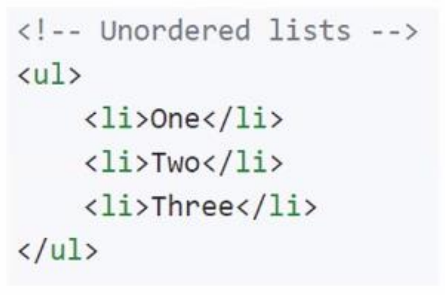
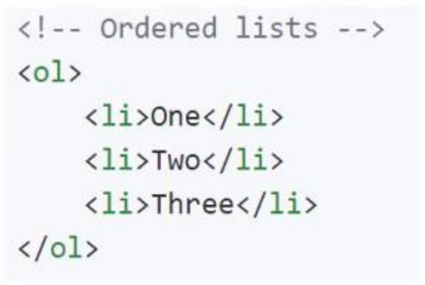
HTML Cont. 2

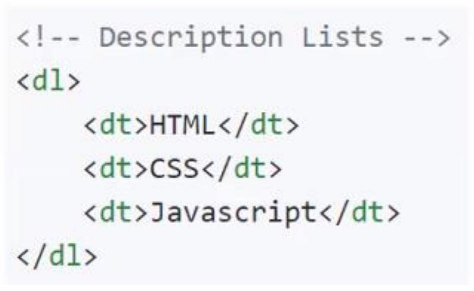
Lists:



```
<!-- Unordered lists -->
<ul>
    <li>One</li>
    <li>Two</li>
    <li>Three</li>
</ul>
```



```
<!-- Ordered lists -->
<ol>
    <li>One</li>
    <li>Two</li>
    <li>Three</li>
</ol>
```



```
<!-- Description Lists -->
<dl>
    <dt>HTML</dt>
    <dt>CSS</dt>
    <dt>Javascript</dt>
</dl>
```

Unordered HTML List:

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag. The list items will be marked with bullets (small black circles) by default:

Example

INPUT:	**OUTPUT:**
<ul> <li>Coffee</li> <li>Tea</li> <li>Milk</li> </ul>	• Coffee • Tea • Milk

Unordered HTML List - Choose List Item Marker

The CSS list-style-type property is used to define the style of the list item marker. It can have one of the following values:

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

Example – Disc

Input:	**Output:**
<ul style="list-style-type:disc;"> <li>Coffee</li> <li>Tea</li> <li>Milk</li> </ul>	• Coffee • Tea • Milk

Example - Circle

Input:	**Output:**
<ul style="list-style-type:circle;"> <li>Coffee</li> <li>Tea</li> <li>Milk</li> </ul>	○ Coffee ○ Tea ○ Milk

Example - Square

Input:	**Output:**
<ul style="list-style-type:square;"> <li>Coffee</li> <li>Tea</li> <li>Milk</li> </ul>	▪ Coffee ▪ Tea ▪ Milk

Example - None

Input:	**Ouput:**
<ul style="list-style-type:none;"> <li>Coffee</li> <li>Tea</li> <li>Milk</li> </ul>	Coffee Tea Milk

Nested HTML Lists:

Lists can be nested (list inside list):

Example:

Input:	**Output:**
<ul> <li>Coffee</li> <li>Tea <ul> <li>Black tea</li> <li>Green tea</li> </ul> </li> <li>Milk</li> </ul>	• Coffee • Tea ○ Black tea ○ Green tea • Milk

Ordered HTML List:

An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.

The list items will be marked with numbers by default:

Example:

Input:

```
<ol>
 <li>Coffee</li>
 <li>Tea</li>
 <li>Milk</li>
</ol>
```

Output:

1. Coffee
2. Tea
3. Milk

Ordered HTML List - The Type Attribute

The type attribute of the <ol> tag, defines the type of the list item 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

Nested HTML Lists:

Lists can be nested (list inside list):

Example

Input:

```
<ol>
 <li>Coffee</li>
 <li>Tea
 <ol>
  <li>Black tea</li>
  <li>Green tea</li>
 </ol>
 </li>
 <li>Milk</li>
</ol>
```

Output:

1. Coffee
2. Tea
    1. Black tea
    2. Green tea
3. Milk

Description List

We see there are no bullet points on any formatting. But when we use <dd> tag,

Ex: 1

INPUT:

```
<dl>
        <dt>HTML</dt>
                <dd>This is a description for HTML</dd>
                <dd>This is a description for HTML</dd>
        <dt>CSS</dt>
        dt>Javascript</dt>
</dl>
```

OUTPUT:

```
HTML
        This is a description for HTML
        This is a description for HTML
CSS
Javascript
```

Ex: 2

INPUT:

```
<dl>
        <dt>Coffee</dt>
                <dd>- black hot drink</dd>
        <dt>Milk</dt>
                <dd>- white cold drink</dd>
<dl>
```

OUTPUT:

```
Coffee

        - black hot drink
Milk
        - white cold drink
```

Block Level Element:

A block-level element always starts on a new line. A block-level element always takes up the full width available (stretches out to the left and right as far as it can). A block level element has a top and a bottom margin, whereas an inline element does not.

Here are the block-level elements in HTML:

<address>
<article>
<aside>
<blockquote>
<canvas>
<dd>
<div>
<dl>
<dt>
<fieldset>
<figcaption>
<figure>
<footer>
<form>
<h1>-<h6>
<header>
<hr>
<li>
<main>
<nav>
<noscript>
<ol>
<p>
<pre>

<section>

<table>

<tfoot>

<ul>

<video>

Inline Elements:

An inline element does not start on a new line. An inline element only takes up as much width as necessary.

Here are the inline elements in HTML:

<a>

<abbr>

<acronym>

<b>

<bdo>

<big>

<br>

<button>

<cite>

<code>

<dfn>

<em>

<i>

<img>

<input>

<kbd>

<label>

<map>

<object>

<output>

<q>

<samp>

<script>

<select>

<small>

<span>

<strong>

<sub>

<sup>

<textarea>

<time>

<tt>

<var>

Summary

- There are two display values: block and inline
- A block-level element always starts on a new line and takes up the full width available
- An inline element does not start on a new line and it only takes up as much width as necessary
- The <div> element is a block-level and is often used as a container for other HTML elements
- The <span> element is an inline container used to mark up a part of a text, or a part of a document

More HTML Elements:

Tables:

The <table> tag defines an HTML table.

Each table row is defined with a <tr> tag. Each table header is defined with a <th> tag. Each table data/cell is defined with a <td> tag.

By default, the text in <th> elements are bold and centered.

By default, the text in <td> elements are regular and left-aligned. They can contain all sorts of HTML elements; text, images, lists, other tables, etc.

Input:	**Output:**

<table>
    <tr>
        <td>1</td>
        <td>2</td>
        <td>3</td>
    </tr>
    <tr>
        <td>a</td>
        <td>b</td>
        <td>c</td>
    </tr>
</table>
	1 2 3
	a b c

<thead> and <tbody> gives more meaning to the content. The browser will understand that everything in <thead> tag would be heading and in the <tbody> is content.

Just as <td> is for <tbody>, <th> is for <thead>

Input:

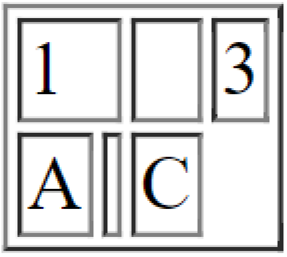
```
<table>
        <thead>
                <tr>
                        <th>Serial No.</th>
                        <th>Name</th>
                        <th>Phone</th>
                </tr>
        </thead>
        <tbody>
                <tr>
                <td>1</td>
                        <td>Yash</td>
                   <td>654984321354</td>
                </tr>
                <tr>
            <td>2</td>
            <td>Cash</td>
        <td>65497321654</td>
                </tr>
        </tbody>
</table>
```

Output:

```
Serial No. Name      Phone
1          Yash  654984321354
2          Cash  65497321654
```

Ex: 02

```
<table border='true'>
        <tr>
                <td colspan='2'>1<td>
                <td>3</td>
   </tr>
   <tr>
                <td>A<td>
                <td>C</td>
        </tr>
</table>
```



1		3
A		C

Input:

Input is a self-closing tag. <input>, creates a box where we can write.

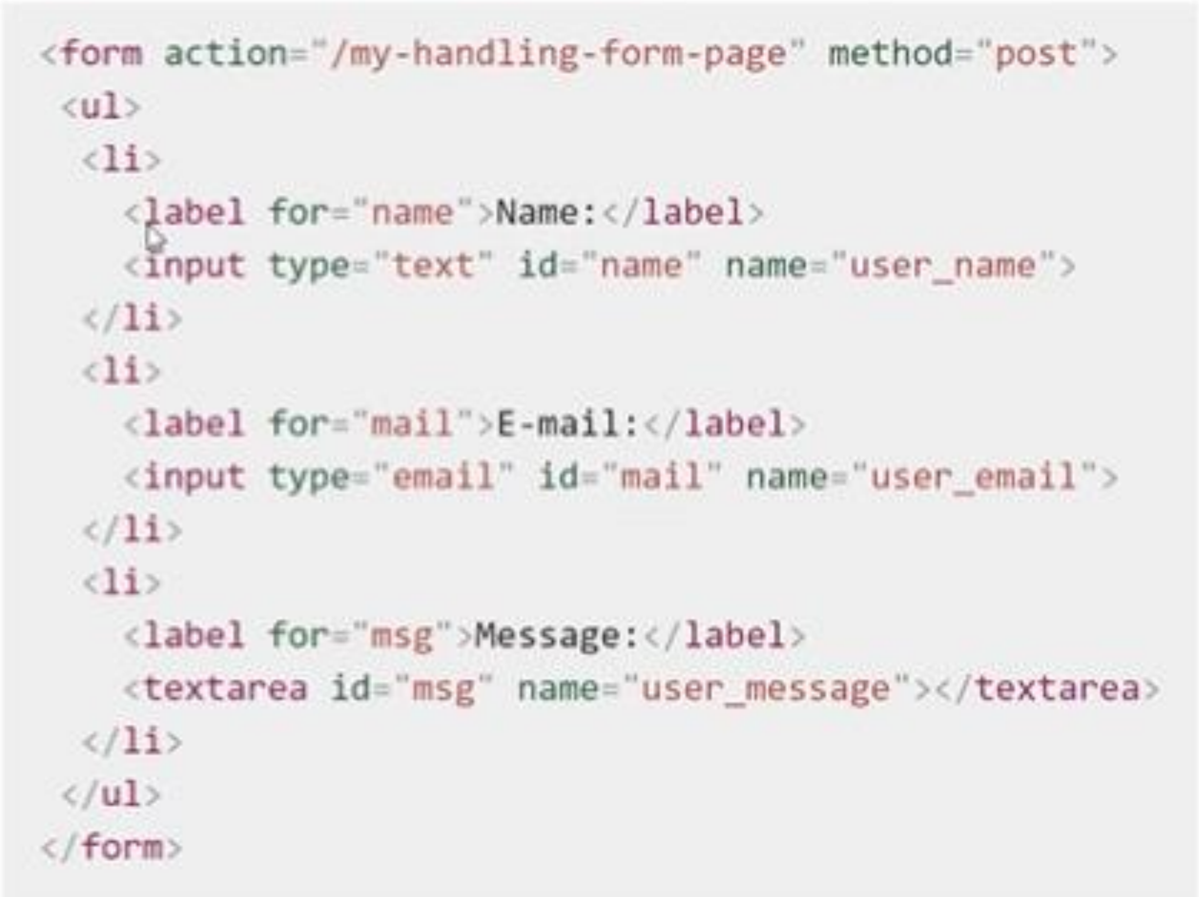
```
<input type="value">
```

Attribute Values

Value	Description
button	Defines a clickable button (mostly used with a JavaScript to activate a
checkbox	Defines a checkbox
color	Defines a color picker
date	Defines a date control (year, month, day (no time))
datetime-local	Defines a date and time control (year, month, day, time (no timezone)
email	Defines a field for an e-mail address
file	Defines a file-select field and a "Browse" button (for file uploads)
hidden	Defines a hidden input field
image	Defines an image as the submit button
month	Defines a month and year control (no timezone)
number	Defines a field for entering a number
password	Defines a password field
radio	Defines a radio button
range	Defines a range control (like a slider control)
reset	Defines a reset button
search	Defines a text field for entering a search string
submit	Defines a submit button
tel	Defines a field for entering a telephone number

text	Default. Defines a single-line text field
time	Defines a control for entering a time (no timezone)
url	Defines a field for entering a URL
week	Defines a week and year control (no timezone)

Form

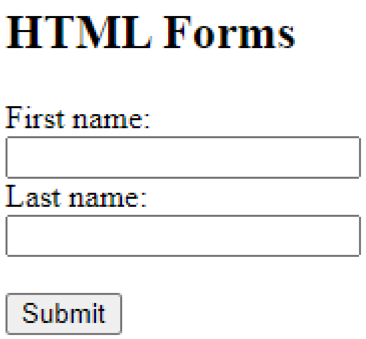


```
<form action="/my-handling-form-page" method="post">
 <ul>
  <li>
    <label for="name">Name:</label>
    <input type="text" id="name" name="user_name">
  </li>
  <li>
    <label for="mail">E-mail:</label>
    <input type="email" id="mail" name="user_email">
  </li>
  <li>
    <label for="msg">Message:</label>
    <textarea id="msg" name="user_message"></textarea>
  </li>
 </ul>
</form>
```

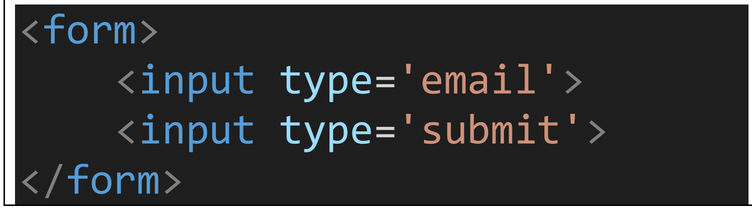
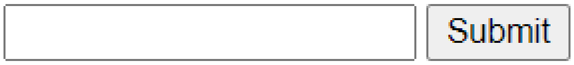
<form> is not a self-closing tag. It is a block level element.

```
<form action="/action_page.php">
        <label for="fname">First name:</label><br>
        <input type="text" id="fname" name="fname"><br>
        <label for="lname">Last name:</label><br>
        <input type="text" id="lname" name="lname"><br><br>
        <input type="submit" value="Submit">
</form>
```

OUTPUT:

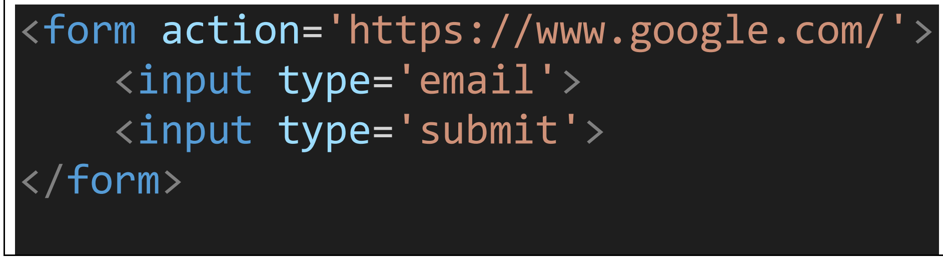


Input:	**Output:**

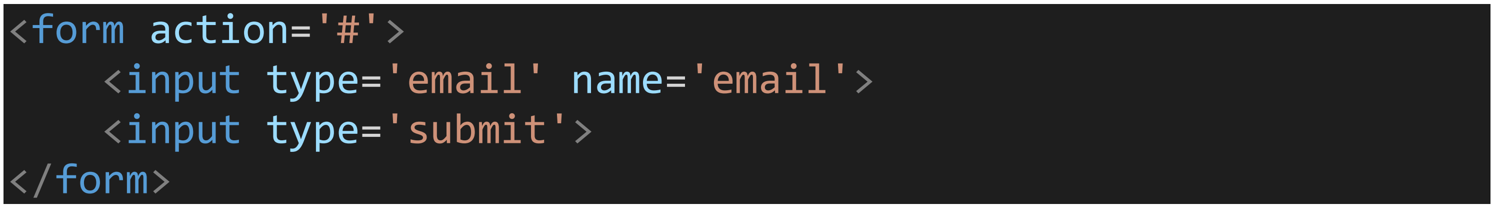

`<form>`
`    <input type='email'>`
`    <input type='submit'>`
`</form>`
	

When we click on submit, there is a '?' added to the URL at the end. When we will a button, it will refresh the page for the browser because it needs to send that data. The data is sent to a server and this happens every time we click submit

Input:	


`<form action='https://www.google.com/'>`
`    <input type='email'>`
`    <input type='submit'>`
`</form>`	

This will open google.com every time you click the submit button.



```
<form action='#'>
    <input type='email' name='email'>
    <input type='submit'>
</form>
```

When you click submit button now, the url changes to **email=name%40mail.com**

file:///C:/Users/Desktop/prc.html

file:///C:/Users/Desktop/prc.html?email=name%40mail.com#

```
<form action='#'>
    <input type='email' name='email'>
    <input type='password' name='pass'>
    <input type='submit'>
</form>
```

When we click submit, the email and password would be added to the url

file:///C:/Users/Desktop/prc.html

file:///C:/Users/aakas/Desktop/prc.html?email=name%40mail.com&pass=password#

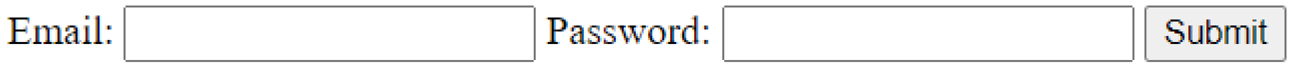
Label:

It is used to label an element. If we want to label this email element,

```
<form action='#'>
    <label for='emailID'>Email:</label>
    <input id='emailID' type='email' name='email'>
    <input type='password' name='pass'>
    <input type='submit'>
</form>
```

The above code will add the label 'Email' to the html code.

```
<form action='#'>
    <label for='emailID'>Email:</label>
    <input id='emailID' type='email' name='email'>
    <label for='password'>Password:</label>
    <input id='password' type='password' name='pass'>
    <input type='submit'>
</form>
```



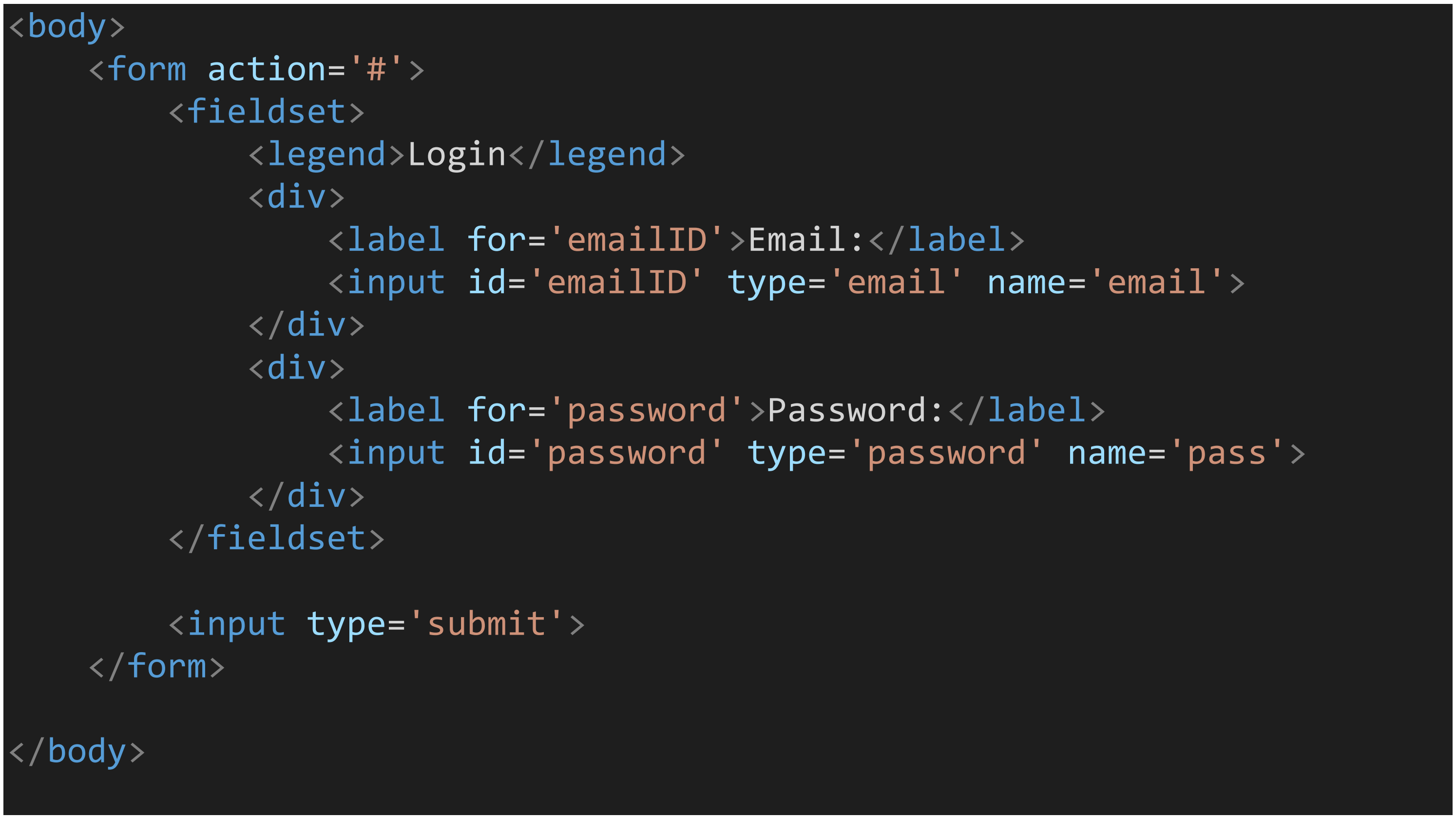
Without providing the 'id' method, we can do nexting in the label.

```
<label>
    Email:
      <input type='email' name='email'>
</label>
```

<div> is a block level element used to divide. If we put the label tag into their own division, then everything in the label becomes a block element and takes the complete space. Until now, every input came side-by-side, but once we put the <form> and <label> into the <div>, it becomes a block level element and occupies the complete width of the web page.

<input> and <label> has some width. Span only takes the combined width of email and input box.

<fieldset>, <legend>

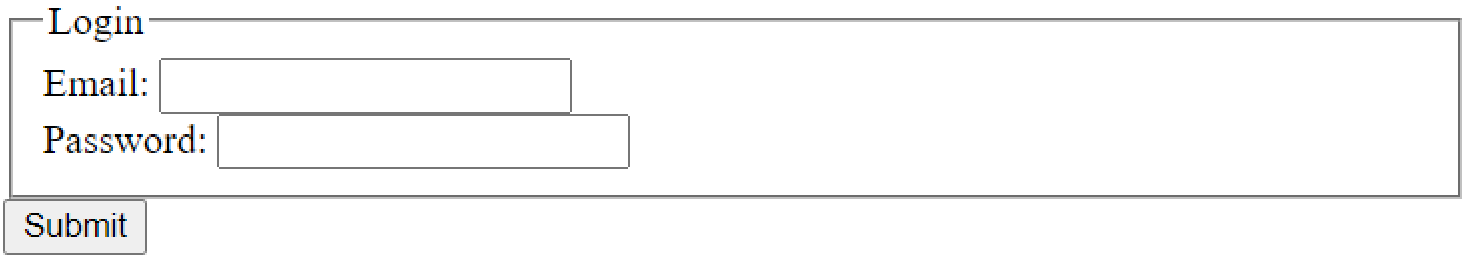


```
<body>
    <form action='#'>
        <fieldset>
            <legend>Login</legend>
            <div>
                <label for='emailID'>Email:</label>
                <input id='emailID' type='email' name='email'>
            </div>
            <div>
                <label for='password'>Password:</label>
                <input id='password' type='password' name='pass'>
            </div>
        </fieldset>

        <input type='submit'>
    </form>

</body>
```

Output:


Login
Email:
Password:
Submit