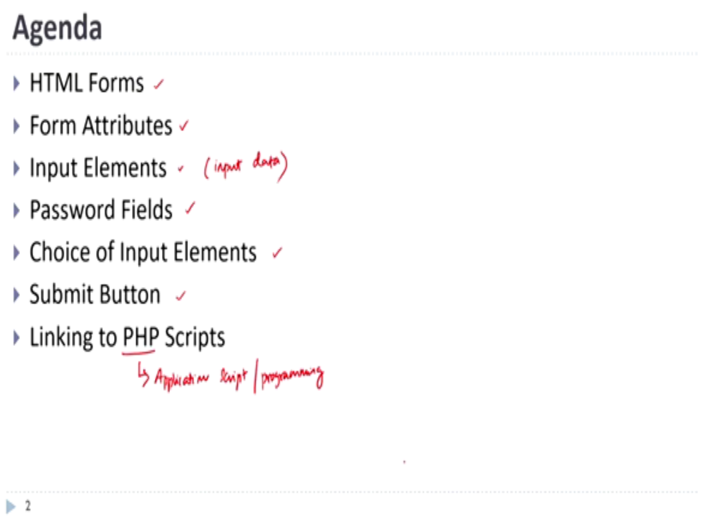
Computer Aided Decision Systems Industrial Practices using Big Analytics
Professor Deepu Philip
Department of Industrial and management Engineering
Indian Institute of Technology Kanpur
Professor Amandeep Singh
Imagineering Laboratory
Indian Institute of Technology Kanpur
Lecture 44
HTML as a User Interface Language 2

Good evening everyone. Welcome to yet another lecture of the Web-based Decision Support System for practitioners and decision makers. And, we are mostly focusing on business decisions as we have seen in the previous lectures. And, we are looking into different aspects of the components of the Decision Support System and also different case studies of Decision Support Systems.

So, today we are getting into what we call the usage of Hypertext Markup Language (HTML) as a User Interface. We have already seen the basics of HTML and some of the aspects of HTML. And now, we are going to move forward and see what are the basic aspects and some advantages of HTML as a UI language. So, today's lecture title is HTML as a User Interface language. It is part two of the lecture. I am Deepu Philip from IIT Kanpur.

(Refer Slide Time: 01:05)



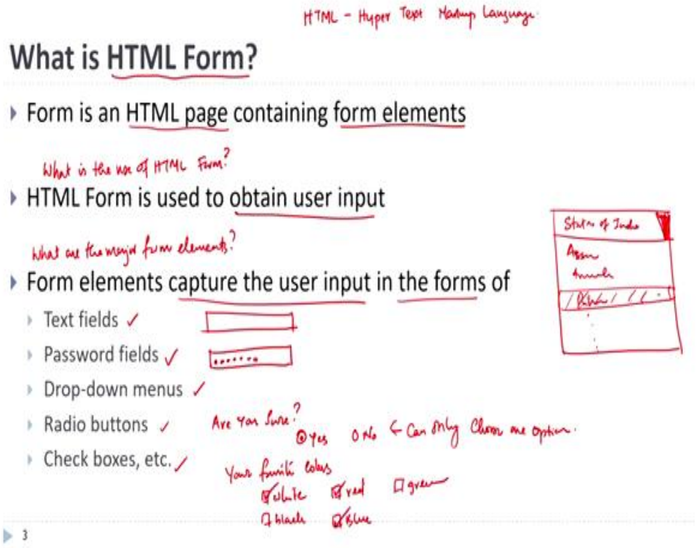
Agenda

- HTML Forms ✓
- Form Attributes ✓
- Input Elements ✓ (input data)
- Password Fields ✓
- Choice of Input Elements ✓
- Submit Button ✓
- Linking to PHP Scripts
  ↳ Application Script / programming

2

So, today's agenda is mostly HTML Forms. So, what are our HTML Forms? What are the attributes? We have seen the tag attributes of HTML, now, we are talking about the form attributes of HTML. There are certain input elements, so, this is to input data, how do you input data into the system, that one, then, we also talk about what is Password field, and how do you choose input elements, the Submit Button and how do you connect it with the PHP script. So, PHP is what we are going to use in this class as the application scripting or programming. So, the application layer is done with the help of PHP in this course.

(Refer Slide Time: 01:56)



What is HTML Form?

- Form is an HTML page containing form elements
- HTML Form is used to obtain user input
- Form elements capture the user input in the forms of
  - Text fields ✓
  - Password fields ✓
  - Drop-down menus ✓
  - Radio buttons ✓
  - Check boxes, etc. ✓

3

HTML - Hyper Text Markup Language
What is the use of HTML Form?
What are the major form elements?
••••••
States of India
Assam
Arunachal
/ Bihar / / /
Are You Sure?
● Yes ○ No ← Can only Choose one option.
Your favorite colors
☑ white ☑ red □ green
□ black ☑ blue

So, we know what an HTML stands for. So, just to remind you HTML stands for Hypertext Markup Language. So, what is markup and how do we use tags and everything, so, the question is, what is HTML Form? What is the form in this?

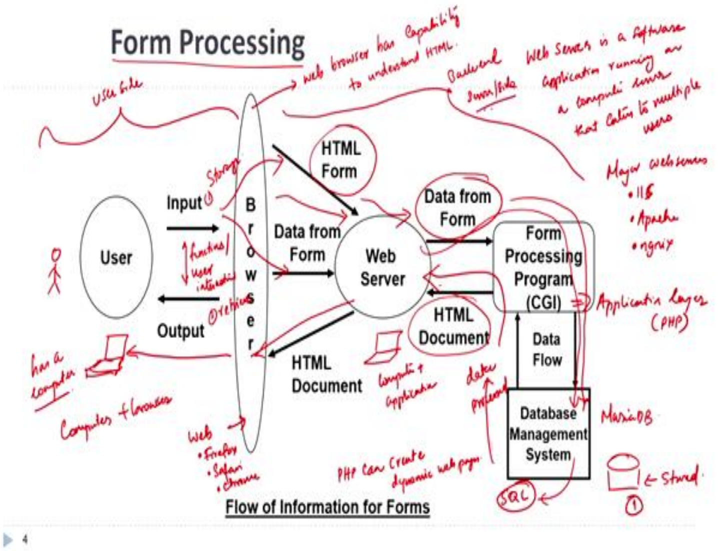
- So, form is a HTML page, HTML web page which contains form elements and there are so many form elements. What are form elements, we will discuss.
- And, what is the use of HTML Form elements? What is it used for? What is the use of HTML Form? The form is to obtain user input and capture the user input.
- What are the major form elements? So, the purpose of the form elements as we said their layer is to capture the user input from the HTML page and it uses one is, text fields. It also uses Password Fields, drop down menus.

So, a Text field is something like this where you type your text. A Password Field is also a Text field. But whatever you type shows in like black dots, so, you do not get to see what is being typed. Drop down menus are pretty much shown something like this.

So, the states of India and you have like an arrow down here and you have Assam, Arunachal, Bihar, like this and you move your mouse around and whatever it is, you get it clicked then, selected then, it gets selected. That is, what you call a drop down.

Radio Buttons are like, are you sure and then, you have a yes or a no. And, if you click this then, black dot appears here. If you click this then, the no appears, you cannot select both of them at the same time, you can can only choose one option. And, whereas Checkboxes are your favorite color, or colors, and you have white, black, red, blue, green, etcetera, and you can take a couple of them and then, you can have multiple selections as part of it. So, that is what he called a Checkbox.

(Refer Slide Time: 04:57)



Form Processing

Flow of Information for Forms

So, this is important and how HTML Form is being processed. And, there is a user. So, in this case, you can think about it as a user. There is a human being sitting here. And, the user has a computer terminal, it can be a laptop, it can be a desktop, it does not matter. That is a computer on which the user is able to access our browser. And, the browser is a Web Browser. And, it can be Firefox, it can be Safari, it can be Google Chrome, can be Internet Explorer, whatever it is, it can be any one of them. So, the computer has a Web Browser.

So, the Web Browser has capability to understand HTML, whatever the markup language you say, it has the capability to understand. So, the user uses a computer. And, with the computer, there is a browser, and you open the browser and conduct the input and the output functions. These are the functions or user interactions. Either the functions or the user interactions are done with the help of a browser, so it is a computer plus browser.

You can also use the current word, you can extend it to a mobile and a browser also, nobody is going to question you about that. But that is the one part. So, that is the user side of it. So, this whole thing that we talk about is the user side, or the front side of it.

So, then, the input by the user uses the browser and it goes into this channel, what we call as the HTML Form. So, using the HTML Form and the data from the HTML Form, it goes into something called a Web Server. This is another computer.

So, what is a Web Server? A Web Server is a software application running on a computer server that caters to multiple uses. So, what happens is, there is a computer and then, there is an application. So, there is a computer Plus application.

So, the major Web Servers, I think we have discussed already in the class. Microsoft was in the IIS (Internet Information System) Linux OS, Apache Tomcat, the most popular one, then there is ngnix, I believe. So, there are a couple of those Web Servers available.

So, what Web Servers are capable of doing is, they can capture the data from the HTML Form, which is inputted by the user in a user terminal, that data gets transmitted to the through from the browser through the internet comes to this computer. And, in this computer, this data from the form is aggregated. And, this one, this form gets processed, it is CGI, it is basically an application program, application layer, you can call in this case we are using PHP at this one. And, this data using this PHP, comes to the DBMS. Now, DBMS, it is Maria DB.

So, whatever the user on the computer that he types, using the HTML Form, the Web Server captures the data. And, the data from the form is taken and passed through the PHP, which is the CGI portion in this case, and then, that goes into what you call as the Database Management System. And, you have the hard disk here, it gets stored.

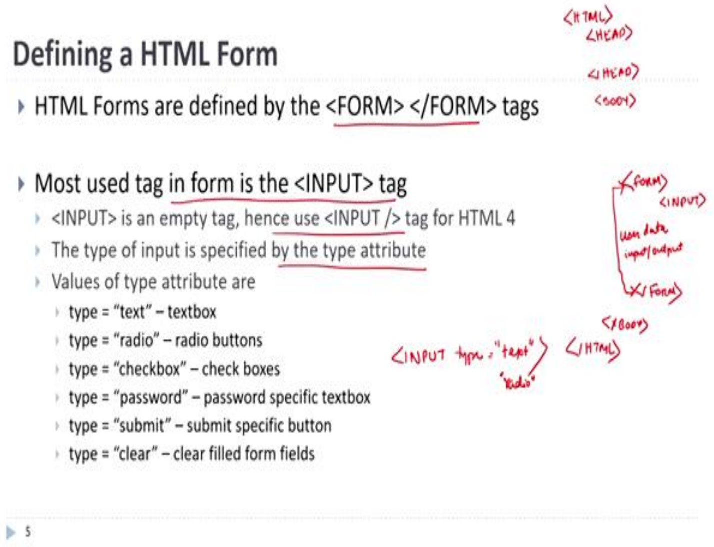
So, also in a similar way instead of this one, the user said I do not want to store data, I want to retrieve data. So, what does the user say.? The user says the same information. So, part one is storage, storage happens. Part two is, the second option is to retrieve. I want to see what data is stored.

So, the same way he uses the HTML Form, communicates to the Web Server, the Web Server catches this information, goes through this one, comes to the DBMS and the DBMS runs the SQL. And, the data is captured from the stored data and the data is sent back, data processed. It is sent back to the Web Server through PHP. Come through this application program as part of this. And, in this process, when the data is sent from the PHP to the Web Server, it is sent as a HTML document. So, that is why PHP is capable of creating dynamic web pages, that is the PHP part. And then, the Web Server, once the dynamic HTML comes into the Web Server, then, the Web Server basically moves it to what you call as the browser, the HTML document which is being displayed by the DBMS (the Database Management System) is sent to the browser as an HTML document and it is then, taken as an output and display it on to the terminal of the user. So, the user gets to see it.

So, the query, the request for the data went through the browser to the Web Servers in the form of an HTML Form. And, from the data from the form went to the Database Management System, the query got run as part of this, and that data was processed. And, that processed data when goes back to PHP, PHP creates an HTML document out of it, gives it back to the Web Server, Apache or whatever it is, and Apache sends it to the browser as an HTML document with all styles formatting everything, and that browser displays the content to the user on the terminal of his or her computer.

So, this whole thing in the backside is the backend or people call it as a server side. So, this is the Form. Flow of Information the Forms actually gets processed as part of this exercise. Hope you guys understand this.

(Refer Slide Time: 12:08)



Defining a HTML Form

- HTML Forms are defined by the <FORM> </FORM> tags
- Most used tag in form is the <INPUT> tag
  - <INPUT> is an empty tag, hence use <INPUT /> tag for HTML 4
  - The type of input is specified by the type attribute
  - Values of type attribute are
    - type = "text" – textbox
    - type = "radio" – radio buttons
    - type = "checkbox" – check boxes
    - type = "password" – password specific textbox
    - type = "submit" – submit specific button
    - type = "clear" – clear filled form fields

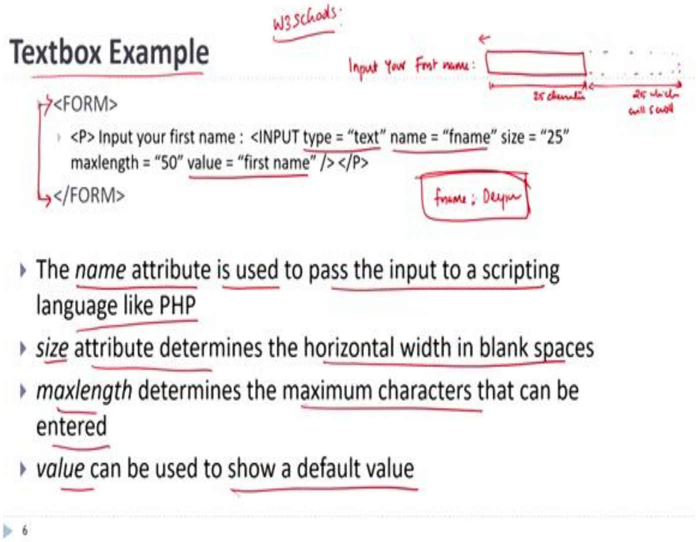
<HTML>
<HEAD>
</HEAD>
<BODY>
<FORM>
<INPUT>
user data
input/output
</FORM>
</BODY>
</HTML>
<INPUT type = "text"
"Radio"

5

So, how do you define an HTML Form?

- HTML Forms are defined by the form, slash form tags. So, as we said earlier, you have the <HTML>, </HTML> tags, and within that, you have <HEAD>, </HEAD> tags. And then, you have <BODY>, </BODY> tags. Within the body tags is where the form tag comes into picture, some people use a <FORM>, </FORM>. So, whatever comes here is the user data input output. So, whatever is enclosed in that, the minute HTML sees that a HTML set of a user is going to provide some data and I need to have a mechanism to capture it.
- The most use in a tag in the form element is the input tag, and input is an empty tag, empty tag means it does not have a pair, solely just one tag. So, typically, for HTML for onwards, use the input, slash option. And, the type of input, what you are going to talk about as part of it is the, it is specified by the type attribute of the input. So, if I say, <INPUT type = "text"> then, HTML understands, okay this guy is trying to create a textbox, where I can type 256 characters text as part of it. Instead of the text, if I type "radio" then, HTML knows it is going to be a Radio Button. It can be a Checkbox, Submit, clear, etcetera, all these kinds of things as part of the input. And, it does not have to be a Textbox, it can also use the input to create what we call a button, and all those kinds of things are also part of this. So, the input, whatever it is, comes under the format attribute, and then, you specify what type of input you want, using the type attribute of the input. So, that is the major form aspect.

(Refer Slide Time: 14:17)



Textbox Example

```
<FORM>
  <P> Input your first name :  <INPUT type = "text" name = "fname" size = "25"
  maxlength = "50" value = "first name" /> </P>
</FORM>
```

- The *name* attribute is used to pass the input to a scripting language like PHP
- *size* attribute determines the horizontal width in blank spaces
- *maxlength* determines the maximum characters that can be entered
- *value* can be used to show a default value

6

W3Schools:
Input Your First name:
25 characters
25 which will scroll
fname : Deepa

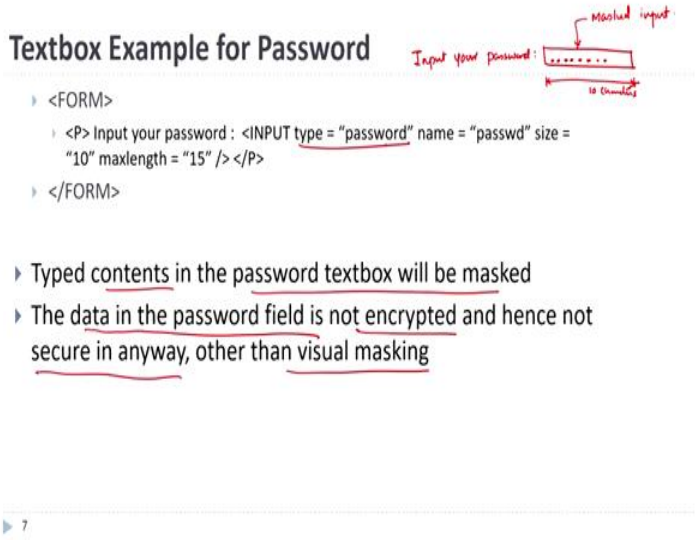
So, here is an example of a Textbox. So, the Textbook Example basically shows this and by the way, remember, go to W3 schools, the open free web page, where we can actually do a lot of these forms. Create these ones and try your best, as part of it. So, please go through it, this web page and try a website and online free tutorial for you.

So, you can see this is the <FORM>, </FORM> tax, which is the payer of the tax from there, the paragraph as we say <P> Input your first name: so that is shown as a text and then, input type. So, this will be shown something like this. It will input your first name, column, and it says input type is text. The name is F, size is 25. So, it takes a Textbox and draws like this, and this size is 25 characters. But it can have a max length of 50. That means there will be another 25, which will scroll, scroll to accommodate, the max length of 50. And, the value of this is the first name. So, one of the critical aspects is, you need: the type, the name, the way to identify what a textbox is, and what the value of that textbox is, pretty much. So, we distinguish the user input. Whatever the user input as part of the HTML Form.

- The name attribute is what we use to pass the input to a scripting language like PHP. So, it will actually say when it parses PHP, it will say that F name, and they might say the matter outside is this person type it as deep or something like that. So, this pair, the name of the Textbox, and the value associated with it, goes into languages like PHP, and then, PHP parses and sends it to the database.

- The size attribute, size determines the horizontal width in blank spaces, whatever be the size, you say, this is the horizontal width, the 25 characters that I showed you.
- The max length that to be determined, the maximum characters that can be entered. So, if I say 25, and max length is 25, everything you type, you do not get to scroll. So, if you type the 26th character, the first character will move to the left, and then, you can keep on typing, and the whole thing will scroll accordingly.
- And, value is used to show a default value. If you do not type anything, it will just show the first name will be stored as what it will be there. If you do not use value also, it is not a big deal. But, the name of the field is very critical and the type also, whether it is a text, whether it is a password, etcetera, is quite critical as required, as part of this.

(Refer Slide Time: 17:19)



Textbox Example for Password

- <FORM>
  - <P> Input your password : <INPUT type = "password" name = "passwd" size = "10" maxlength = "15" /> </P>
- </FORM>

- Typed contents in the password textbox will be masked
- The data in the password field is not encrypted and hence not secure in anyway, other than visual masking

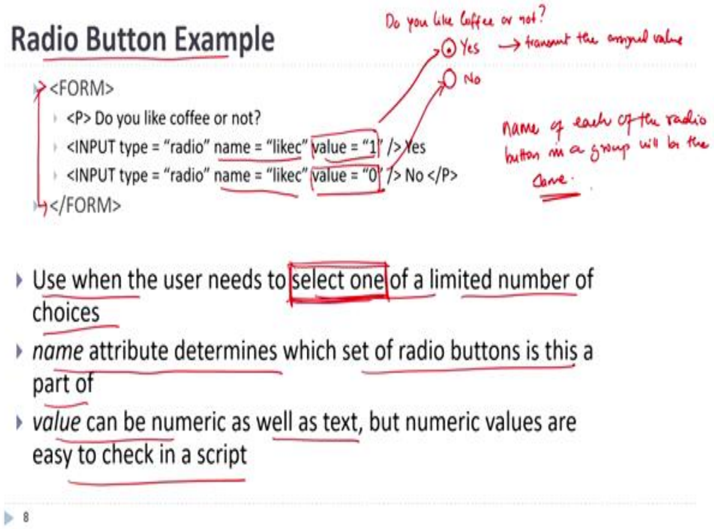
7

So, the same Textbox, if you want to create. So, in this type, it said input type is text, we change it into input type as ‘password’. So, then, immediately this whole thing that you saw there, it will show something like this, input your password and it will show you something like this. And, this size will be at this point 10 characters. The important aspect of this is whatever you type here, you do not get to see what it is typed, you will basically get to see black dots as part of it. So, it is masked, so this is a masked input, you do not get to see what the user is typing there. So, both are textboxes. Both the data entry and also whatever the password entry both are textboxes, except that the type here is if you type it as password then, HTML understands, hides the input. If the type is text, textual input does not matter.

- So, typed contents in the password Textbox will be masked, sizes just mentioned.
- So, the other thing that you need to remember is data in the Password Field is not encrypted, unless you are embedded, created or added an encryption in the system, it is not encrypted. So, all it does is, there is a visual masking, visually you cannot really see. But, otherwise, it is not secure.

You should remember that it is if you encrypt it, then, it is protected and the people cannot figure out what is being typed and those kinds of stuff. But, in a visual masking all you get to see somebody looking at your screen is not able to figure out what is the aspect there.

(Refer Slide Time: 19:03)



Radio Button Example

```
<FORM>
  <P> Do you like coffee or not?
  <INPUT type = "radio" name = "likec" value = "1" /> Yes
  <INPUT type = "radio" name = "likec" value = "0" /> No </P>
</FORM>
```

- Use when the user needs to select one of a limited number of choices
- *name* attribute determines which set of radio buttons is this a part of
- *value* can be numeric as well as text, but numeric values are easy to check in a script

Do you like Coffee or not?
● Yes → transmit the assigned value
○ No

name of each of the radio button in a group will be the same.

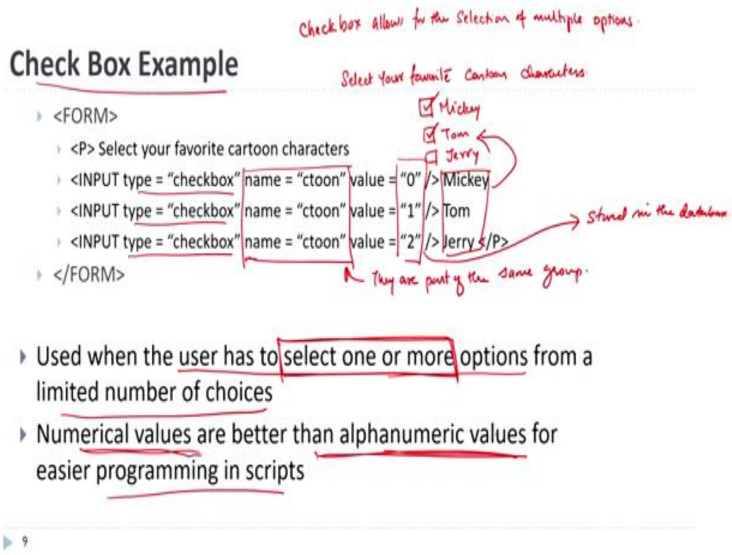
Then, comes the next one, what do we call a Radio Button example. So, the Radio Button example is again, it is a form like this between the form tags, you can see that and then, Do you like coffee or not? so, it will show something like this, Do you like coffee or not, so, this text is printed. Then, input type is radio and it says yes, so, it will actually do something like this, ☐Yes, ☐No. So, the value for this if you click this, it will transmit the assigned value. So, in this case, the assigned value for Yes is 1, for No is 0. So, if you look into, how it will be stored in the database, this coffee will be stored as 1s and 0s in the database. And then, when you read it back, you read 1 and 0 and understand whether it is yes or no and then, put the data back in it.

So, when do you use the Radio Button?

- The Radio Button is used when the user needs to select one from a limited number of choices, he is to select one of a limited number of choices, you do not have huge expanded list, it is a limited number of choices from there, you have to select one, the key word is this, select one, you have to make one and only one selection.
- The name attribute of the Radio Button determines which group of the Radio Button it is part of. So, if you look into this, each one of the Radio Buttons will have different values associated with it, but the name of all of them will be the same. So, the name of each of the Radio Buttons in a group will be the same, that you need to remember. So, you say that likec Yes, likec No, then no, it would not work. So, when you say likec, likec, likec, there are let say 10 options and 10 like Cs, that means it will allow you to only pick one out of the options available to you.
- The value can be numeric, or text, but the numeric values are easy to check in a script. Most of the time, you will see people using 1s and 0s because they are very easy to check and store instead of writing the alphanumeric yes and no, because then, you are to tree transcript part of it. So, the program is going to use numeric values. Some of the people say you should store yes or no, the alphanumeric value or whatever it is.

Now, the next Checkbox example, the next one, so, as we just saw before is the Radio Button, so, here you are selecting exactly one from a limited number of choices. And, all the Radio Buttons in a particular set have the same name, this is what we mentioned.

(Refer Slide Time: 22:15)



Check box allows for the Selection of multiple options.

Check Box Example

Select Your favorite Cartoon characters

☑ Mickey
☑ Tom
☐ Jerry

```
<FORM>
  <P> Select your favorite cartoon characters
  <INPUT type = "checkbox" name = "ctoon" value = "0" /> Mickey
  <INPUT type = "checkbox" name = "ctoon" value = "1" /> Tom
  <INPUT type = "checkbox" name = "ctoon" value = "2" /> Jerry </P>
</FORM>
```

Stored in the database

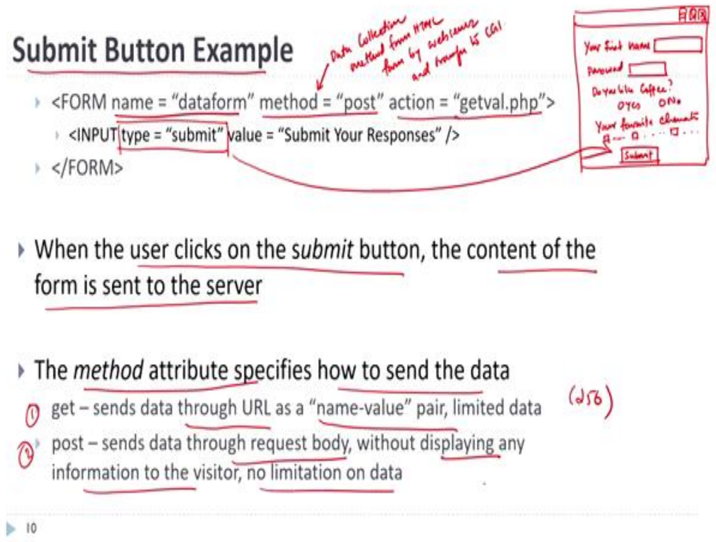
They are part of the same group.

- Used when the user has to select one or more options from a limited number of choices
- Numerical values are better than alphanumeric values for easier programming in scripts

Whereas in the time of Checkbox, the only difference between the Checkbox is, it allows for the selection of multiple options. You can select multiple options as part of it. So, here is what your favorite character is. So, the input will be select your favorite cartoon characters then, it will allow something like ☐Mickey, ☐Tom, ☐Jerry. So, the idea is that you can pick Mickey and Tom or you can pick Mickey and Jerry or whatever it is. So, the important thing is, you can see that all these Checkboxes have the same name. So, that means they are part of the same group whatever you see here is what gets displayed here. This value, you do not get to see that only gets stored into the database, So, this is what gets stored in the database. So, that is this part. And then, you can see that the input type is Checkbox, compared to the previous one, where we said the type is radio. The radio is selected as one of the limited options.

- So, in a Checkbox, the user can select one or more options from a limited number of choices. So, you can have one or you can select more. So, the keyword is, select one or more. You can choose one or more as part of this.
- So, the numerical values are better than alphanumeric values because it is easy to program. As of now, it was an old school concept purely because of the fact that alphanumeric values, the checking, and all those kinds of things were increasing the programming overheads. So, people have turned to prefer numerical values but recently people do not really care about. So, they can write Mickey, Tom and Jerry and then, that value can be stored as such, whatever you want.

(Refer Slide Time: 24:42) 25:09R



Submit Button Example

Data collection method from HTML form by webserver and transfer to CGI

```
<FORM name = "dataform" method = "post" action = "getval.php">
    <INPUT type = "submit" value = "Submit Your Responses" />
</FORM>
```

- When the user clicks on the *submit* button, the content of the form is sent to the server
- The *method* attribute specifies how to send the data
  - ① get – sends data through URL as a "name-value" pair, limited data (256)
  - ② post – sends data through request body, without displaying any information to the visitor, no limitation on data

10

So, then, we talked about Input Password, Checkbox, Radio Buttons. Now, let us talk about what you called a Submit Button. When you have created all this data and stuff like that, how do you transmit this value? So, the Submit Button is somewhere here. It translates the value, the input from the HTML Form to the Web Server. Now, we have already just seen the form.

Now, we are giving the attributes to the form. So, first is the name of the form. So, that is the name of the data form. So, that means that particular web page from where the data is captured will be named as data form and then, there is something called a method equal to post. So, there are two ways actually, you can transmit data in an HTML Form.

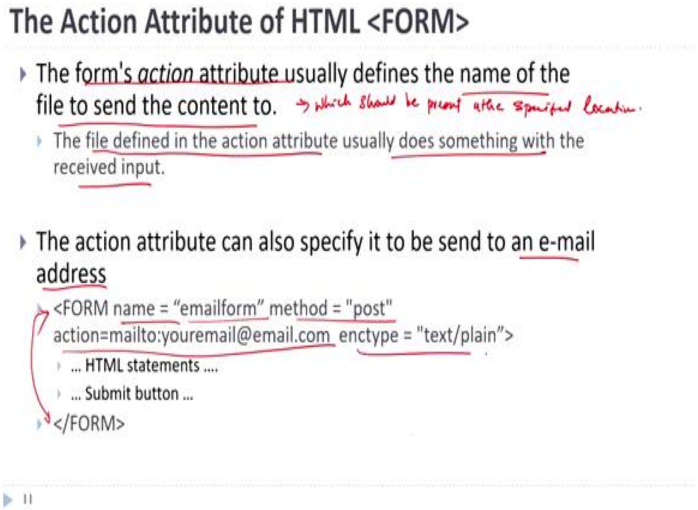
There is, let us say, here is just creating for a visual reference. Here is your first name, password, do you like coffee, yes, no, then, your favorite characters as Mickey, Minnie, Tom, Jerry, etcetera. And then, once you have filled all these things, you need to tell the HTML that I am ready to submit this data. I am willing to transfer this data from HTML Form to the Web Server, so that you can store it in this. When you submit it, how does that data get collected, that is called the data collection method from HTML Form by Web Server my ware and transferred to CGI, in this case, a CGI is getval.php action.

So, when you say when somebody clicks the Submit Button, what action did you take, you call the file called getval.php and using the post method, transfer the values, whatever the values user have inputted in this form and transmitted to it. So, this Submit Button, the type equal to submit, basically creates a button at the end of it.

- So, when the user clicks the Submit Button, the content of the form is sent to the server, whatever is the content it goes into the server.
- The method attribute has specifically the method attribute specifies how the data should be sent, what is the mechanism of sending the data and there are ideally two mechanisms one is the get and another is the post. So, the get, basically, sent the data through the URL, the data URL is used name, value, pair, and you get this limit, there is something like 256 characters you can send as part of this. So, it is sent through the URL. It is an unsafe mechanism, because anybody who sees the URL will be able to find or identify what the data is. The post, on the other hand, sends the data through the request body, without displaying any information to the visitor and there is no limitation on data.

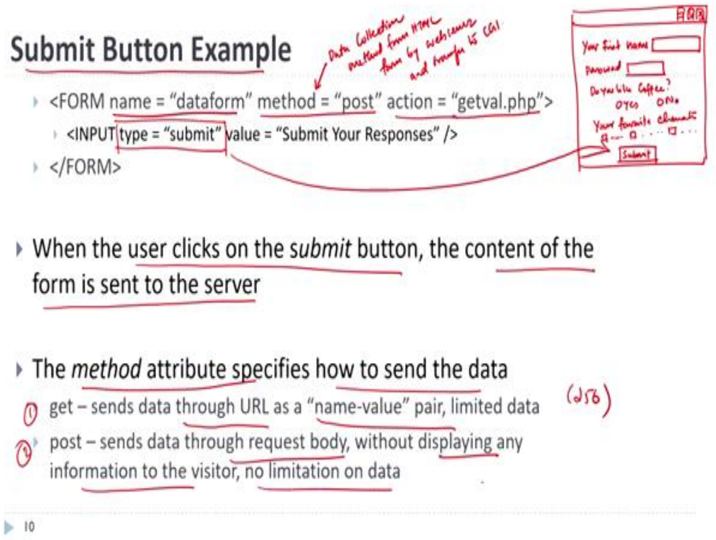
So, post is more preferred over the get, purely because of the fact that you can send a more massive amount of data and the user does not get to see it unlike in the get, where it is get, you get to see through the URL of the system.

(Refer Slide Time: 28:46)



The Action Attribute of HTML <FORM>

- The form's *action* attribute usually defines the name of the file to send the content to. → which should be present at the specified location.
  - The file defined in the action attribute usually does something with the received input.
- The action attribute can also specify it to be send to an e-mail address
  - <FORM name = “emailform” method = "post" action=mailto:youremail@email.com enctype = "text/plain”>
    - ... HTML statements ....
    - ... Submit button ...
  - </FORM>



Submit Button Example

```
<FORM name = “dataform” method = “post” action = “getval.php”>
    <INPUT type = “submit” value = “Submit Your Responses” />
</FORM>
```

- When the user clicks on the *submit* button, the content of the form is sent to the server
- The *method* attribute specifies how to send the data
  - get – sends data through URL as a “name-value” pair, limited data
  - post – sends data through request body, without displaying any information to the visitor, no limitation on data

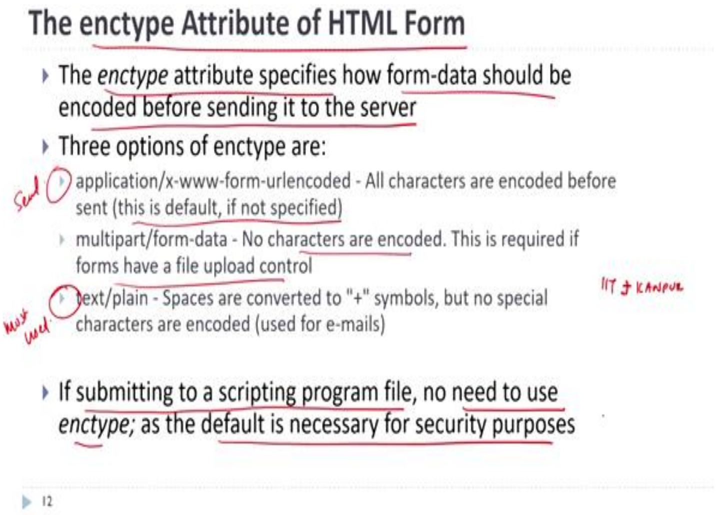
10

So, then, the Action Attribute. This is the next one we need to look into.

- So, the forms Action Attribute, usually defines the name of the file to send the content. Most of the time, the majority of the time, it tells you which file should be present at the specified location. So, the file can be, if it just says getval.php within two chords, it tells the computer that in the current folder wherever this HTML pages, there is another script called getval.php. So, send the data to that. So, that is what the Action Attribute says.
- So, the file defined in the Action Attribute usually does something with the received input. So, what happens is the Web Server revokes that file, sends the data through the method that you have said and tells this, okay now you do whatever the hell you want to do. Action Attribute instead of sending it into a file, you can send it to a specified email address. If you want to do that, that also can be done.

So, the form name is email form and method is post action is basically mail to a specific email, and then, whatever the HTML stuff and Submit Button. So, what happens is, when you click the Submit Button, an email is created and sent to your email address. That is also done with the help of HTML Forms.

(Refer Slide Time: 30:13)



The enctype Attribute of HTML Form

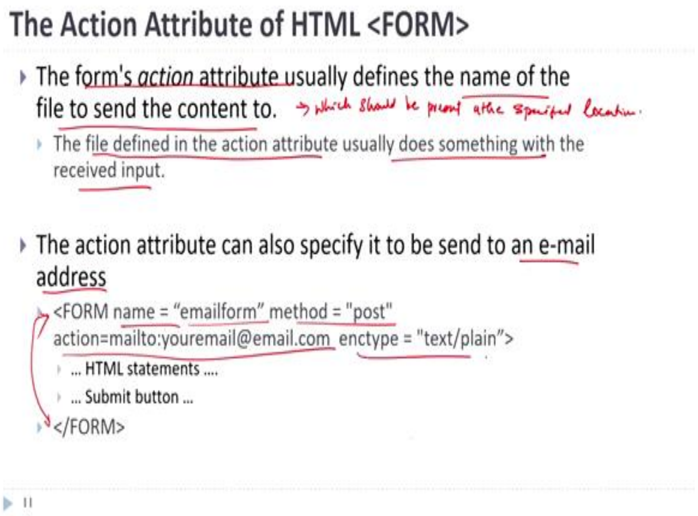
- The *enctype* attribute specifies how form-data should be encoded before sending it to the server
- Three options of enctype are:
  - application/x-www-form-urlencoded - All characters are encoded before sent (this is default, if not specified)
  - multipart/form-data - No characters are encoded. This is required if forms have a file upload control
  - text/plain - Spaces are converted to "+" symbols, but no special characters are encoded (used for e-mails)
- If submitting to a scripting program file, no need to use *enctype;* as the default is necessary for security purposes

Send

most used

IIT ‡ KANPUR

12



The Action Attribute of HTML <FORM>

- The form's *action* attribute usually defines the name of the file to send the content to. → which should be present at the specified location.
  - The file defined in the action attribute usually does something with the received input.
- The action attribute can also specify it to be send to an e-mail address
  - <FORM name = “emailform” method = "post" action=mailto:youremail@email.com enctype = "text/plain">
    - ... HTML statements ....
    - ... Submit button ...
  - </FORM>

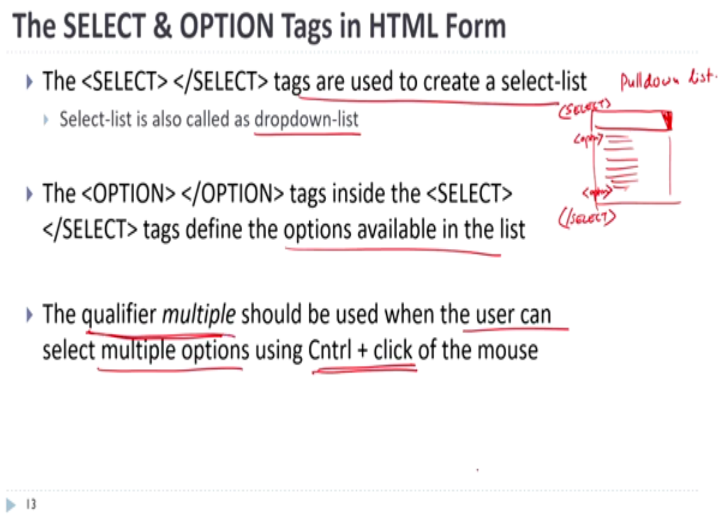
11

Now, the encryption type, so,

- The enctype Attribute of the HTML Form literally tells you this attribute specifies how the form data should be encoded before sending it to the server, so, the enctype is this part, what do we say. So, there are three options available for enctype. The number one is, the first one is I will go from the bottom, the most widely used is the:

- Text or the plain. So, the thing is the spaces, any blank spaces will be converted to plus symbols, but no special characters are encoded. So, if you have some whitespace, if you say IIT, space, Kanpur, the space will be done with the plus sign on.
- The URL encoded. And, all characters are encoded before being sent. And, this is default, if not specified. So, if you do not specify anything, then, this encoding happens by default.
- And, the last one is the multipart form data. So, no characters are encoded, this is recorded in the form of a file upload control. So, if you want to upload a file, you do not do encoding because that file will not get uploaded properly. So, you basically have to do multipart form data so that the file upload can work.
- If you are submitting to a scripting program file, if you are sending it to a program file, which is basically like a PHP, what we just showed, you do not need to use the enctype, because the default is necessary for security purposes. So, that is what happens and part of this.

(Refer Slide Time: 31:47)



The SELECT & OPTION Tags in HTML Form

- The <SELECT> </SELECT> tags are used to create a select-list
  - Select-list is also called as dropdown-list
- The <OPTION> </OPTION> tags inside the <SELECT> </SELECT> tags define the options available in the list
- The qualifier *multiple* should be used when the user can select multiple options using Cntrl + click of the mouse

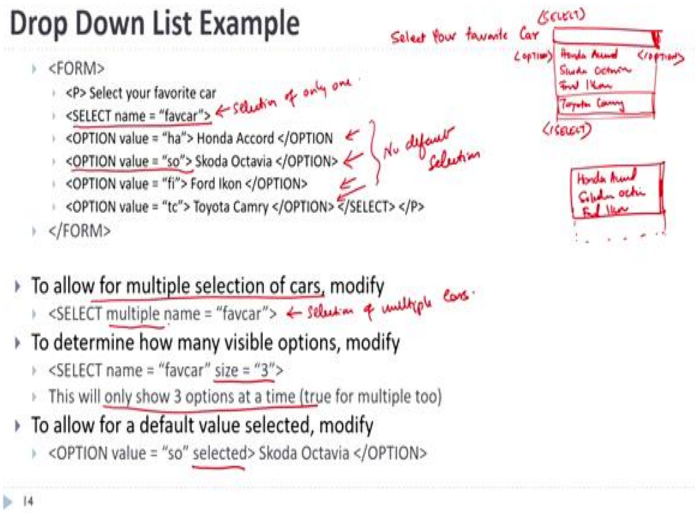
Pulldown list.
<SELECT>
<opt>
<opt>
</SELECT>
13

And, the Select option tag of the HTML Form is what we will get into now.

- And, the <SELECT> tags are used to create something called a Select List. The most popular name for this is called a drop down list or other name for it is pull down list. So, it is something that the states of India if you want something it is a textbox like this with an arrow at the bottom and the minute you click it the list comes out and then, you can choose from the list. That is what we call the select list or the drop down list or the pull down list.

- And, the <OPTION> tags within the select tags define the options that are available in the list. So, the beginning and ending is a select tag and then, the individual ones are taken care of by the option tag. And, so each option in the list is created by the option tag.
- The qualifier multiple should be used. When the user is allowed to choose multiple, more than one option which is accomplished with the help of a control button and click at the same time. Press the control button, click the mouse then, you can use it to do the multiple selections. This is only possible if the qualifier multiple is enabled and by default it is not enabled.

(Refer Slide Time: 33:11)



Drop Down List Example

- <FORM>
  - <P> Select your favorite car
  - <SELECT name = "favcar">
  - <OPTION value = "ha"> Honda Accord </OPTION
  - <OPTION value = "so"> Skoda Octavia </OPTION>
  - <OPTION value = "fi"> Ford Ikon </OPTION>
  - <OPTION value = "tc"> Toyota Camry </OPTION> </SELECT> </P>
- </FORM>

- To allow for multiple selection of cars, modify
  - <SELECT multiple name = "favcar">
- To determine how many visible options, modify
  - <SELECT name = "favcar" size = "3">
  - This will only show 3 options at a time (true for multiple too)
- To allow for a default value selected, modify
  - <OPTION value = "so" selected> Skoda Octavia </OPTION>

14

So, here is an example form to select your favorite car. So, it will appear something like this,

Select your favorite car- Honda Accord

Skoda Octavia

Ford Ikon

Toyota Camry

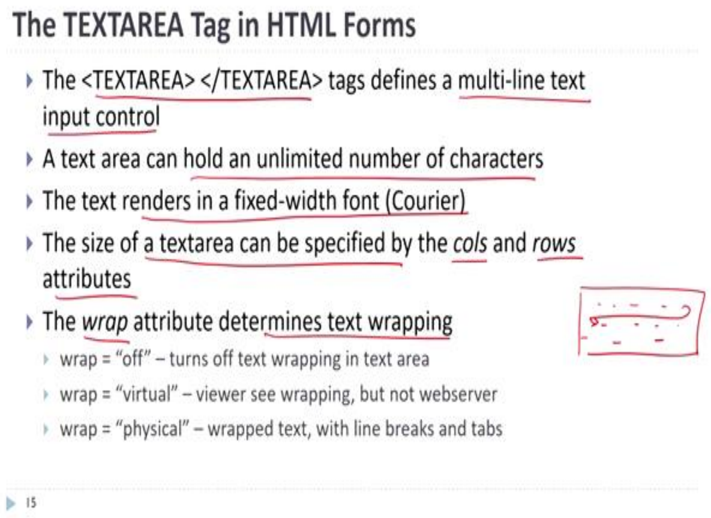
So, then, you can use the mouse to select whatever be the option that you want to select. So, each one of these is an option tag, each one. And, the pull down list is the select tag.

- So, if you want to allow multiple selections of cars to allow for multiple selection of cars, you change the select, name is the fav car, instead of this, this is the current one, so, it allows for select. Selection of only one. But, if you say name fav car, but add the keyword multiple, it allows you to select, selection of multiple cars. How many visible options we want to select, how many visible options, you can say modify size is equal to 3. And, it will only show three options at a time.

So then, what happens is, it will be the last one, you will see what we call as a, it will look like this and you will have a scroll bar right here. So, you can move it up and down, you will see a Honda Accord, Skoda Octavia, Ford icon, the fourth one Toyota Camry will be below this, you will only see that, and you will have to move the thing up and down to see what is the number of visible options there. And, this is true, immaterial or what is this. So, if you do not want the list to be very, very large then, that is what this is.

- So, if there is a specific default value to be selected, then, modify the option value. So, you can add a modifier here also, all of these have no default selection. Whereas, if you add the word selected, right after the name, option value name, then, immediately that means that will be selected by default.

(Refer Slide Time: 36:15)



The TEXTAREA Tag in HTML Forms

- The <TEXTAREA> </TEXTAREA> tags defines a multi-line text input control
- A text area can hold an unlimited number of characters
- The text renders in a fixed-width font (Courier)
- The size of a textarea can be specified by the *cols* and *rows* attributes
- The *wrap* attribute determines text wrapping
  - wrap = "off" – turns off text wrapping in text area
  - wrap = "virtual" – viewer see wrapping, but not webserver
  - wrap = "physical" – wrapped text, with line breaks and tabs

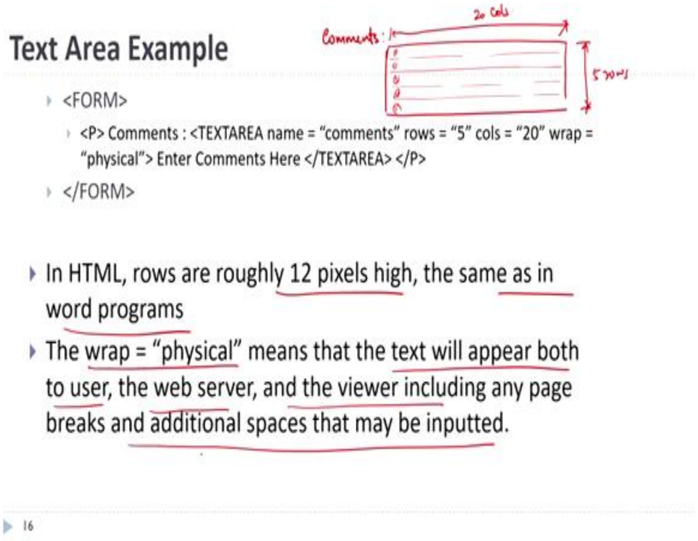
15

Then, comes the one of the last we are getting close to the end of, which is the <TEXTAREA> tags, where you want to input multi line text control when you have large text that you need to type it in. That is the multi line text control is what is part of this.

- So, the Textarea can hold an unlimited number of characters, there is no limit with what type of characters can do this.
- And, it always renders with a fixed width font which is courier, by default it renders in that way.
- And, the size of the Textarea, the user can specify the size of the Textarea by using the columns and row attributes.
- And then, there is a wrap attribute, which determines whether you should allow text wrapping. So, text wrapping means the ideal when you are typing something here, and then, the text would wrap here kind of a thing, that is how if you can turn it off and there is no text wrapping. It can be virtual, the user will see text wrapping with the Web Server would not see any text wrapping.

The physical means the wrap text with the line breaks and tabs are stored. So, now most of the time the wrap is virtual so, you would just want the user to see the wrapped text, you do not want the Web Server to see the wrapped text.

(Refer Slide Time: 37:29)



Text Area Example

- <FORM>
  - <P> Comments : <TEXTAREA name = "comments" rows = "5" cols = "20" wrap = "physical"> Enter Comments Here </TEXTAREA> </P>
- </FORM>

- In HTML, rows are roughly 12 pixels high, the same as in word programs
- The wrap = "physical" means that the text will appear both to user, the web server, and the viewer including any page breaks and additional spaces that may be inputted.

16

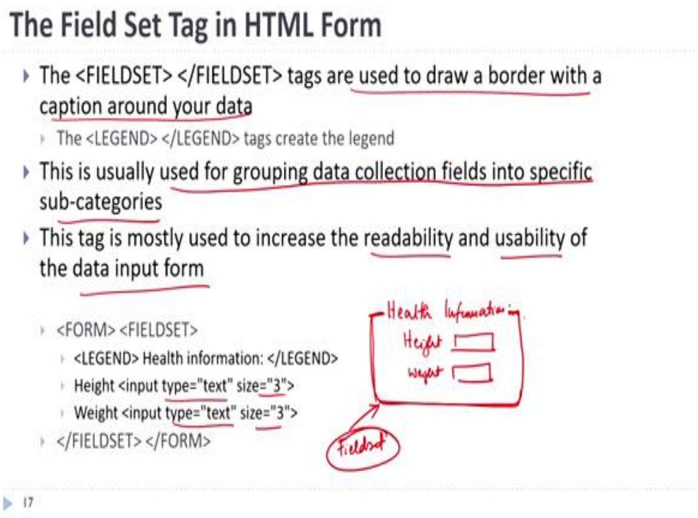
Here is an example of a Textarea. So, it says form comments. So, it will materialize into something like this. Comments, it will show a column and rows are 5, columns are 20. So, it will be something like this, the Textarea like this.

So, there will be 1, 2, 3 so, there will be 5 rows, so, I do not think I did the rows correctly there. So, you would not get to see the rows, but I am drawing this 1, 2, 3, 4, 5. So, this is row one, row two, row three, row four, and row five, so these 5 rows, and this will be the five rows and this will be the 20 columns. And then, whatever you do, you type the comments there. And, just whether you want to wrap it does not wrap all this kind of thing.

- And, in HTML, remember, rows are roughly 12 pixels high. And, as same in the case of the word programs, approximately 12 Pixel font is what is being used.
- The wrap = "physical", the physical tab means that the text will appear both to the user and the Web Server, and the viewer, including any page breaks and additional spaces.

So, when you pull that data from the database table and display it to the user, if the physical one will show it us wrapped whenever it is, but the virtual one when you display it back, it would not show the wrapping.

(Refer Slide Time: 38:59)



The Field Set Tag in HTML Form

- The <FIELDSET> </FIELDSET> tags are used to draw a border with a caption around your data
  - The <LEGEND> </LEGEND> tags create the legend
- This is usually used for grouping data collection fields into specific sub-categories
- This tag is mostly used to increase the readability and usability of the data input form

```
<FORM> <FIELDSET>
  <LEGEND> Health information: </LEGEND>
  Height <input type="text" size="3">
  Weight <input type="text" size="3">
</FIELDSET> </FORM>
```

Then, what we have is the <FIELDSET> tag in the HTML Form.

- The Fieldset tag is something which actually allows you to draw a border with a caption around your data. So, if you look into this example what it actually does is, it will create something like this, health information: and you have what you call as height, you have a Textbox, weight, you have another Textbox and this is displayed like this. So, this is the Fieldset.
- Fieldset is usually used for grouping data collection fields into specific subcategories. So, when you are collecting data, you can use this to group them into specific categories.
- And, this is mostly to increase the readability and usability of the data input form. So, some of these elements that are provided in the HTML are part of the usability analysis or increasing or enhancing the usability of the form. So, anybody, any user who looks into this will see that okay this is the health information and this height and weight are associated or as part of the one and then again, you can see the input type, it is a text input and size is 3. So, you will actually get to only be small with the 3 size font.

So, with that, we have just come to the conclusion of the HTML Form, which is also a user input system. As part of our course, I would request you guys to, we have covered the usability, we have seen what an HTML is, we have seen what an HTML Form is. And, we have also seen slightly what a PHP is, and we have seen how the PHP is invoked from the HTML Form.

So, in the next session, what I will be covering is this CGI or the application, the user, the application in between ties everything together. So, we will learn a little bit of PHP and we will go from there. So, thank you for your patient hearing and we will continue in the next class. Thank you.