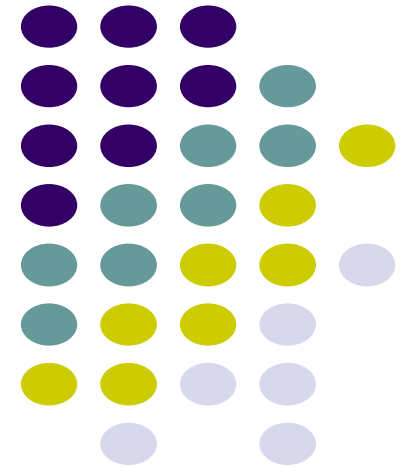


# HTML Forms

**Dr. Arul Xavier V. M**





# HTML Form

- HTML Forms are required, when you want to collect some data from the site visitor.
  - For example, during user registration you would like to collect information such as name, email address, credit card, etc.
- A form will take input from the site visitor and then will post it to a back-end application.
  - The back-end application will perform required processing on the passed data based on defined business logic inside the application.

# Creating a HTML Form

← → ↻ ⓘ localhost:8383/MyHTMLProject/sample.html

## Sign In

Email

Password

← → ↻ ⓘ localhost:8383/MyHTMLProject/sample.html

## Register Here

Name

Email

Phone Number

Choose Password

# The `<form>` Element



- The HTML `<form>` element defines a form that is used to collect user input and submit to registered server URL.
- An HTML form contains **form elements**.
  - Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons, and more.

## HTML Form Example

First name:

Last name:

`<form>`

•  
*form elements*

•  
`</form>`



# <form> tag

- The **<form>** tag is used to create an HTML form for user input.
- Also, it is a container for input elements or tags.
- Basic Attributes: **action**, **method**
  - **action**-Specifies where to send the form-data when a form is submitted(URL).
  - **method**-Specifies the HTTP method to use when sending form-data(GET or POST)

```
<body>  
  <form action="server.php" method="POST">  
    <!-- Form input elements goes here -->  
  
  </form>  
</body>
```

server url

**action** and **method** are an important **<form>** attributes to specify the server URL and method of submitting form data such as get,post



# What is HTTP?

- The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers.
- HTTP works as a request-response protocol between a client and server.
  - Example:
    - A client (browser) submits an HTTP request to the server; then the server returns a response to the client.
    - The response contains status information about the request and may also contain the requested content.



# Get method vs. Post method

- **GET is one of the most common HTTP methods.**
  - GET requests **not secured** and it can be cached
  - GET requests remain in the browser history
  - GET requests can be bookmarked
  - GET requests should never be used when dealing with sensitive data
  - GET requests have length restrictions(2,048 characters-minus number of characters in URL)
- **POST is one of the most common HTTP methods.**
  - POST requests are **secured**, never cached
  - POST requests do not remain in the browser history
  - POST requests cannot be bookmarked
  - POST requests have no restrictions on data length

# Form Elements



- The `<input>` Element
  - The most important form element is the `<input>` element.
  - The `<input>` element can be displayed in several ways, depending on the type attribute.
- The `<select>` Element
  - The `<select>` element defines a drop-down list.
- The `<textarea>` Element
  - The `<textarea>` element defines a multi-line input field (a text area).
- The `<button>` Element
  - The `<button>` element defines a clickable button.





## <input> tag

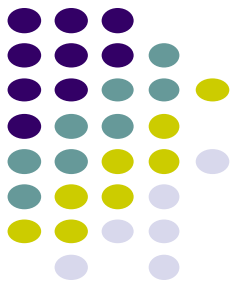
- The <input> tag is the most important form element.
- The <input> tag can be displayed in several ways, depending on the type attribute.



# HTML Input Types

Input Types	Description
<code>&lt;input type="text"&gt;</code>	defines a one-line text input field
<code>&lt;input type="password"&gt;</code>	defines a password field
<code>&lt;input type="email"&gt;</code>	defines a pre-validated email box
<code>&lt;input type="date"&gt;</code>	defines a calendar date picker
<code>&lt;input type="time"&gt;</code>	allows the user to select a time (no time zone).
<code>&lt;input type="number"&gt;</code>	defines a numeric input field.
<code>&lt;input type="color"&gt;</code>	is used for input fields that should contain a color. color chooser
<code>&lt;input type="radio"&gt;</code>	defines a radio button
<code>&lt;input type="checkbox"&gt;</code>	defines a checkbox.

# HTML Input Types



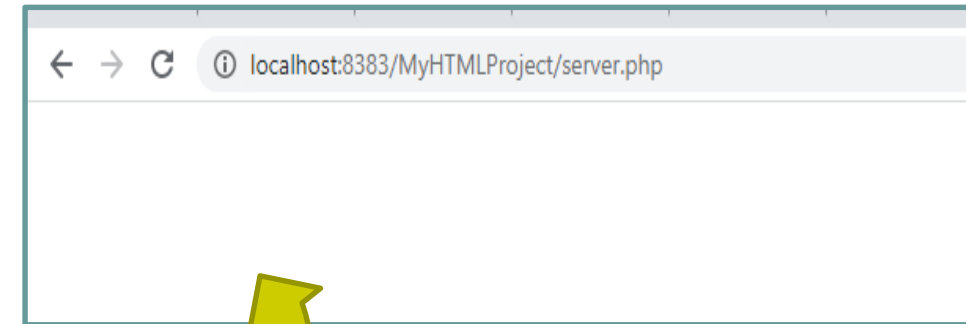
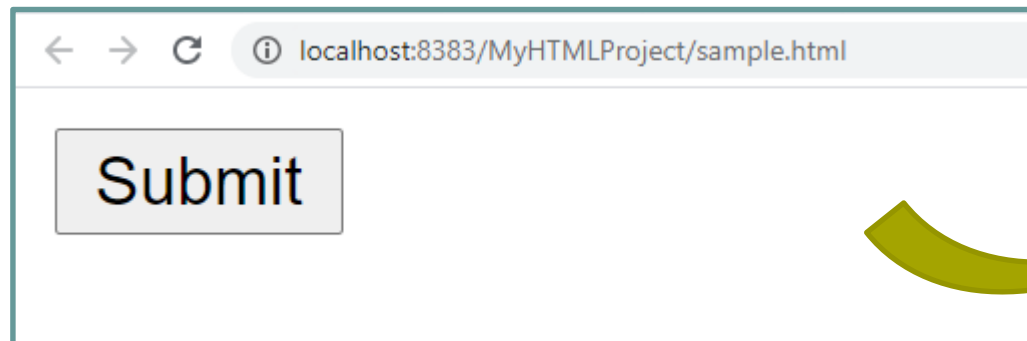
Input Types	Description
<code>&lt;input type="submit"&gt;</code>	defines a button for submitting form data to a form-handler
<code>&lt;input type="reset"&gt;</code>	defines a reset button that will reset all form values to their default values
<code>&lt;input type="button"&gt;</code>	defines a button
<code>&lt;input type="color"&gt;</code>	is used for input fields that should contain a color. color chooser
<code>&lt;input type="file"&gt;</code>	defines a file-select field and a "Browse" button for file uploads.
<code>&lt;input type="url"&gt;</code>	defines input box with url validation



# Submitting a form data to server

- `<input type="submit">` will create a button, it will submit the collected user data to the server URL specified in action attribute.

```
<body>  
  <form action="server.php" method="POST">  
    <!-- Form input elements goes here -->  
  
    <input type="submit" value="Submit" >  
  </form>  
</body>
```



When user click the "submit" button, page navigated to server.php

# Text Input Elements



- **Single-line text input box**– This control is used for items that require only one line of user input, such as search boxes or names.

Enter your Name: `<input type="text">`

- **Password input controls** – This is also a single-line text input but it **masks** the character as soon as a user enters it.

Enter Password: `<input type="password">`

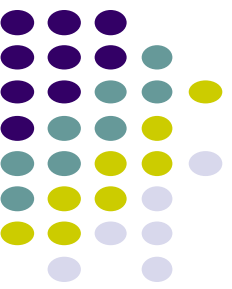
- **Multi-line text input controls** – Multi-line input controls are created using HTML `<textarea>` tag with mandatory attributes **rows(number of lines)** and **cols(number characters in each line)**.

Comments: `<textarea rows="5" cols="10"></textarea>`

- **Clear the Text Boxes** – HTML provides an input type element called “**reset**” which is used to clear the user entered data in text box, password box and text areas.

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

# Example



```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Table</title>
  </head>
  <body>
    <form action="/server" method="get">
      Name:    <input type="text" >
      <br><br>
      Password: <input type="password">
      <br><br>
      Comments:
      <textarea rows="5" cols="10"></textarea>
      <br><br>
      <input type="submit" value="Submit"> <input type="reset" type="reset">
    </form>
  </body>
</html>
```

Single Line Text Box

HTML Table

← → ↻ ⓘ File | D:/CLASS/Web%20Tech%202020/Manifold/images/sample.html

Name: Joy

Password: .....

Comments: This is so interesting website...

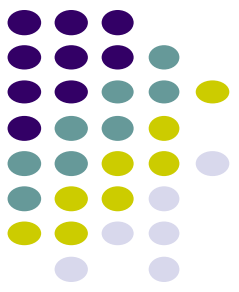
Submit Reset

Single Line Password Box

submit button reset button

Multi-line Text Box

# Other Input Elements



- **Normal Button:** `<input type="button" value="convert" onclick="JS_function">`

Example:

```
<body>
  Dollar($): <input type="text" id="dollar">
  <br><br>
  Rupee(₹): <input type="text" id="rupee" readonly>
  <br><br>
  <input type="button" value="convert" onclick="JS_function">
</body>
```

A screenshot of a web browser window. The address bar shows the file path 'D:/CLASS/Web%20Tech%202020/Manifold/images/abc.html'. The page content displays the rendered HTML: 'Dollar(\$):' followed by a text input field, 'Rupee(₹):' followed by a disabled text input field, and a 'convert' button below them.

← → ↻ ⓘ File | D:/CLASS/Web%20Tech%202020/Manifold/images/abc.html

Dollar(\$):

Rupee(₹):

convert



# Other Input Elements



- **Radio Button:** The `<input type="radio">` defines a radio button. Radio buttons are normally presented in radio groups (a collection of radio buttons describing a set of related options). Only one radio button in a group can be selected at the same time. The radio group must have shared the same name (the value of the name attribute) to be treated as a group.

```
<form action="sever.php" method="get">
```

Select the Gender:

```
<input type="radio" name="gender" value="Male">Male
```

```
<input type="radio" name="gender" value="Female">Female
```

```
<br><br>
```

```
<input type="submit">
```

```
</form>
```

localhost:8383/Example/index.html

Select the Gender: ☒ Male ☐ Female

Submit



localhost:8383/Example/sever.php?gender=Male

gender=Male is submitted

localhost:8383/Example/index.html

Select the Gender: ☐ Male ☒ Female

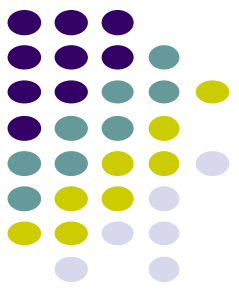
Submit



localhost:8383/Example/sever.php?gender=Female

gender=Female is submitted

# Other Input Elements



- **Checkboxes:** The HTML `<input type= “checkbox”>` is used to define a checkbox field. The checkbox is shown as a square box that is ticked when it is activated. It allows the user to select one or more option among all the limited choices.

```
<form action="sever.php" method="get">
```

Choose your Skills:

```
<input type="checkbox" name="skills[]" value="HTML">HTML
```

```
<input type="checkbox" name="skills[]" value="CSS">CSS
```

```
<input type="checkbox" name="skills[]" value="PHP">PHP
```

```
<input type="checkbox" name="skills[]" value="AngularJS">AngularJS
```

```
<input type="checkbox" name="skills[]" value="Javascript">Javascript
```

```
<br><br>
```

```
<input type="submit">
```

```
</form>
```

Choose your Skills: ☒ HTML ☐ CSS ☒ PHP ☒ AngularJS ☐ Javascript

Submit

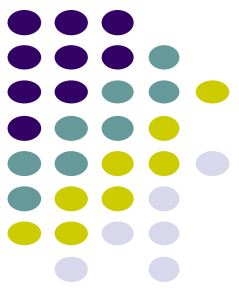


localhost:8383/Example/sever.php?skills=HTML&skills=PHP&skills=AngularJS

The selected values will be submitted in skills array variable

# Creating a Drop Down List Box

## - (Combox Box)



- The `<select>` element is used to create a drop-down list. The `<select>` element is most often used in a form, to collect user input.
- The name attribute is needed to reference the form data after the form is submitted (if you omit the name attribute, no data from the drop-down list will be submitted).

```
<form action="sever.php" method="get">  
  Select Your State:  
  <select name="state">  
    <option>Kerala</option>  
    <option>Tamilnadu</option>  
    <option>Andhra Pradesh</option>  
    <option>Karnataka</option>  
    <option>Manipur</option>  
    <option>Punjab</option>  
  </select>  
  <br><br>  
  <input type="submit">  
</form>
```

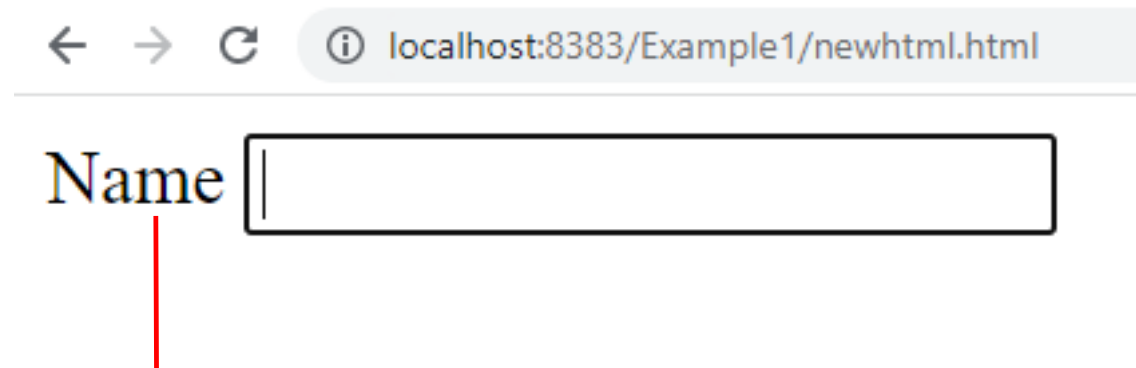


# <label> tag for input elements



- The <label> tag defines a label for input elements
  - Users who have difficulty clicking on very small regions (such as checkboxes)
  - So, when a user clicks the text within the <label> element, it toggles the input (this increases the hit area).
  - The **for** attribute of <label> must be equal to the **id** attribute of the related element to bind them together.

```
<label for="namefield">Name</label>  
<input type="text" id="namefield">
```



When you click label, the linked input tag gets the focus automatically.



## Use of <fieldset>

- The <fieldset> tag is used to group related elements in a form.
- The <fieldset> tag draws a box around the related elements.

# Use of <fieldset> in HTML form



```
<!DOCTYPE html>
<html>
<body>

<form>
  <fieldset>
    <legend>Personalia:</legend>
    Name: <input type="text"><br>
    Email: <input type="text"><br>
    Date of birth: <input type="text">
  </fieldset>
</form>

</body>
</html>
```

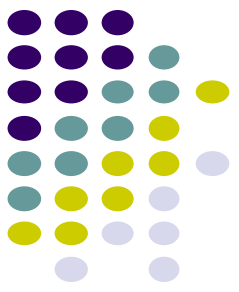
Personalia: \_\_\_\_\_

Name:

Email:

Date of birth:

# Other Input Elements



- **Email** input box:

Enter Email-Id: `<input type="email" name="emailid">`

A screenshot of a web browser window. The address bar shows the file path 'D:/CLASS/Web%20Tech%202020/Manifold/images/abc.html'. The page content shows a label 'Enter Email-Id:' followed by a text input field containing 'abcd'. Below the input field is a yellow warning icon and a message: 'Please include an '@' in the email address. 'abcd' is missing an '@'. At the bottom of the form is a 'Submit' button.

- **Phone number** input box:

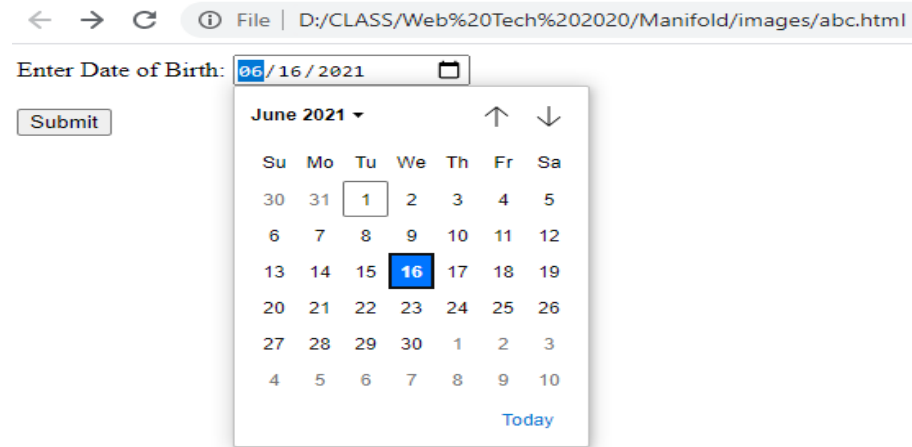
Phone Number: `<input type="tel" name="phone" pattern="[0-9]{10}">`

A screenshot of a web browser window. The address bar shows the file path 'D:/CLASS/Web%20Tech%202020/Manifold/images/abc.html'. The page content shows a label 'Phone Number:' followed by a text input field containing '22323'. Below the input field is a yellow warning icon and a message: 'Please match the requested format.' At the bottom of the form is a 'Submit' button.

# Other Input Elements

- **Date picker:**

Enter Date of Birth: `<input type="date" name="dob">`



Enter Date of Birth:

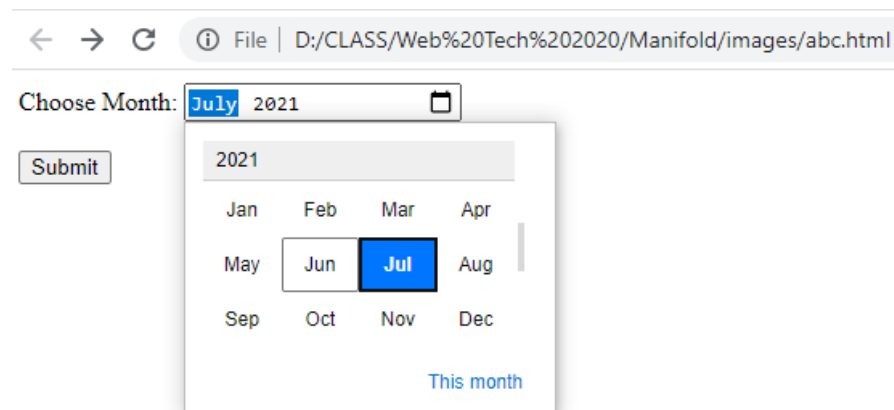
June 2021

Su	Mo	Tu	We	Th	Fr	Sa
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

Today

- **Month Selector:**

Choose Month: `<input type="month" name="month">`

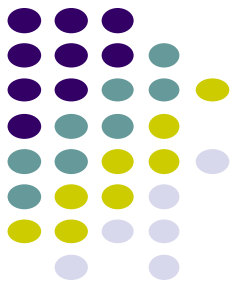


Choose Month:

2021

Jan	Feb	Mar	Apr
May	Jun	Jul	Aug
Sep	Oct	Nov	Dec

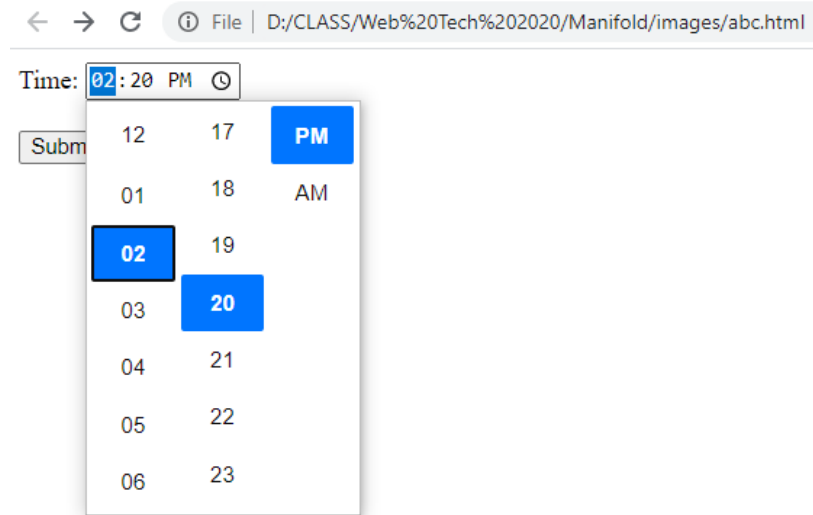
This month



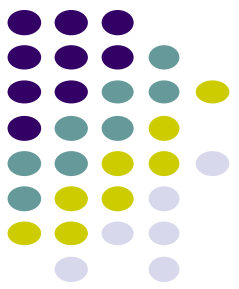
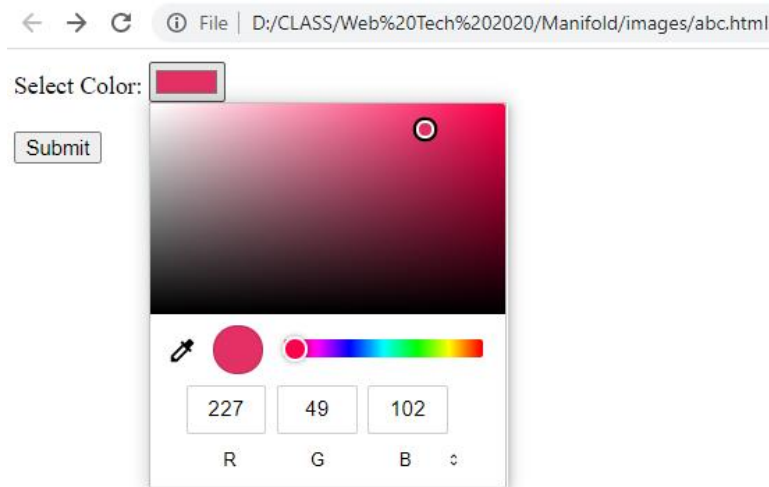


# Other Input Elements

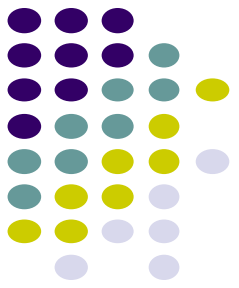
- **Time Picker:** Time: `<input type="time" name="timedata">`



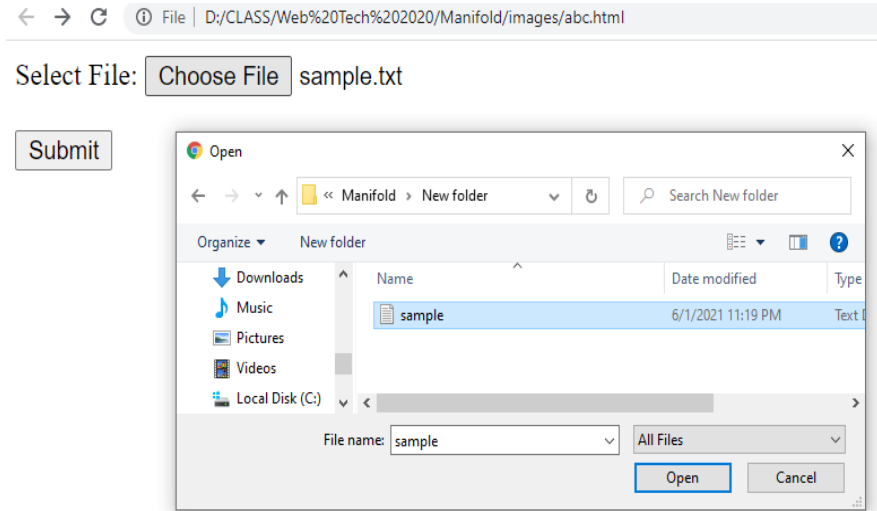
- **Color Picker:** Select Color: `<input type="color" name="usercolor">`



# Other Input Elements

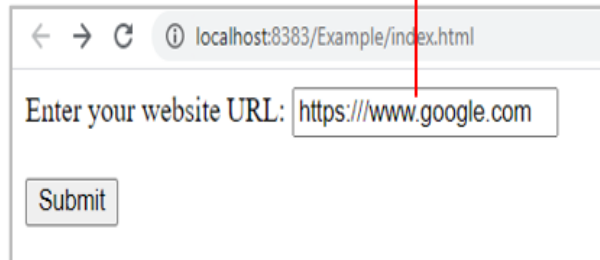


- **File Upload:** Select File: `<input type="file" name="filename">`

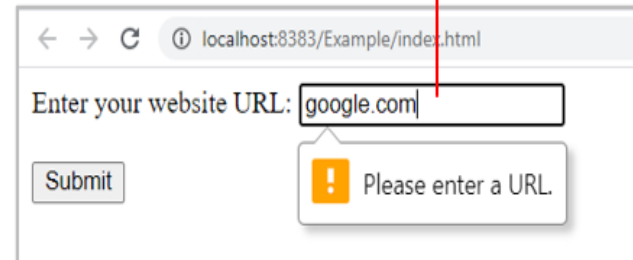


- **URL Text Input:** Enter your Website URL: `<input type="url" name="weburl">`

Correct URL Format



For invalid URL format it will generate Error





# The `<button>` Element

- The `<button>` tag is used to create clickable buttons on the web page. The difference between these elements and buttons created with the `<input>` tag is that you can place the content (images or text and tags like `<i>`, `<b>`, `<strong>`, `<br>`, `<img>`, etc.) inside the `<button>`
- **Note:** Always specify the `type` attribute for a `<button>` element, to tell browsers what type of button it is. By default, this tag will create a normal clickable button.
- You can change using the attribute `value` as “submit” or “reset”.

```
<button type="button">Normal Button</button>
```

```
<button type="submit">Submit</button>
```

```
<button type="reset">Reset</button>
```

# HTML Interactive Element



- The **<progress>** tag:

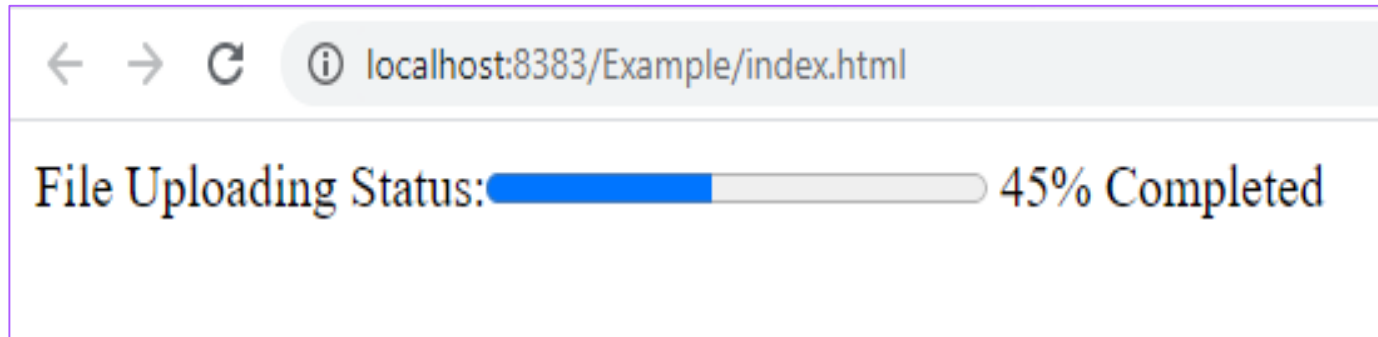
The **<progress>** tag represents the completion progress of a task. Use the **<progress>** tag in conjunction with JavaScript to display the progress of a task.

```
<body>
```

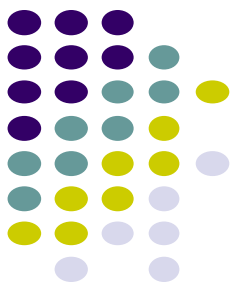
```
  File Uploading Status:
```

```
    <progress value="45" max="100"></progress> 45% Completed
```

```
</body>
```



# HTML <input> Tag Attributes

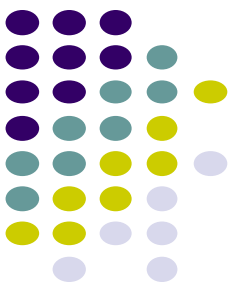


Input Attribute	Description
value	Specifies an initial value for an input field
readonly	Specifies that an input field is read-only which cannot be modified by the user. (boolean type)
size	Specifies the visible width, in characters, of an input field
maxlength	Specifies the maximum number of characters allowed in an input field
placeholder	Short hint that describes the expected value of an input field (a sample value or a short description of the expected format). It is displayed in the input field before the user enters a value.
required	Specifies that an input field must be filled out before submitting the form. (boolean type)
autofocus	Specifies that an input field should automatically get focus when the page loads. (boolean type)
name	Specify unique <b>name</b> for the input field such as text box, etc. Also, this name attribute used to autocomplete the value and also for sending data to server as parameter.

# HTML <input> Tag Attributes



Input Attribute	Description
disabled	Specifies that an input field should be disabled which is unusable and un-clickable (boolean type)
min	Specify the minimum value for an input field
max	Specify the maximum value for an input field
pattern	Specifies a regular expression that the input field's value is checked against, when the form is submitted.



# Example

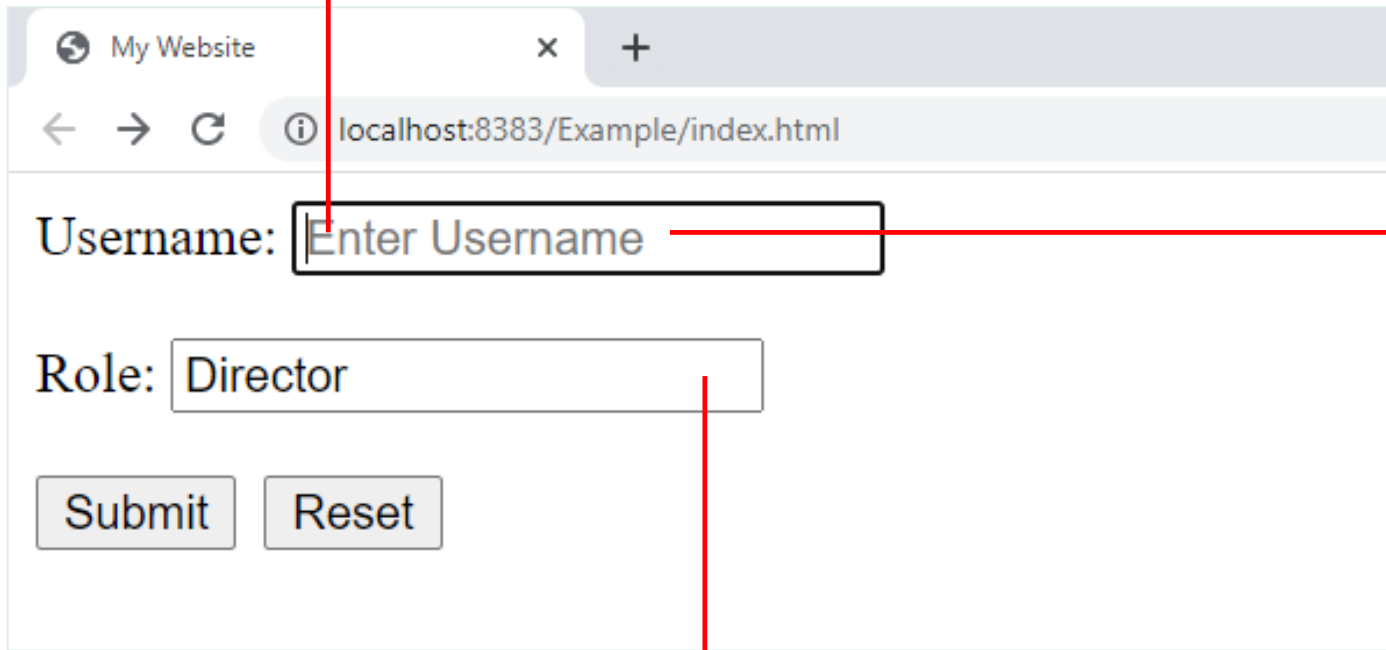
```
<form action="server.php">
    Username:  <input type="text" size="20" placeholder="Enter Username"
                maxlength="10" autofocus required>

    <br>
    <br>
    Role:  <input type="text" value="Director" readonly>
    <br>
    <br>
    <input type="submit">&nbsp;&nbsp;&nbsp;<input type="reset">
</form>
```

**&nbsp;** - HTML Special Character to display 1 space, you can repeat this depends on how many spaces you want.

# Output

Cursor is focused inside the text box due to **autofocus** attribute is set.



The screenshot shows a web browser window with the title 'My Website' and the address bar displaying 'localhost:8383/Example/index.html'. The page contains a form with two text input fields and two buttons. The first field is labeled 'Username:' and contains the placeholder text 'Enter Username'. A red vertical line points from the text above to the cursor in this field. The second field is labeled 'Role:' and contains the text 'Director'. A red vertical line points from the text below to this field. Below the fields are two buttons labeled 'Submit' and 'Reset'.

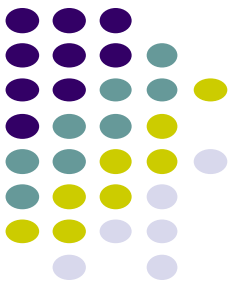
\* Here, “Enter Username” is placeholder text due to **placeholder=“Enter Username”**.

\* Also, text box is 20 characters wide since **size=20**

- In Role Text Box showing initial text as “Director” since **value=“Director”**
- Also, the value cannot be modified due to **readonly** attribute is set.



# Output



- When user click the submit button without typing any data in Username text box will generate an error message as shown below, since the **required** attribute is set which makes that field is mandatory to fill.

A screenshot of a web browser showing a form at localhost:8383/Example/index.html. The form has a 'Username:' label and a text box containing 'Enter Username'. Below it is a 'Role:' label and a text box containing 'Dire'. An orange error message box with an exclamation mark icon is displayed over the 'Role' text box, stating 'Please fill out this field.' At the bottom of the form are 'Submit' and 'Reset' buttons.

- Moreover, we have set the maxlength as 10 characters wide using the attribute **maxlength="10"**. Due to this, user cannot type more than 10 characters inside the text box.

A screenshot of a web browser showing the same form at localhost:8383/Example/index.html. The 'Username' text box now contains the text 'abcdefghij', which is exactly 10 characters long. The 'Role' text box still contains 'Director'. The 'Submit' and 'Reset' buttons are at the bottom.

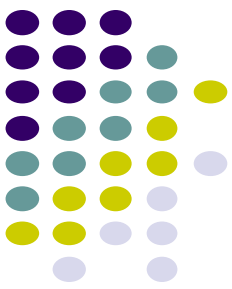


`<input type="number">`

- defines a numeric input field.

## Display a Number Field

Quantity (between 1 and 5):



# min, max, name

```
<form action="server.php" method="get">
  Account Name: <input type="text" name="name">
  <br><br>
  Enter the PIN: <input type="number" name="pin"
                  min="1000" max="9999">
  <br><br>
  <input type="submit">&nbsp;&nbsp;&nbsp;<input type="reset">
</form>
```

Autocomplete(this will work only  
if the name attribute is specified)

The screenshot shows a web browser window with the address bar displaying 'localhost:8383/Example/index.html'. The form contains two input fields: 'Account Name' and 'Enter the PIN'. The 'Account Name' field has the letter 'A' entered, and a dropdown menu is visible below it, showing a list of suggestions: 'Arul Xavier', 'a', 'Arul', 'arulvmax', 'arul', and 'Arul Xavier V M'. Below the input fields are two buttons: 'Submit' and 'Reset'.

The number box provides increment and decrement  
button to increase or decrease number.

The screenshot shows a web browser window with the address bar displaying 'localhost:8383/Example/index.html'. The form contains two input fields: 'Account Name' and 'Enter the PIN'. The 'Account Name' field has the text 'Arul' entered. The 'Enter the PIN' field has the number '1246' entered, and a small increment/decrement button is visible to the right of the input field. Below the input fields are two buttons: 'Submit' and 'Reset'.



If the number entered is below the min=1000 or above the max=9999 specified will generate an error message when user click the “submit” button as given below.

A screenshot of a web browser interface. The address bar shows 'localhost:8383/Example/index.html'. Below it, there is a form with two fields: 'Account Name:' with the value 'Arul' and 'Enter the PIN:' with the value '1234567'. A red error message box is displayed below the PIN field, stating 'Value must be less than or equal to 9999.'.

← → ↻ ⓘ localhost:8383/Example/index.html

Account Name:

Enter the PIN:

❗ Value must be less than or equal to 9999.

# pattern attribute



- The **pattern** attribute specifies a **regular expression** that the **<input>** element's value is checked against on form submission.
- **Note:** The pattern attribute works with the following input types: text, date, search, url, tel, email, and password.

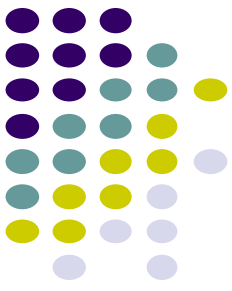
# pattern attribute



The **pattern** attribute specifies a regular expression that the **<input>** element's value is checked against on form submission.

```
<form action="server.php" method="get">
  Choose Username:<input type="text" name="username">
  <br><br>
  Create Password:<input type="password" name="password"
                    pattern="[a-zA-Z]{8,}">
  <small>(Password that must contain 8 or more characters including lowercase,
uppercase letters)</small>
  <br><br>
  <input type="submit" value="Sign Up">
</form>
```

# pattern attribute



Another example, if you want to check a password must contain 8 or more characters that are of at **least one number, one uppercase , lowercase letter and one special symbols**: You can use the following pattern

```
<form action="server.php" method="get">
  Choose Username:<input type="text" name="username" required>
  <br><br>
  Create Password:<input type="password" name="password"
                    pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[@$*!^&~]).{8,}" required>
  <small>
    (Password Must contain at least one number and one uppercase and
    lowercase letter, and at least 8 or more characters)
  </small>
  <br><br>
  <input type="submit" value="Sign Up">
</form>
```

Regex	Description
<code>(?=.*\d)</code>	It matches any numerical digit in unlimited times
<code>(?=.*[a-z])</code>	It matches any lowercase letter from a-z in unlimited times
<code>(?=.*[A-Z])</code>	It matches any uppercase letter from A-Z in unlimited times
<code>.{8,}</code>	It matches minimum 8 characters and unlimited maximum characters

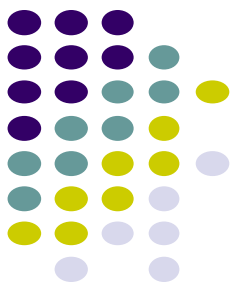
# disabled attribute



- The **disabled** attribute is a boolean attribute. When present, it specifies that the **<input>** element should be disabled. A disabled input element is **unusable** and **un-clickable**.
- The **disabled** attribute can be set to keep a user from using the **<input>** element until some other condition has been met (like selecting a checkbox, etc.). Then, a JavaScript could remove the disabled value, and make the **<input>** element usable.
- **Note:**
  - Disabled **<input>** elements in a form will not be submitted!



# disabled attribute



```
<form action="sever.php" method="get">
  Firstname: <input type="text" name="fname">
  <br><br>
  Lastname: <input type="text" name="lname" disabled>
  <br><br>
  <input type="submit" disabled >
</form>
```

A screenshot of a web browser window. The address bar shows the file path "D:/CLASS/Web%20Tech%202020/Manifold/images/abc.html". The form contains three elements: a "Firstname:" label followed by a text input field, a "Lastname:" label followed by a disabled text input field (indicated by a light gray background), and a disabled "Submit" button (also with a light gray background). Red lines point from the text annotations to the disabled input fields and the disabled submit button.

Here, the text box is disabled, we can enable it using Javascript on certain conditions

Here, this submit is disabled which is not clickable.  
**Usage:** we can include Javascript logic to enable it when user enter all correct data.