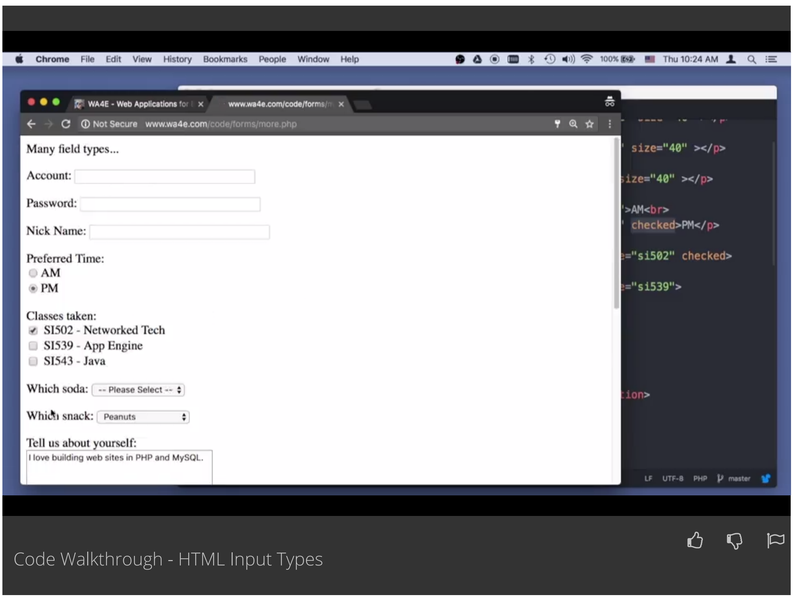
Lecture Materials

Assignment

Bonus Materials



Code Walkthrough - HTML Input Types

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

Interactive Transcript

Search Transcript English

0:00
Hello everybody and welcome to Web Applications for Everybody. I'm going through some of the sample code in the form section of the course. Right now, I'm just going to go through a few of the extra form fields and literally my- this page is not the greatest. You can google all these form types, away you go. So here's the code we're looking at. We have your basic text and you'll notice I'm using this label for pattern over and over and over again just because I'm trying to make my markup semantic. You'll notice I'm not perfect at that. The next type that we're going to talk about is the password. Blah. And then the password is super secret, "secret," the most common password. And so password is just like a text field except that it doesn't show it while you're typing. Now if I'm going to submit this as a post variable, look. The secret just goes in plain text. It goes into the PHP in plain text. If I was to do a "View Developer Console" and watch the network. What was that? Credit card? So if I type in the password for secret and I press submit, and I look at the post that just happened, it sent the password across the network in plain text. So password is really only there for you to keep people from shoulder surfing as you're typing the password. Okay, so a nickname is another text. Okay, this is a radio button. And so type="radio" and radio button is like a station selector on a car radio. And the idea is that no matter how many of these things there are, only one can be turned on at the same time and you do this by naming it type="radio", but then grouping them by having the same name. Remember name is what's sent to the server. Key value. And then different values, so value="am" is what's going to be sent to the server if this one is checked. value="pm" is sent to the server if this one is checked. And then there's an optional attribute that just says "checked," which indicates when this first is refreshed that "PM" is going to be the default one that's checked. So if I hit "AM" and press submit, then you can see when="am". So it really picked, among the things, when="am" from the source code. Okay. So that's that one. Checkboxes. Checkboxes are independent, so you can have any combination of checkboxes that you want: all of them, none of them, whatever. So they each have a different and distinct name, and a different and distinct value. If the value is not given, then "on," the string "on" is what's sent. So I'm going to see all three of these. Class one, class two and class three are going to be checked from that checkbox, okay. And "on" is because I didn't specify a value, but most of the time we're just doing an "isset()" to see if this key is set in the $_POST as compared to looking at the actual value. But it depends. Okay. So the next two examples are drop-downs and again, I got a label. And I have the name of the data that's going to be submitted. And then, "option," "option," "option," "option," "option." And those options effectively, in order, pick these things. And then when we hit submit, three. If I pick Mountain Dew then three is going to be sent. In this next one, all we're doing is we're picking a selected one so that it's not the top one, right. Peanuts. I can change it to chips cookies if I want to. And then in this one, snack="chips" is what's going to be sent in this one soda="3" is what's going to be sent. So let's go ahead and submit that. And so soda="3" and snack="chips" and so that's how the drop-down works. Text areas. Text areas are cool in that they don't have a "value=" like most. You just put the existing text that you want in between "textarea" and "/textarea". Name is how it's going to be sent. You can put blanks in here. Blah blah blank blah blank blank blank blah blah blah blah. Okay. And now if I hit submit, you'll see all this is bundled up into a single value, right. So this is the "about" and it's a string and it has new lines in all the spaces, in whatever, and that's why you can write little paragraphs in here. Now there's all kind of plugins that will turn this into something that turns into bold because I could say, and I'm not filtering this very well so it's going to look bad. And so when I submit some HTML. And look at that. It's bold. Now the danger is I could submit some JavaScript and ugly stuff and so there's another filtering any time I'm not I'm clearly not filtering this stuff when I'm just using "print_r()". And so this is a dangerous page because HTML could come out. The last thing is something that you pretty much, well second-to-last thing, is a multiple select. And so this is sort of like the "option value" except you've said it's multiple, and it actually sends an array of codes because they can all be turned on at the same time. This is generally a hard way to do user interface. And let's see if I can get it. "Option" doesn't work. I can never remember. "Ctrl" doesn't work. Command works and now I've got two of them done. This is a terrible user, because no one, including myself, knows how to use it. But when you send it, you get an array that tells you which of the things were checked based on this thing. And then the last little trick that we'll show, we got a submit button and in this case we got a name and a value. So the value in submit buttons is a little bit weird in that it both changes the text on the button, and it sets what is sent. Most people just name their buttons different names. So this name="dopost" and check to see the existence of the button in the post data to figure out which of more than one button, if there are more than one button, rather than looking at the value because the value is often translated. If you have a multilingual application, "Submit" would not be the text that was here. So checking for the text is a little problematic. And then we have one more little trick, and that is how to turn a button into a anchor tag. And that as you say input type="button," "onclick" is a bit of Javascript, and then you simply say, switch this browser to this particular URL and then don't submit the form. That's what "return false" is, and you'll just see we will use this for escape or cancel or done or all kinds of things to get out of a form without actually submitting the form. So if I click this it's going to take me back to Web Applications for Everybody. So, that was a really quick zoom through the sample form code and I hope that it was helpful to you.

Downloads

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

Would you like to help us translate the transcript and subtitles into additional languages?