

Simple Login Form



Paul Cheney

SPARTAN DESIGN UNIVERSITY

spartandesignuniversity.com



Agenda



Start file

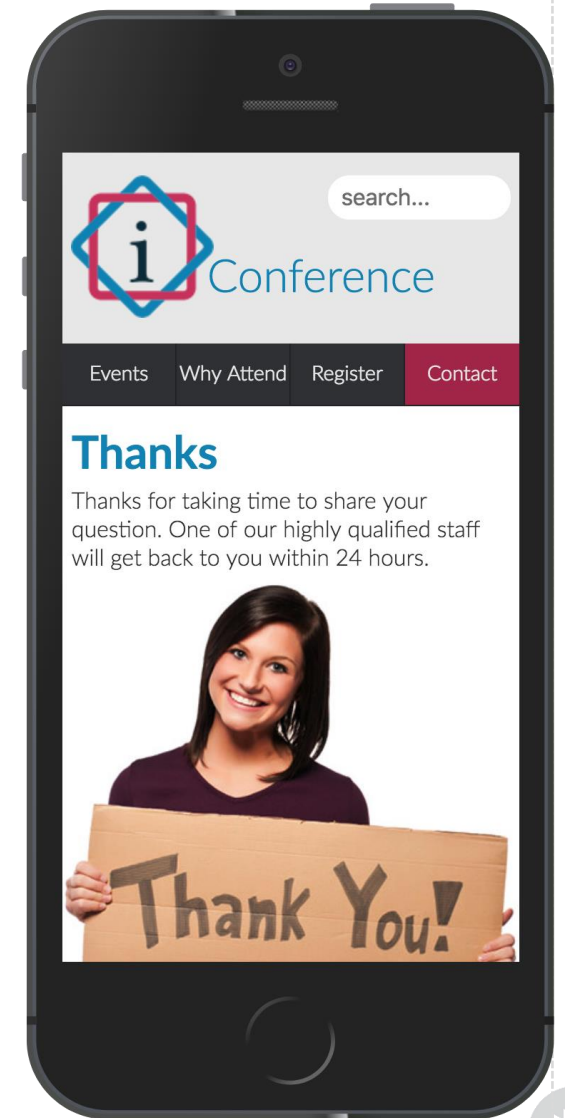
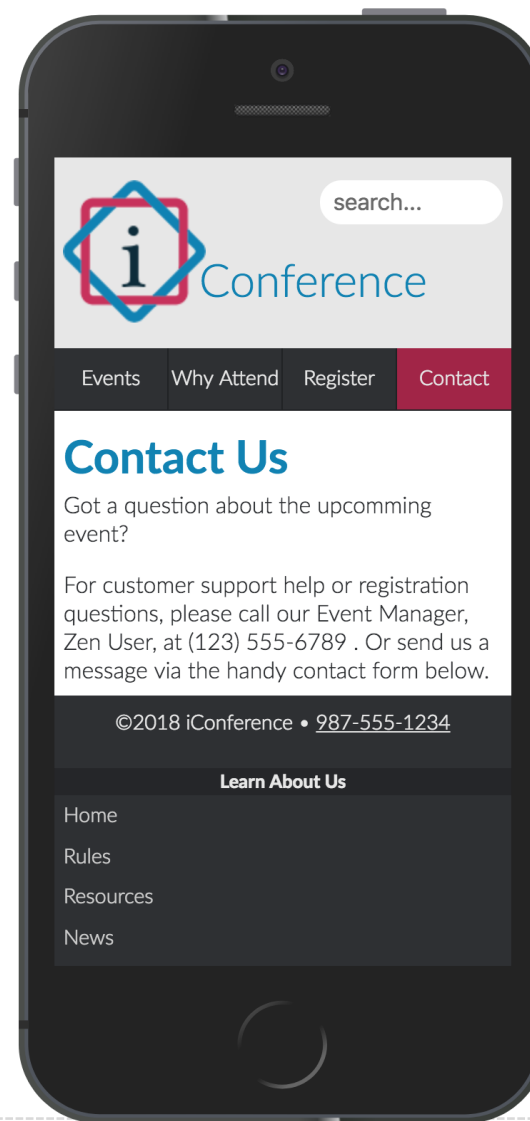
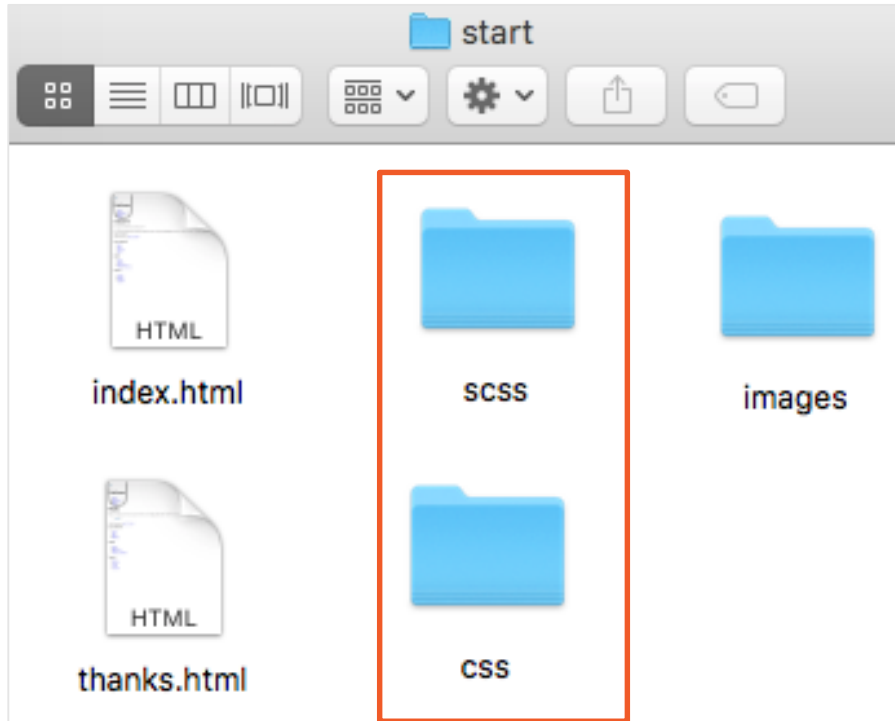
Build a form

Style a form

Test on mobile



Start File



For my code editor, I will be using a free program called Brackets. Let's start by dragging the start folder from our desktop to the left pane of Brackets. Here you can see all the files we just talked about.

I will now open my CSS preprocessor and add the start folder to the interface. Please verify that the input folder is your scss folder and your output folder is css. Then click the blue start button to have it watch your files for changes. When it detects any change to the scss, it rebuilds the css file immediately. You may hear a beep when you save your files. Scout is letting you know its made some changes.

Down here you can choose development or production. When I am building a site and I want to be able to read the css, I will use Nested which looks like this and lets you know with comments where the source scss code is located.

Once I am done and ready to deliver the site, I change to production and compressed. This removes all the comments and extra spaces and created the smallest file possible.

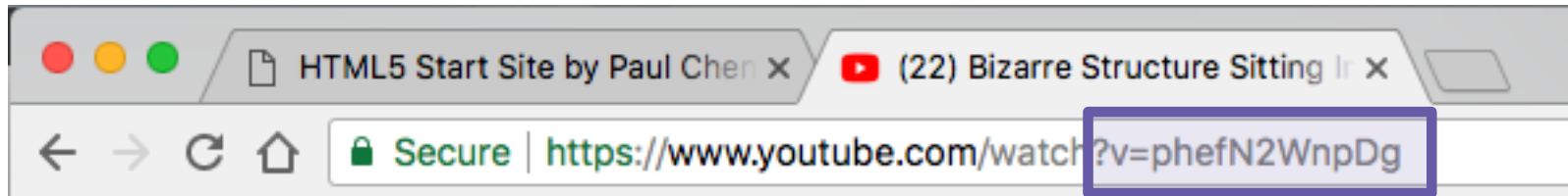
Finally lets open a browser. I will use Chrome for this module.

Brackets has a live preview which works most of the time and it's really good for hand-on tutorials like this because you can see the code changes instantly without pressing reload all the time.

Scroll down to line 52 and enter an opening and closing form tag with a bunch of returns between.



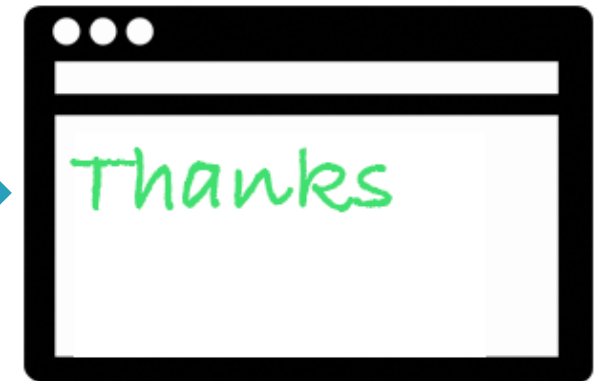
Processing a Form



GET



POST



Please add `method="GET"` and `action="thanks.html"` which we can see here in the right hand panel. Now lets build our first input.

Type an input tag and remember that inputs do not have closing tags.



Inputs

type

value

name

required / not

user instructions

placeholder



For this input we want the users full name so lets add type = text.

For the name we will consult our google developers page and we can see that under name, we have options for full name, first name, last name and even middle name.

<https://developers.google.com/web/updates/2015/06/checkout-faster-with-autofill>

So we will add name = name

We also want this to be a required entry, so just add the word required.

Now for the instructions to the user. In front of the input type "First Name" followed by an opening span tag and then type (required) in parenthesis. The span tag allows us to style the word required differently.

Now all we have to do is wrap the whole thing in a label tag. Remember to move the closing label to the end.

Hopefully you have been watching over here as we were building this input.

Notice that when I click on the user instructions, the input activates and I could start typing my name. This is because we we wrapped them both in a label tag.



Next we will ask for an email address. Please copy what you have done and paste it below.

In this one we are asking for an email so change the user instructions.

Leave required in the span tag

Change the input type to email. This is necessary so that the browser can validate if the input is an email or not.

If we consult our google developer page, we can see that the name is also email. No surprise there.

Finally add a placeholder ="someone@gmail.com"

Lets build the phone from scratch.

Start with an opening label tag. Then user instructions of Phone.

Next the input tag. Our Google developer page tells us to use tel for the type. This also has the added benefit of bringing up a keypad on our mobile device. We should set the name = to phone. This is not a required field.

Finally we will add a place for comments.

Begin with a label tag and user instructions

Follow that with a textarea opening and closing tag.



The last thing we need is a button to submit the form. For this we also use an input but the type = submit. You can see that by default I have the dreaded submit button in my web page. From our best practices we know that we should say "I want to.." and then what follows becomes the value for the button. In this case I want to "Ask My Question"

If you click on the button you can see the browser flagging you to fill out the first name. Once we put something in both required fields, we see the thanks page.

Now lets go back and work on the CSS to make it look nice. But before we do, we need to add a class of "submitBtn" to the button that submits the page.





Please open the small default file.

Around line 105 lets add a nested section for all our form styling.

In this case we are styling 4 labels and according to our best practices the user instructions should appear above and left aligned.

Notice how the form is touching the left edge of the screen and the button is touching the footer. Please add margin of 1rem top and bottom and the variable gutter which we can see is 2% here in the variables file.

When we widen the page we can see that everything is on a single line. According to our best practices the elements should be stacked in a straight line.

Add a nested label and display them as blocks. Lets also give them some breathing room by adding padding to the top and bottom only.

Next lets style the word required which if you remember is inside a span tag. Set the font-size to .75rem and change the color to the variable \$secondaryColor.



After the closing span, lets work on the three inputs and one text area.

Display them as blocks so that everything is now in a straight line.

We will follow our best practices and set the font size inside to 16ps using 1rem which made them a little larger but to get them to be about 44px tall we need to add padding of .75rem all the way around

:Lets add a very light grey fill color and notice that they now look 3 dimensional. We will override that with a thin grey border

Notice when I type that the text is black. We can control that as well by adding color and setting it to the variable primaryColor.

Notice that the descender of the letter Q is touching the input boxes below. Lets add a tiny bit of separation using margin-top of .2rem. Now they are separated.

Finally lets set the width to 100%.

While this works great on a small screen it looks pretty bad on a large screen.

Lets now go back to form and limit the max width to 25rem which is 400px;

The whole purpose of this form is to get a question, so lets increase the size of the text area. I believe that wraps up the changes for the labels.



We have identified the required fields using text but let's back that up with a colored line. Please add a selector of input and a pseudo class of required. Let's set the border to 2px solid and the secondary color. Now that makes it very obvious what is required.

Let's also provide visual feedback once the field is approved.

Add another selector of input colon required another colon and valid. This is for any required input that is now valid. I guess that was obvious.

Please change the border color to green.

Notice that when I activate an input there is a blue line provided by the browser. This will be different on different browsers. Let's add our own visual indicator by adding input pseudo class focus comma and textarea colon focus. We will darken the background color to #ddd. Now as I click between each input, I see a darker fill color.

I think we are ready to help the button out with some styling.



Since the button is not within a label we need to go down here after the closing curly of the label.

Add input dot submitBtn.

First we will turn off the default border.

Next lets set the background color to our variable secondary color.

In order to read the text, lets change the color to white.

To make it easier to read increase the font size to 1.5rem.

Now the text looks heavy so lets remove the bold and change it to our variable of normalWeight.

You can see that this is equal to 300

Lets add padding of .4rem and 1rem so that the button meets the recommended height of 44px.

Finally I will make it appear more buttonish by rounding the corners.

Since I want this call to action to really call out, I am going to set the width to 100%.

When I widen the page, the button stays the same width as the other form elements.

Now its time to upload this to a real server and check it on a real phone.



Real Phones

No Service 11:59 AM 46%
sandbox.spartandesignweb.com

Email (required)
someone@gmail.com

Phone

Your Question

Ask My Question

©2018 iConference • [987-555-1234](tel:987-555-1234)

Learn About Us

Home

No Service 12:00 PM 46%
sandbox.spartandesignweb.com

Email (required)
someone@gmail.com

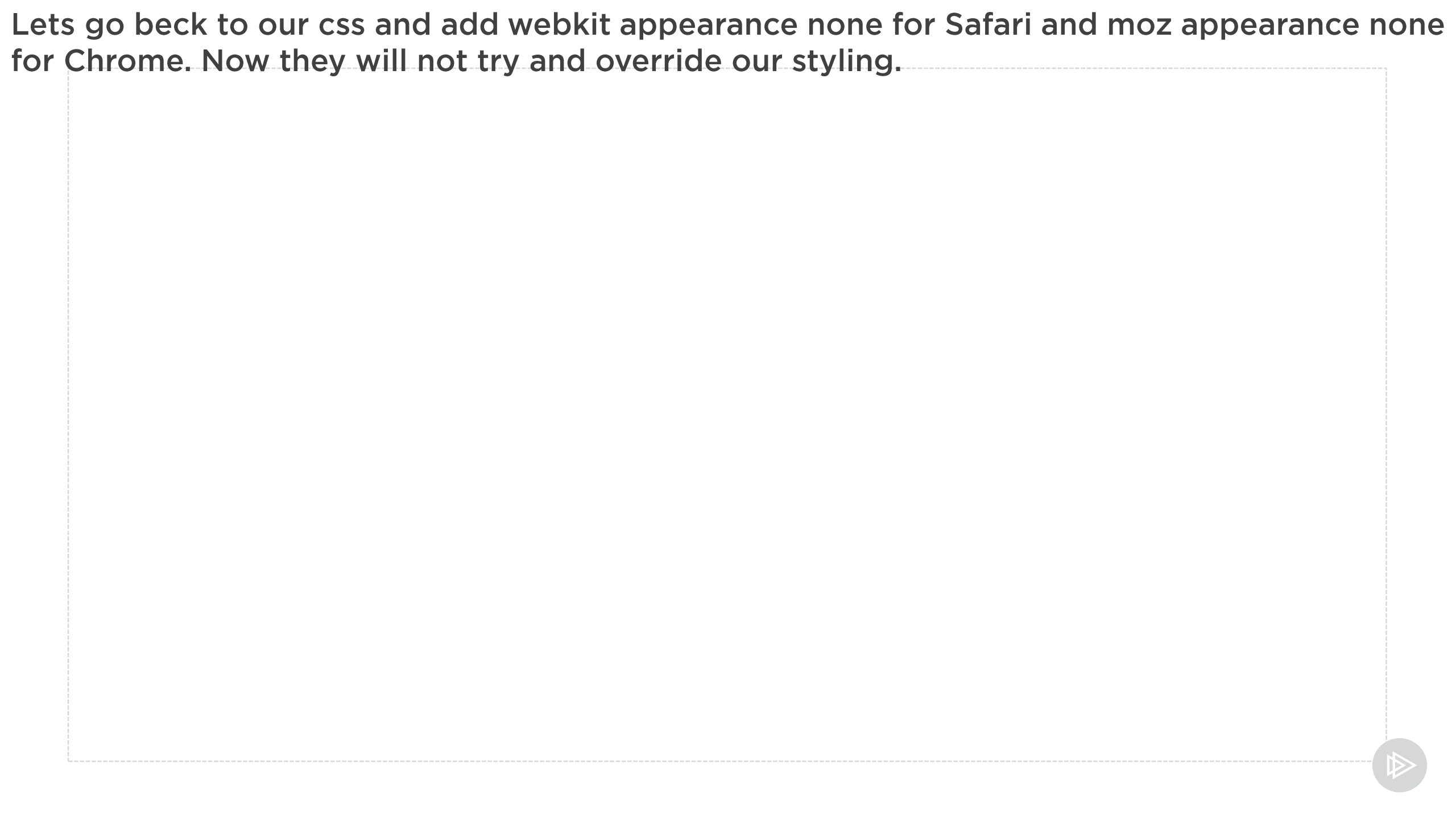
Phone

Your Question

Ask My Question




©2018 iConference • [987-555-1234](tel:987-555-1234)





Lets go beck to our css and add webkit appearance none for Safari and moz appearance none for Chrome. Now they will not try and override our styling.

Real Phones

No Service  12:07 PM  45% 
sandbox.spartandesignweb.com

Email (required)

someone@gmail.com




Phone

Your Question

Ask My Question

©2018 iConference • [987-555-1234](tel:987-555-1234)

[Learn About Us](#)

No Service  12:08 PM  45% 



andbox.spartandesignweb.com



Email (required)

someone@gmail.com

Phone

Your Question


Ask My Question

©2018 iConference • [987-555-1234](tel:987-555-1234)

[Learn About Us](#)



Search Input




Events	Why Attend	Register	Contact
--------	------------	----------	---------

Contact Us

Got a question about the upcoming event?

For customer support help or registration questions, please

```
<form>
  <input type="search" placeholder="search...">
</form>
```




Events	Why Attend	Register	Contact
--------	------------	----------	---------

Contact Us

Got a question about the upcoming event?

For customer support help or registration questions, please

```
input {
  width: 8em;
  transition: width .5s;
} // end input
input:focus {width:12rem;}
```



Events	Why Attend	Register	Contact
--------	------------	----------	---------

Contact Us

Got a question about the upcoming event?

For customer support help or registration questions, please



Summary



Start file

Build a form

Style a form

Test on mobile





Introduction

Web Form Best Practices

Simple Login Form

Long Web Form

JavaScript Validated Form

