form elements

input tag

> input tag is used create a form elements

> these elements are used to take input/accept data from users.

> controls nothing but text field, button, radio button, file, color, password, etc...

> input is un-paired tag and inline element

Syn:

<input attributes/>

Note: by default, input tag creates “text field”.

attributes of input tag:

type ➔ it reps which type of control you want to create.

if we are not specified by default it creates a textbox.

input type="text|hidden|password|number|button|reset|submit|image|checkbox|radio|file|color|email|url|range|date|time|datetime-local|month|week..."

name ➔ it represents name of control, used in server-side programming (not unique)

id ➔ it represents id of control, used in client-side programming (unique)

value ➔ it used to initialize input elements (default value)

readonly ➔ this attribute not allowed to change the value of control

size ➔ this attribute specifies size of control (width of control)

disabled ➔ this attribute disabled the control

placeholder ➔ this attribute used to display prompting text with in the element

maxlength ➔ this attribute used to set max limit of data (no.of chars)

required ➔ this attribute used to force the user to enter data (mandatory field)

autofocus ➔ this attribute set's cursor position (Default location)

tabindex ➔ this attribute controlling cursor movements (when we r pressing tab key, where cursor is moving)

min ➔ this attribute used to set minimum limit of number value (lower bound)

max ➔ this attribute used to set maximum limit of number value (upper bound)

step ➔ this attribute used to specify incremental & decrement value of number

text fields

>text field used for taking input(any type of data) from user or text box allows us to type data.

>text fields are used for typing of username, first name, pincode, phone, address etc…

>by using "**input**" tag we can create text fields

>user can type any no.of char's but single line, if u want to set a limit use "maxlength" attribute.

>text field allows you to type data in 1 line.

Syn:

```
<input type="text" attributes/>
```

Note:

id ➔ used by client-side programming (javascript)

=> unique (duplicate id are not)

name ➔ used by server-side programming (servlet/asp.net/php...)

=> either unique or not

hidden field

>an in-visible text field with default value is called a "hidden field".

>hidden fields are used to send some information about user/client to a server-sideprogram without asking the user.

>to perform session tracking, we are using hidden fields.

>like normal text fields, hidden fields data is also sent to the server when we click the submit button.

Syn:

```
<input type="hidden" name="NAME" value="VAL" attributes>
```

password control

> password is a text field but the data is not visible.

> it's used for accepting password, pin, cvv, otp, verification code, passcode... from the user.

> password is displayed as * or dot

Syn:

<input type="password" attributes>

number control

>this control used to accept only numerical value (numbers only) from user

>it prevents typing of alphabets, special chars (in some browsers), except **minus**, **dot** and **e** alphabet.

>but some browser allows to type all these data but given error while submitting form

> some browsers displaying number fields with increment/decrement buttons

SYn:

```
<input type="number" attributes>
```

attributes:

min =>it rep starting value of number field (by default no min)

max =>it rep ending value of number field (by default no max)

step =>it rep increment/decrement value (by default +1/-1)

general button control

>button control is used to execute some code in the background whenever the user clicked/pressed on that button.

>we can execute either JS or jQuery or Angular

>its standard button (not submit), it is unable to send requests to the server.

> These buttons are used to perform some client-side operations.

> these button values are not sent to the server.

Syn:

<input type="button" attributes>

reset button

>reset is one type of button, used for reset the html form/UI, meaning it clears all values of form.

>reset button must be part of the "form" tag.

> these button values are not sent to the server.

Syn:

<input type="reset">

submit button

>submit is one type of button, used to submit/send the html forms/page/UI to a server-side program.

>while submitting the html form, all input parameters (enter by user), hidden parameter (by programmer) are prepared as a "Query String"

ex:

https://www.redbus.in/search.aspx ? param-name=value&param-name=value &...

param-name => name of text box or name of password field or name of button...

value => value of text box or value of password field or value of button...

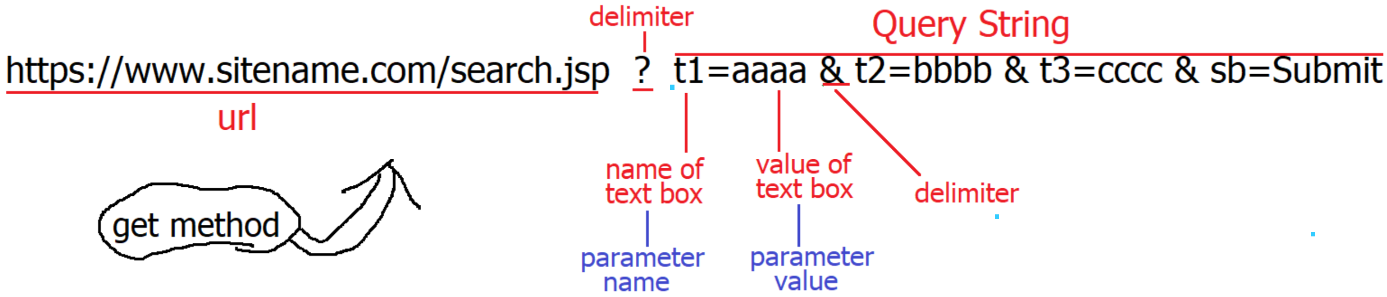
>outside the form, if we design anything that is not submitted to the server and without name attribute also not submittable.

>submit button value also sent to server

Syn:

<input type="submit" value="VAL" attributes>

<form action=”search.jsp”>



public String getParameter(String param)

String st = request.getParameter(“t1”)

post

https://www.sitename.com/search.jsp

image button

> "image submit button" is used to submit a form to the server.

> when the user clicks on the image button, browser is submitting data with x-co & y-co of image button

Syn:

<input type="image" src="filename" …/>

working with html forms

application forms => white paper

> form is collection/group of html input elements

> by using <form> tag we can create html form(s) (mean application form)

> web document/page it contains only one body, but a body can contain multiple forms.

> forms are used to collect info from users(ex: signup page, singing page, user registration, product delivery info etc…)

info we collected by using some fields (created by tag) like text field, password, checkbox, radio button, combo-box, list box, date, submit button, etc…

> form is used to submit input values (user's data) to a server-side program.

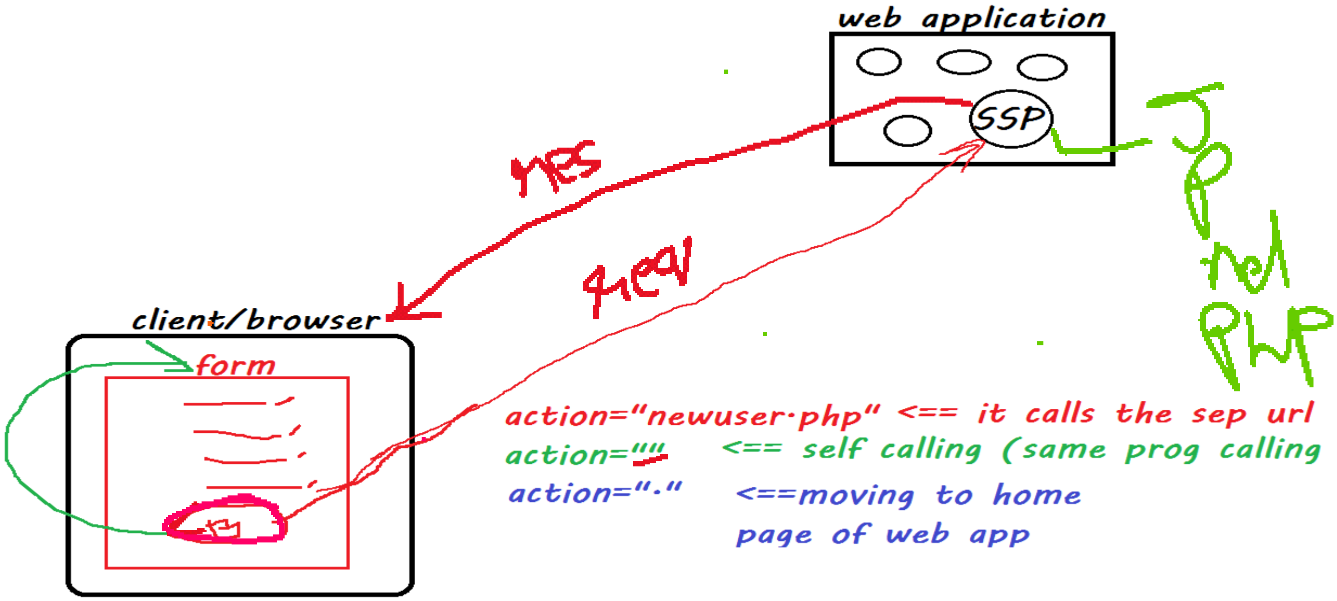
> form is a paired tag & block level

> form tag is sub tag of body tag

Syn: <form attributes>

UI designing

</form>



action="newuser.php" <== it calls the sep url

action="" <== self calling (same prog calling

action="." <==moving to home page of web app

attributes:

1 action :url represents destination program address or which program we want to call specify here

diff forms of urls:

https://www.irctc.co.in

https://www.sitename.com/login.class|login.aspx|login.php|login.ns|login.cgi|login.py

"" self-calling (its calling/sending data to current prog)

"." home of current application

2 method : it represents the way of sending data from client to server

it supports two ways, those are

get:

> it displaying data in address bar

https://www.redbus.in/search?fromCityName=Hyderabad&toCityName=Guntur&onward=10-Jan-2022&opId=0&busType=Any

> get method stores user inputs in browser history

> it's less secured

> get is a default method

> we can bookmark these pages

> get methods max data limit is 5.7kb

> get is faster than post

use-case's ⍰ search page, retrieving data from db, …

Post:

> it not displaying any data in address bar

https://www.redbus.in/search

https://www.irctc.co.in/nget/booking/train-list

> post method doesn’t store user inputs in browser history

> it's more secured

> we can't bookmark these pages

> post method data no limit

> post method slower than the get

use-case's ⍰ login page, sign-up page, registration pages, …

3 target : its rep where to open destination page,

_self, _blank, parent, framename ..

default is _self

4 enctype : it rep in which format we are sending data to server

html support two types, those are

>application/x-www-form-urlencoded

If you want to send data to the server without attachment and file uploading use this method.

It is default option

>multipart/form-data

if u want send data to server with attachment and file uploading

5 autocomplete : it automatically saves data while typing in the UI (textbox, password, address, pincode, ...)

on/off

on is default

6 novalidate : while submitting html perform some basic validation, if u don't do those validations, switch off this.

check box

>check boxes are used to allow the user to select some options, for example product, class, color selection, sport selection, select branch, select collage etc.

>whenever we want to select more than one option use check box's

>if the checkbox is selected/checked it returns "on'' (true) value, if the checkbox is unchecked it returns "off" (false) value.

checked➔ on (SSP), true (CSP)
un checked ➔ off (SSP), false(CSP)

100cb => same purpose => 100cb name should be same

100cb =>20cb =>purpose games => 20 name should be games

30cb =>purpose sports => 30cb name is sports

50cb => name3

SYn:

<input type="checkbox" attributes>

Note: all check boxes should be created with the same name.

"checked" attribute of the check box makes the checkbox by default checked, while opening the page.

Syn: checked="checked" before html5 ver

checked since html5 ver

radio button

> RB is used to display two or more options to the user, but allows the user to select any one of them.

SYn:

<input type="radio" attributes>

Note:

> all RBs should be created with the same name.

> "checked" attribute of RB makes the RB by default selected, while opening the page.

file

>file used to upload or attachment

>we can upload or attach any type of file, but @time a only one file

Syn   <input type="file" attributes>

attributes:

  multiple => it allows multiple to upload @time

  accept => filtering type of file

Note:

while uploading file method should be "post" and enctype is "multipart/form-data"

color

>used to select color by user, selected color we can apply on any control using JS

Syn: <input type="color">

date & time controls

date

>used to create a date box (date picker/popup calendar), where the user can select a date.

>the browser by default provides a built-in date picker.

Syn: <input type="date" attributes>

time

>used to create a time box, where the user can enter/select time (in the form of hours, minutes and seconds)

Syn: <input type="time" attributes>

datetime

>used to create a date-cum-time control

Syn: <input type="datetime-local" attributes>

month

>used to create a month box, where the user can select a month.

Syn: <input type="month" ...>

week

>used to create a week box, where the user can select a week.

Syn: <input type="week" ...>

email

>used to create an email textbox, where the user can enter a valid email id only.

>it displays an error message automatically (built-in validation), if the user enters other than email id (should contain @ and .).

Syn: <input type="email" attributes>

url

>used to create an url box, where the user can enter a valid url for downloading files or playing videos.

>it displays an error message automatically (built-in validation), if the user enters other than url.

Syn: <input type="url" attributes>

range

>used to create a slider bar, based on the specific range.

>this control req min value and max value, if we are not specified then browser takes default values.

Syn: <input type="range" attributes>

attributes

min => it sets min value of slider

max => it sets max value of slider

step => it sets increment value

value => sets indicator init position

appearance: slider-vertical;

search

> used to create a search box, where the user can enter some search text, it also displays a clear button to clear the text inside the search box.

Syn: <input type="search" ... >

dropdown (combo box) control

>using the "select" tag we can create a dropdown list/CB.

>dropdown list is used to display some options/items to the user and allows the user to select any one of them.

>"option" sub tag of "select" tag.

> "option" tag used to create/to add items/options to dropdown list.

> both are paired tags.

> select is inline element and option is block level element

Syn: <select attributes>

<option attributes> Text </option>

<option> Text </option>

```
      ...
    </select>
```

new Syn:

```
<option> Text </option>
    <option value="ws" > text </option>
```

Note: by default, DDB is displaying 1st added option/tiem

attributes:

selected : this attribute of "option" tag, used to change default selected option/item of DDB

<u>list box</u>

>using the "select" tag we can create a list box.

>list box is used to display some options/items to the user and allows the user to select any one of them (by default).

> "option" sub tag of "select" tag.

> "option" tag used to create/to add items/options to the list box.

> "size" attribute used to change DDB into List box.

```
Syn:<select size="N" attributes>
<option attributes> Text </option>
<option> Text </option>
<option> Text </option>
        ...
     </select>
```

Note: by default DDB is displaying 1st added option

attributes:

selected : this attribute of "option" tag, used to change default selected option/item of DDB

multiple :it allows user to select more than one option @time

size :attribute used to change DDB into List box and no.of options to display @time

<u>option groups</u>

> "optgroup" tag is used to group-up some options/items inside the "select" tag.

> one "select" tag can contain many "optgroup" tags, the "optgroup" tag contains many "options".

>its paired tag

> "optgroup" tag is the sub tag of "select" tag.

Syn:

```
        <select>
                <optgroup label="Text">
                <option> ..</option>
                <option> ..</option>
                <option> ..</option>
        </optgroup>
        <optgroup>
                        ...
        </optgroup>
</select>
```

textarea

>"textarea" tag is used to create a multi-line textbox.

use case: comments, address, feedback, delivery instr, ...

>its paired tag& inline tag

>it creates a multi-line text box, with default sizes (2row & 20 col)

Syn: <textarea attributes>

init value

</textarea>

attributes:

rows : it rep no.of lines to display @time, if more lines of data entered

automatically scrollbar is activated.

cols : it rep no.of chars per line

maxlength : it rep total no.of chars allowed in textbox.

Note: user can resize the textarea, at runtime in the browser.

progress bar

> "progress" tag is used to display the progress of a task.

> to move progress bar dynamically, we have to use "JS"

> it is a paired tag.

SYn: <progress attributes></progress>

attributes:

min max value

label

> label tag is used to create heading/prompting messages for elements or controls.

> label providing description for controls, it gives an idea to user what we have to type.

> when the user clicks on the label, the cursor will appear in the associated control automatically.

>its paired tag.

SYn: <label attributes> text </label>

attributes

for : used to specify the id of the control that is associated with the control

Note: labels are not sent to the server while submitting the form.

HTML5 tags

Datalist

Syn: <datalist attributes>

<option>text</option>

OR

<option value=”text”>

….

</datalist>

Mapping: **<**input type=”text” list=”datalist-id”/>

output tag

> this tag used to print data/output value on webpage.

> it is paired & inline element.

Syn <output attributes> text </output>

details and summary

>details and summary tags are used to allow the user to expand/collapse some information, when the user clicks on the heading.

>both are paired& block level

> details tag is the main tag and summary is the sub tag of details tag.

Syn:

```
        <details>
                <summary>Short info</summary>
  Detailed Information
</details>
```

header

>"header" tag represents header bar, which may include website logo, search box, main links, etc…

>it doesn’t provide any styles by default; we have to apply styles manually, using CSS.

>its paired tag.

```
Syn: <header>
                place header content here
        </header>
```

nav

>"nav" tag represents navigation bar, which may include top navigation menus.

>it doesn’t provide any styles by default; we have to apply styles manually, using CSS.

>its paired tag.

```
Syn: <nav>
```

place menus/links here

</nav>

section

>"section" tag represents a specific section of the page(box or container), which may include main-content/information.

>it doesn't provide any styles by default; we have to apply styles manually, using CSS.

>its paired tag.

```
Syn: <section>

       place main content here

     </section>
```

footer

>"footer" tag represents the footer part of the web page, which may include information of contact, faqs, location, copyrights, etc…

>it doesn’t provide any styles by default; we have to apply styles manually, using CSS.

>its paired tag.

```
Syn: <footer>
          place footer content here
        </footer>
```

<u>aside</u>

>"aside" tag represents the "right-side" part of the web page, which may contain ads/other promotional information.

>it doesn’t provide any styles by default; we have to apply styles manually, using CSS.

>its paired tag.

```
Syn: <aside>
          place ads/extra info here
        </aside>
```

article

element specifies independent, self-contained content.

- Newspaper articles
- Forum posts
- Blog posts
- User comments

```
Syn: <article>

          place ads/extra info here

        </article>
```

figure

The <figure> element is used for indicating self-contained content. The tag can include images, diagrams, illustrations, code examples, etc.

Syn:

```
        <figure>

                <img …/>

        </figure>
```

figcaption

The <figcaption> element is used for adding signature or annotation to the <figure> tag.

```
Syn: <figure>
          Image
          <figcaption> caption/text </figcaption>
     </figure>
```

mark

The <mark> element is used to mark a part of the text which has relevance. It can be used to highlight a text for showing emphasis, highlight search terms in search results to provide context, or distinguish a new content added by the user by showing it differently

Paired tag & inline

```
Syn:      <mark> text </mark>
```

bdi

The <bdi> element is used to isolate bidirectional text when a language with a right-to-left directionality, such as Arabic or Hebrew, is used inline with left-to-right languages

Paired tag & inline

Syn: <bdi> text </bdi>

base tag

this tag is used to specify a base URI or URL to use for all relative links contained within an HTML document.

Only one "base" element can be specified within a document, and it must be placed within "head" element.

We can also specify how other links should open using the target attribute.

Its unpaired tag

Syn: <base href="base-url" target="target"/>

Audio tag

The <audio> is one of the HTML5 elements added to allow embedding audio files to a web page. Since not all browsers support all audio formats, the audio file is encoded using special codecs.

The **<source>** tag or the src attribute is used to indicate the variations of the same audio file. The path to an audio file can contain absolute or relative URLs.

Syn:-

```
<audio attributes></audio>

    OR

<audio attributes>

    <source src="filename" type="audio/type">

</audio>
```

Attribute   Definition

src   URL => Specifies the path to the audio file.

controls Displays the control panel (start button, scroll, volume control).

If the controls attribute is missing, the audio file will not be displayed on the page.

autoplay Plays the audio file automatically after loading the page.

loop Repeat the audio file from the beginning after its completion.

muted Mutes the sound when the audio file is played.

Video tag

The <video> is one of the HTML5 elements added to allow embedding video files to a web page. Since not all browsers support all audio formats, the audio file is encoded using special codecs.

The <source> tag or the src attribute is used to indicate the variations of the same audio file. The path to an audio file can contain absolute or relative URLs.

Syn:-

<video attributes></video>

```
        OR
<video attributes>
        <source src="filename" type="video/mp4">
        ...
</video>
```

Attribute **Definition**

src URL => Specifies the path to the video file.

controls Displays the control panel (start button, scroll, volume control).

If the controls attribute is missing, the video file will not be

displayed on the page.

autoplay Plays the audio file automatically after loading the page.

loop Repeats continuously the audio file from the beginning after its

completion.

muted	Mutes the sound when the audio file is played.
width	width of video player
height	height of video player
poster	to set wall poster/paper of video

frameset & frame

iframe

> "iframe" stand for inline-frame, html5 rel tag

> placing one webpage result within another webpage.

> its paired tag& inline tag

Syn:

```
<iframe attributes>
</iframe>
```

att:

src => which page we want to include

srcdoc => it displaying text

width =>

height =>

frameborder => disable/enable border

map tag

> this is used to map an img to other images or webpages.

> map tag used to explore internal details about par area of img

> area is sub tag map tag

> area tag used to create mapping co-ordi

> map tag should link img tag ny using "usemap" attribute of img

> map is paired and area is non-paired

Syn:

<imgsrc="image">

<map name="">

```
        <area shape="" href="" coords="" alt="">
        <area shape="" href="" coords="" alt="">
        ...
</map>
```

Shape: it rep mapping area shape.

it sup 3 shapes, those are

> circle

>rect

> poly

Hrefits destination url or image

coords   shape x and y co-ords

https://www.image-map.net/

<u>marquee tag</u>

> used to move text/element in different directions

> its deprecated tag.

> its paired tag

Syn:

<marquee attributes>text|img| component </marquee>

attributes:

- direction => down, up, left (default), right
- loop => continues moving (default)
- scrollamount => 6
- scrolldelay => 84ms
- behavior => alternate

meta

> data about data means we provide some info/details about a webpage.

> meta is unpaired

> meta is subtag of <head> tag

> by using meta we provide info to browser, search engines, users/programmers.

Define keywords for search engines:

```
<meta name="keywords" content="HTML, CSS, JavaScript">
```

Define a description of your web page:

```
<meta name="description" content="HTML and CSStutotrial">
```

Define the author of a page:

```
<meta name="author" content="SK">
```

Refresh document:

```
<meta name="refresh" content="45">

<meta http-equiv="refresh" content="time; url=URL">
```

Setting the viewport:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```