls loop muted autoplay width="400" height="500"
poster="image.jpg" preload="auto">

    <source src="music.mp4" type="audio/mp4">
```

```
    <source src="music.webm" type="audio/webm">
    <source src="music.ogg" type="audio/ogg">

    <track src="a.vtt" kind="captions" srclang="en" label="English">
    <track  default src="a.vtt" kind="captions" srclang="ar" label="Arabic">
  </video>
```

Addition attributes in video :

- `poster` : to make a poster for the video.
- `preload` attribute : controls how a browser preloads media (audio or video) files to manage bandwidth and improve load times. It has three main options:

1. `preload="auto"` : The browser will load the entire file if possible, anticipating that the user will play it.
2. `preload="none"` : The media only begins to load when the user initiates playback, saving the most bandwidth.
3. `preload="metadata"` : Only the metadata (like duration and dimensions) is loaded, not the full file. This uses less bandwidth.

`<track>` tag : Improves accessibility by displaying text in sync with media content.

Common Attributes of `<track>` :

- `src` : The file path to the text track file (typically a `.vtt` file).
- `kind` : Specifies the type of track includes options :

1. `subtitles` : Provides translations of the dialogue.
2. `captions` : Contains transcription for accessibility (including non-dialogue sounds).
3. `descriptions` : Gives audio descriptions of visual content.
4. `chapters` : Divides content into chapters.
5. `metadata` : Used for custom data not visible to the user.

- `srclang` : Specifies the language of the track like (`srclang="en"`) .
- `label` : A user-visible label for the track, often displayed in media controls like (label="English")
- `default` : Marks the track as the default choice.
  NOTE: the difference between `scrlang` and `label` is : `srclang` identifies the track's language code for the browser, while `label` is a descriptive name for the user.

Tables

- Table tag : organizes data in rows and columns.

```
<table border="1">
  <thead>
    <tr>
      <th>Product</th>
      <th>Price</th>
      <th>Quantity</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Apples</td>
      <td>$1.00</td>
      <td>10</td>
    </tr>
    <tr>
      <td>Bananas</td>
      <td>$0.50</td>
      <td>20</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>Total</td>
      <td>$1.50</td>
      <td>30</td>
    </tr>
  </tfoot>
</table>
```

Structure of the table

- `<tr>` **(Table Row)**: Defines a row (Each `<tr>` is a new row in the table).
- `<th>` **(Table Header)**: Defines a header cell within a row.
  - Typically used in the first row (or column) to label each column or row (usually be bold and centered by default).
- `<td>` **(Table Data)**: Defines a regular data cell within a row.
  - Each `<td>` is a cell containing data within the table.

- `<thead>`, `<tbody>`, **and** `<tfoot>` **(Optional)**: Grouping elements for structure.
  - `<thead>`: Wraps the table's header rows.
  - `<tbody>`: Wraps the main content rows.
  - `<tfoot>`: Wraps the footer rows, typically containing summaries or totals.

  Important Attributes :
- `colspan`: Makes a cell span across multiple columns.
- `rowspan`: Makes a cell span across multiple rows.
- `border`: Makes a borders between rows and columns.
- `begclor`: Makes a background color for a row or a column.

  ex :

```
<table border="1">
<tr>
<th rowspan="2">Product</th>
<th rowspan="2">Price</th>
<th bgclor="red">Quantity</th>
</tr>
</table>
```

Lists

There are three types of lists are used to display items `ul`, `ol` and `dl`.

- `<ul>` **(Unordered List) :Creates an unordered (bulleted) list.**
  Attributes of `<ul>` :** `type` attribute and its options are :
- `"disc"` (default bullet point).
- `"circle"` (hollow circle).
- `"square"` (square bullet).
  ex : `<ul type="circle"> <li>apple </li>`
- `<ol>` **(Ordered List) :** It is typically displayed with numbers, letters, or Roman numerals.
  Attributes of `<ol>` :
  `type` attribute and its options are :
- `"1"` (default), `"A"`, `"a"`, `"I"`, `"i"`.
  `start` : Specifies the starting number (or letter or Roman numeral) for the ordered list
  ex : `<ol start="5">`
  `reversed` : Causes the list to be numbered in reverse order (i.e., descending).
  ex : `<ol reversed>`
- `<dl>` **(Description List) :** It consists of `<dt>` (description term) and `<dd>` (description definition) elements.

ex :

```
<dl>
  <dt>HTML</dt>
  <dd>A markup language used to structure web content.</dd>
  <dt>CSS</dt>
  <dd>A language used to style and layout web pages.</dd>
</dl>
```

- `<dt>` for the title and `<dd>` for the description .

Semantic Elements

```
 <div> </div>

    <header> </header>

    <main> </main>

    <section> </section>

    <footer> </footer>

    <aside> </aside>

    <nav> </nav>

    <article> </article>
```

- The purpose of semantic elements is to improve the **accessibility, readability, and maintainability** of web pages and making it easier for both developers and browsers to understand the content.

links

- The `<a>` (anchor) tag in HTML is used to create hyperlinks, allowing users to navigate to other web pages, files, email addresses, or sections within the same page.
  Essential attributes in links :

1. `href` **(Hypertext Reference)**: Specifies the URL the link points to, and it can direct users to:

- An **external site** (e.g., `<a href="https://www.example.com">`),
- An **internal page** on the same site (e.g., `<a href="/about.html">`),
- A **specific section** on the same page (e.g., `<a href="#section-id">`),
- An **email** link (`mailto:`) or a **phone number** link (`tel:`).

2. `target`: Controls where the linked document will open:

- `_self`: Opens in the same tab (default).
- `_blank`: Opens in a new tab or window.
  NOTE : id attribute is used to give the element an unique name ,ex: `<section id="contact">Contact Us</section>`

Form and input action method

```
<form action="textx.php" method="dialog">
 <fieldset>
<legend>login</legend>

<div>
    <label for="txt">user name</label>
    <input id="txt" type="text" required name="user">
</div>

<div>
    <label for="pass"> password</label>
    <input type="password" id="pass" required name="pass">
    <br>
    <input type="submit">
</div>

</fieldset>
</form>
```

- `<form></form>` Creates a form allowing users to input and submit data.
- `action="test.php"` specifies the URL (`test.php`) where the form data will be sent for processing.

- `method="dialog"` : it’s generally used to display form dialogs in certain contexts without actually submitting the data to a server.

Difference between method titles :

1. `get` : Suitable for retrieving or displaying data in the URL, useful for non-sensitive data.
2. `post` : Suitable for securely sending sensitive data, as it doesn’t expose data in the URL.
3. `dialog` : Experimental and used in limited contexts for form modals or dialogs.

- `<fieldset>` : Groups related elements within the form for better structure and styling.
- `<legend>login</legend>` : Provides a caption for the `<fieldset>`, labeled "login" to describe the form's purpose.
- `<div>` : Creates a container to organize form elements (making the in a a separated block).
- `<label for="txt">user name</label>` : This label describes the input field, making it clear that it’s for the username.
- The `for="txt"` attribute links the label to the input field with `id="txt"`, improving accessibility.
- `<input id="txt" type="text" required>` : Defines a text input field where users can enter their username.
  NOTE : `type` attribute have many options like : text, password, number, ...
- `id="txt"` : Gives the input an identifier, linking it to the label.
- `type="text"` : Specifies a single-line text field.
- `required` : Makes the field mandatory before submission
- `name="user"` : Assigns a name to the input data, used by the server to reference this field’s value when processing the form.
- `<input type="submit">` : Creates a submit button that, when clicked, sends the form data to the server or an assigned action.
  Important titles in input :
- `color` : To select a color (ex: `<input type="color">`).
- `file` : To select a file (ex: `<input type="file">`).
- `<input type="range" min="0" max="100" step="2" value="0">`
  `min max` : To control the the beginning and the end of the range .
  `step` :Defines the intervals at which values can be selected .
  `value` : Sets the initial position of the slider.
  NOTE: we can use the same attributes with title `number` .
- `search` : To make a search bar .
- `url` : Cannot accept any input except `url`
- `hidden` : Specifies that the input is hidden.
  Control Date and Time :

- `date` : Controls year, month and day .
- `month` : Controls year and month .
- `time` : Controls time .
- `datetime-local` : Controls year, month, day and time.
- `week` : Controls weeks.

Radio and Checkbox :

```
<div>
    <label for="x">Maths</label>
    <input checked id="x" type="radio" name="subject" value="maths">
</div>
<div>
    <label for="y">Physics</label>
    <input type="radio" id="y" name="subject" value="physics">
    <input type="submit" value="Submit">
</div>
```

- The `<input type="radio">` element allows users to select one option. Each radio button shares the same name attribute, which groups them together and ensures only one option can be selected at a time.
- `checked` attribute : Creates a radio button pre-selected .
- `name="subject"` : The name of the group .
- `value="maths"` : The data the will be sent to the back-end developer or the server.
- `<input type="checkbox" id="pass" name="n" value="pass">` : it is like radio but it provides choosing many options .

Select and Datalist options :

```
 <label for="x">what is your country ?</label>
 <select name="country" id="x" multiple>
<option value="eg">egypt</option>
<option selected value="su">suria</option>
<option value="mor">morocco</option>
<input type="submit">
</select>
```

- `select` : To make a select options
- `multiple` : Provides multiple selection if clicks `ctrl` .

- The options are be written in `option` and to make a pre-selected we use `selected` .
  NOTE : we can divide them into groups with `optgroup` tag .ex :

```
<optgroup label="section">
<option value="eg">egypt</option>
<option selected value="su">suria</option>
<option value="mor">morocco</option>
<input type="submit">
</optgroup>
```

in DATALIST option :

```
<input list="subjects" name="subject" id="subject">
<datalist id="subjects">
  <option value="Maths">
  <option value="Physics">
  <option value="Chemistry">
</datalist>
```

- `<datalist>` : Defines the list of options.
- `<option>` : Represents each option in the list.
- `list` **attribute**: Associates the `<input>` field with the `<datalist>` by referencing its `id` .

Buttons :

- `<button>` : A versatile button that can be used for custom actions like triggering JavaScript or form submission.
- `<input type="submit">` : A button used to submit a form.
- `<input type="reset">` : Resets all form fields to their initial values.
- `<input type="button">` : A basic button used for custom functions, typically with JavaScript.
- `<button type="submit">` : Similar to `<input type="submit">` , but more flexible, allowing HTML content inside.
- `<button type="reset">` : Resets form fields, similar to `<input type="reset">` , but more flexible.

Validate Text Area :

```
<form>
    <textarea name="message"
     required
      minlength="10"
       maxlength="200"
        cols="30"
        rows="5"
        placeholder="Enter your message...">
        autofocus
        </textarea>
    <input type="submit" value="Submit">
</form>
```

- `required` : Ensures the `textarea` cannot be empty before submitting the form.
- `minlength="10"` : Requires at least 10 characters to be entered.
- `maxlength="200"` : Limits the input to a maximum of 200 characters.
- `cols="30"` : Specifies the visible width of the `textarea` (number of characters).
- `rows="5"` : Specifies the visible height of the `textarea` (number of lines).
- `autofocus` : Making the browser to focus on the `textarea` by default .

NOTES:

- you can stop the validation using `<form novalidate>` even writing validation attributes.
- The `readonly` attribute makes the content of an input or text area field un-editable but allows it to be submitted with the form data.
- The `disabled` attribute completely disables the field, making it un-editable and preventing it from being submitted with the form.
- you can use `target` attribute in form to control where the data be sent. ex: `<form target="_blank">` .

iframe, object and embed

- `<iframe>` : Used to embed external HTML documents within the page.
- `<object>` : Embeds various types of external resources (e.g., PDFs, images) with support for fallback content.
- `<embed>` : Embeds multimedia content (audio, video) directly and does not support fallback content.

  ex:

```
<iframe src="https://www.example.com" width="300" height="200"> </iframe>
<object data="example.pdf" type="application/pdf" width="600" height="400">
</object>
<embed src="audio.mp3" type="audio/mpeg" width="300" height="50">
```

Key Difference: `<iframe>` is primarily for HTML documents and it is give more flexibility with different browsers, `<object>` is versatile with fallback, and `<embed>` is simplified for multimedia without fallback.

Download files

using `<a download="profile pic.jpeg" href="profile pic.jpeg">pic</a>`

- we can change the name in `downloads` and the browser will know it automatically .
- we can use live server by downloading live server extension and open the file with it to make a the download process .