* -: In the last lecture, we improved the user experience by taking advantage of the form state handled by Angular.
* Now, let's improve this even more and let's say we ***want to output some help text below this input to assist if there is an invalid value in there.***
* Now we can use a bootstrap class here help-block to make sure that this has the appropriate styling.
* And then we could say, Please enter a valid email.
* We only want to show this if a valid, excuse me if an invalid value has been entered into the control associated with this input though.
* *Well, a quick and easy way of getting access to the control created by Angler automatically as we added ngModel to the input is* ***by adding a local reference*** *to this input element here.*
* For example, email, any name you like and associating this now, not with NG form as we did for the overall form, but with ngModel.
* So just like the form directive automatically added by Angler when it detects a form element.
* The ngModel directive here also kind of exposes some additional information about the control it creates for us on the overarching form by accessing ngModel.
* So with this, we could simply check or say that we want to attach this span here if email is not valid.
* So add exclamation mark at the beginning.

Text

Description automatically generated

* Now if this, if this reloads, you'll see that help text.

Graphical user interface, text, application, email

Description automatically generated

* And of course you can improve this by also chaining another condition.
* That email should have been touched to give the user a chance of changing it.

Text

Description automatically generated

* So now you don't see the warning but if you click in there and click out of there and it is invalid, you see that warning.
* And if you now enter a valid data, it disappears.

Graphical user interface, text, application, email

Description automatically generated

* So this is another way of taking advantage of the state managed by Angular but this also shows you how you can get a quick and easy access to control added by Angular.
* And with that, you should have tools to really provide a pleasant user experience showing the right errors, the right warnings and styling the form correctly depending on the state, off the form.