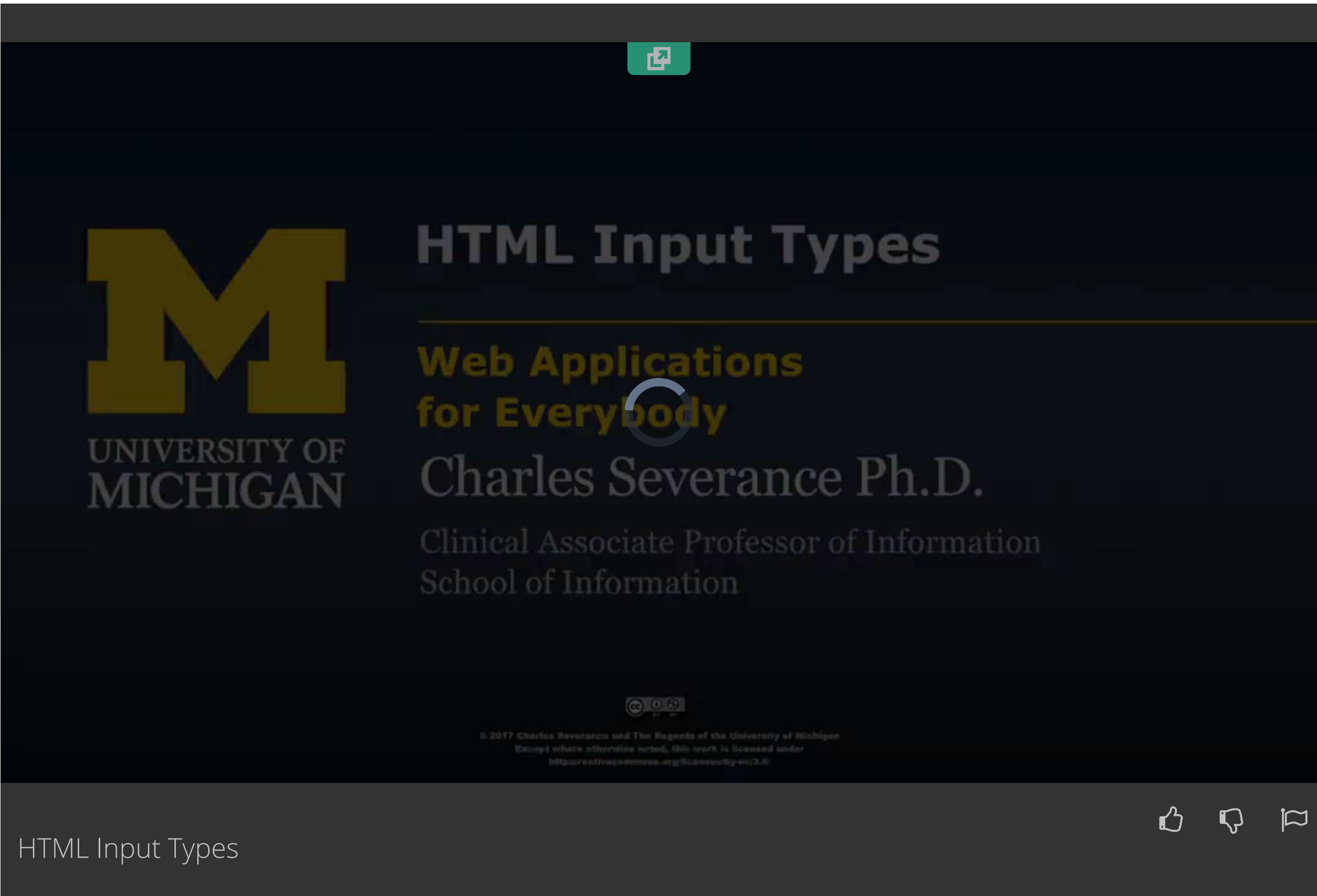


Lecture Materials

▶	HTML Forms	5 min
▶	Using GET and POST with Forms	5 min
▶	HTML Input Types	12 min
▶	Code Walkthrough - HTML Input Types	7 min
▶	HTML5 Input Types	3 min
▶	Processing Form Data and HTML Injection	9 min
▶	Code Walkthoug Forms and HTML Injection (1)	13 min
▶	Guessing Game	4 min
▶	Code Walkthrough - Guessing Game	6 min
▶	Model View Controller (MVC)	8 min
★	Practice Quiz: Forms	17 questions

Assignment

Bonus Materials



Have a question? Discuss this lecture in the week forums.

Interactive Transcript

Search Transcript

English

0:08

So now let's just briefly talk about some input types. I'll give you another video that sort of goes through this in a view source and plays with it in a browser. And so this is just kind of an outline of the kind of types you're going to look at. And often, there's lots and lots of good online documentation, like how to do a textarea. Give me a sample select. And so I tend to just go to the web and say, what, how does the password feel? So this is more of an outline than sort of instruction on how to do this. The web itself is the outline. So here is a couple of fields.

0:40

So text field is the basic field, right? We have a box that we can type stuff in. A password field is effectively exactly the same as a text field. It sends the data in plain text between the browser and the server. That is a plain text, it shows as asterisks, so someone can't shoulder surf. They of course can watch you type, but they can't see your password on the screen, it just hides it visually, nothing else is hidden.

1:07

The thing we're kind of seeing here is how each form field has a different name, and these are the names when we're going to package the data up and send it to the server. Account= PW= and then nick=. So the name is what we're going to show in the dollar get array or the dollar post array and so that's the key value. You'll also see an ID tag on these things. An ID tag, this is a CSS / HTML thing. So in this case I am connecting this input tag with this label. Mostly to improve accessibility but that is an HTML thing, so that's not, here's the browser. And then here's the server.

1:50

The name affects the browser to server connection, but the ID is just all here in the browser. This ID and the for, that's all here in the browser. Size is in the browser. That's all just, size is telling how long to make this little box, right? And so that's all browser stuff. That's HTML and CSS stuff. But the name equals is part of the request response cycle, and how that data is passed in, represented, parsed, and then presented to you inside of your PHP code. The name is the key thing. Some people use the ID and the name to be exact same thing which is perfectly okay, but they have two very different and distinct functions.

2:28

A radio button, this is something that people like. A radio button is a button that you push one, the other one turns off, and you can have many of these, as many as you want. And as soon as you push one, you push this one, this one turns off. And you don't have to have them right next to each other, although sane usability basically says you should. And the way you do this, is you say type="radio", and then you give them the same name. So what this now says is when is going to be equal to am or pm, depending on which of these two things is checked. And so you see this right. In this case we checked pm, hit the submit button and so it looked and figure which one it was, and then it pulled the value out of that one to drive to the pm and then the when. So key value pair. And so literally I could stick another one of these way down here and another one up here, as long as I named them when, then pushing this one would turn all four of these on and off. And so you associate them not by having them near each other, you associate them by having the same name. So that's radio button, pretty good user experience, as long as you got the right number of things, and it's the kind of thing that you want. Check boxes. When you have a list of things, key to check boxes is they can all be on or off. And so you'll give them different names. So class one, class two, class three. And then, I mean, that's what this check box is, class one, class two and class three. And if I check one and then hit submit button, you put a value for each one, right? So class one equals SI502, so class one drives into the post array, and the SI502 tells me which of those things was checked. If you don't put it, it assumes value equals on, So by default, that's the default. So if I check the third one and hit submit, I would get on.

4:20

Select drop down, we use these all the time. They're really cool. Especially because we can read like what brand of car or whatever from a database and fill this stuff up and it's really nice. It saves us a lot of pre typing. Although later we'll find out how to pre type stuff and put that in a form field. Text field so you can type something in a prefix but that's way later in the class. For now we're keeping it simple. So this is one field, not six fields. So you put the text of please select, and if I just submitted the button right now, it would actually send soda=0, which is exactly what is shows right here okay? So it equals zero because that's the one that was selected. Okay, but if you pick, and you switch down and you pick Mountain Dew, then it will send soda equals 3, so that's what it's doing. There is one name, and multiple values. And based on the gesture the user is picking, that's going to do it. And if you give it this top one, your code will be like hey you were supposed to select one, so you'll know that zero. You write the zero, one, two. Doesn't have to numbers could be Fred, Sam, Sally, whatever. You pick them and then you make sense of them inside your program. So inside your program, if you got this one that said zero. You might say hey, you didn't select a soda. And then go back to the screen with an error message and ask for the soda again.

5:44

You can have a default that's something other than the first one, and again this value is just a thing you're choosing, right? Chips, peanuts, cookies in that case. It's just a string, it's not always a number. A lot of people do use a number, because it's just a way for us to keep things sane in our minds. But it's just a string, so we got a name which is going to give us our key. Snack gives us our key, right? And then based on whichever one this is selected, if you don't have this one, this one is selected by default. But if you say selected you only want to do it on one of them and then that says it'll start at peanuts. So if you pop it up, there will be some things above it and some things below it. You can switch but if you hit submit, you're going to get peanuts because that's the value. The value decides that name, the name value pairs.

6:29

Okay, the textarea is like a blog comment, or a blog post, or whatever. And there's actual little JavaScript, which we'll talk about later, that makes these things, and where it puts bold tags and fancy stuff, and you can see these text areas sometimes at toolbars. That's all JavaScript that's augmenting the textarea. The fundamental thing that's in HTML for blocks of text as text area. Now, the textarea is intended for longer chunks of text, and so it has a little different pattern. Everything we've seen up till now has a value equals. That sort of determines the thing that's going, but that's not how it works in textarea. Textarea and /textarea surround a bit of text. There's things like rows and columns and names. The name is how it's going to be submitted to the server. Whatever you type in here is going to be submitted to the server. The previous data, you can put the old data in here. I love building websites in MySQL. Then you come and you type anything you want. [You hit the submit button, and then this is all bundled up and](#) sent in under the key about in your POST data. Okay, so that's a little different, everything else we've done. The thing that get sent is part of the value that attribute, all right, the value attribute, value peanuts. But in this one, it's not the value attribute, it's the stuff in between.

7:55

So, this is a multiple select check box, and most people just say never ever do this, build some other user interface. I just include it for completeness. So, it's a type, a multiple select and it's the key that you're going to send in is actually an array, that's what this whole thing is saying, it's array. And then you have a set of options, values, and with enough gestures you can check more than one. And then when you check more than one, you get an array of the checked things. So code itself inside a post, remember we talked about how you build simple data structures by putting arrays within side of arrays. So this is an outer array with a key of code and dopost. And the data under code is itself an array, which is a two item array with item 0 CSS and HTML, because we've checked those two. And then we hit submit, and that's what comes into our server. But no, no, no, no. Most UI people will tell you just don't do that.

9:00

And I have to agree totally with that.

9:02

Another thing that we see is the Submit button. Submit button itself can have a name and a value, name and value. And the value is weird in that the value also shows the text. And that's completely different than everything else. Meaning that when I said value equals zero in the previous thing. You never saw the zero as the user. This actually you see as the user, which to me kind of, ugh. I mean it value should not be that, it should be the value should be the thing that get sent to the server. The name of course is the thing you can send to the server, the value is the thing that get sent to the server. But the value also is the thing that you see, and that's the part that I don't like about this. But hey, what I think about it, this is like 25 years old. So we're not going to argue about it anymore. I just think it's tacky. So what happens is when I'm writing code I sort of don't think about this, I use the existence of the key using isset, remember isset. So I can say if the dopost is set, then I know that this button got submitted. And if I have multiple buttons, I can determine which of them is by giving each of the buttons a different name. Rather than having the buttons have the same name and different values. The value I just used for the visible part of that.

10:28

And another thing that you're going to see is

10:31

how we're going to actually override the behavior of a button and do something with it that's kind of more like an anchor tag. And so just you'll see this and so it's a button, value is escape, which tells what text it is. And onclick is a bit of a JavaScript, we will talk more about this when we get to JavaScript. Onclick is a bit of a JavaScript code, and for now, you'll just see me use this in lots of my examples, so I want to explain it. What we're are saying is Location.href. And what that really is doing is, if this is your browser and you've got this Location bar that's http [SOUND]. What this is saying to the browser, in JavaScript, because remember JavaScript can modify the document object model, it actually is putting this string, this string right here, into that as if you typed it. And then hitting the Enter for you. Location.href= says, put that in the browser and then go to that page. So its causes the get request to go to that page. And this return false, which you'll see in a lot of these little onclick gets, what the return false says is don't actually submit the form, because this is a button, and it might submit the form. And what I've really done here is I have made an anchor tag that looks like a button.

11:47

Now those of you who are CSS experts will say, you should have done that with CSS. And the answer is, yeah, I probably should have. And if I'm using bootstrap or another CSS framework I would do that, but this is kind of the old school no CSS way to accomplish that.

12:03

So the next thing we'll talk about is HTML5 input types.

Downloads

Lecture Video	mp4
Subtitles (English)	WebVTT
Transcript (English)	txt

Would you like to [help us translate](#) the transcript and subtitles into additional languages?