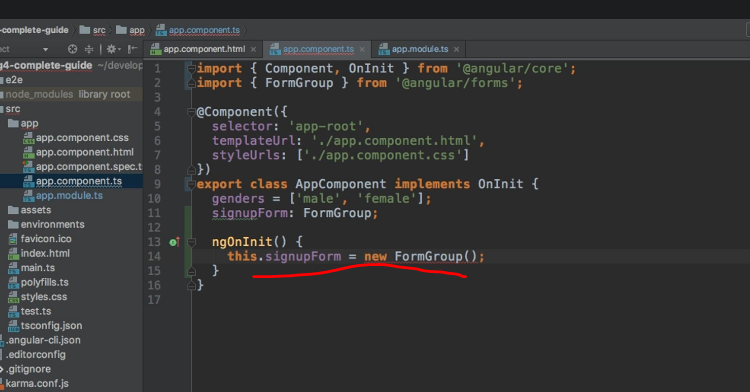
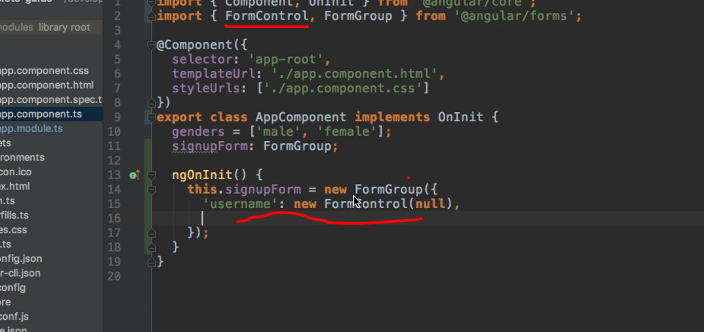
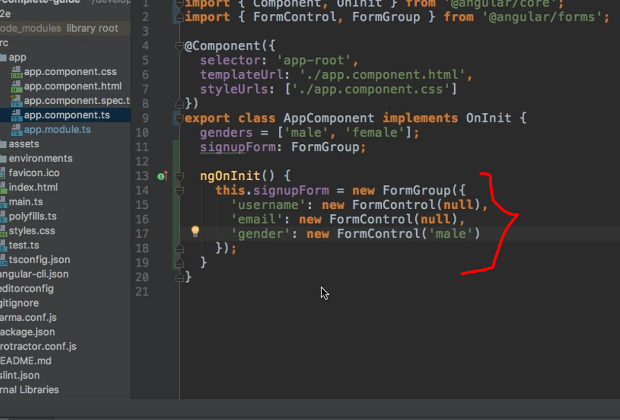
* -: So we set everything up in the last lecture, Now let's create our form.
* Now we could initialize it right here to set up value immediately, but I don't like to do this here.
* I want to have it in a method because it actually is quite a lot of code and it's not the best place to have it right here at the top in the property declaration area.
* So I will use the OnInit lifecycle for this.
* So I will import OnInit, make sure to import it from at angular core.
* And therefore I'm required to implement the NGOnInit method.
* And in here I will initialize my form.
* You should initialize it before rendering the template, of course.
* So make sure to use a lifecycle hook, which is called before the template is rendered.
* Here, I will set up my sign up form and we'll set it equal to, to what? Well keep in mind, sign up form is of type form group.



* So we need to create a new form group here just like this.
* Now we have almost a valid form.
* We need to pass a JavaScript object here and we're theoretically done.
* This is our first created form.
* This JavaScript object configures it and as it's empty, it simply says, or it tells angular, Hey, this form doesn't have any controls, so let's add some.
* Controls are basically just key value pairs in this object we pass to the overall form group.
* So let's have a look at our HTML code.
* We have a username field and a email field and then we got the radio buttons here at the bottom.
* So it would make sense to have a username control.
* I'm wrapping this in quotation marks to make sure that during minification, when this code gets mangled, this property name is kept because I will reference it in the HTML code.
* This might not be necessary, it might not get destroyed but just to be sure, I'm wrapping it as a string.
* So here, what is this now? Well this is a **form control** another class made available by the reactive forms module or by this forms package here I should say.
* So I will set up or create a new form control here.
* And form control should also be added from the forms package from at angular forms.
* Now to this form control constructor, we can pass a couple of arguments.
* ***The first argument is the initial state, the initial value of this control*** you could say.
* ***The second argument will be a single validator or an array of validators we want to apply to this control***.
* I'll come back to this.
* ***The third argument will be potential asynchronous validators.***
* I will also come back to this later.
* For now, I want to set an initial state of null to have an empty field, but you could also pass a string like default username.
* And now this would later be displayed as a default username.
* Again, I'll keep it at null for now.
* Now have a look at the template again, the next is to email.
* So I'm going to add a number control and a number key value pair in this JavaScript object, which will also be a new form control.



* And here I will also set a default value of null.
* Again, validation will be added later.
* Now, gender does the gender, the user select is also a new form control.
* It might be a radio button, but that is just a normal input in the end.
* So to angular, it simply is a control.
* And here I could set this to null, but I want to select a default gender.
* So here, I will actually select male because that's my gender.
* Feel free to pick female of course or add more genders to this array of genders up here to begin with.



* With this a basic form is initialized, it has free controls, and that is all for now.
* This is our first basic form.
* Now, in the next lecture we're going to connect this form we created here in TypeScript to our form we have set up in the HTML code.