**101. Understanding ngSwitch**

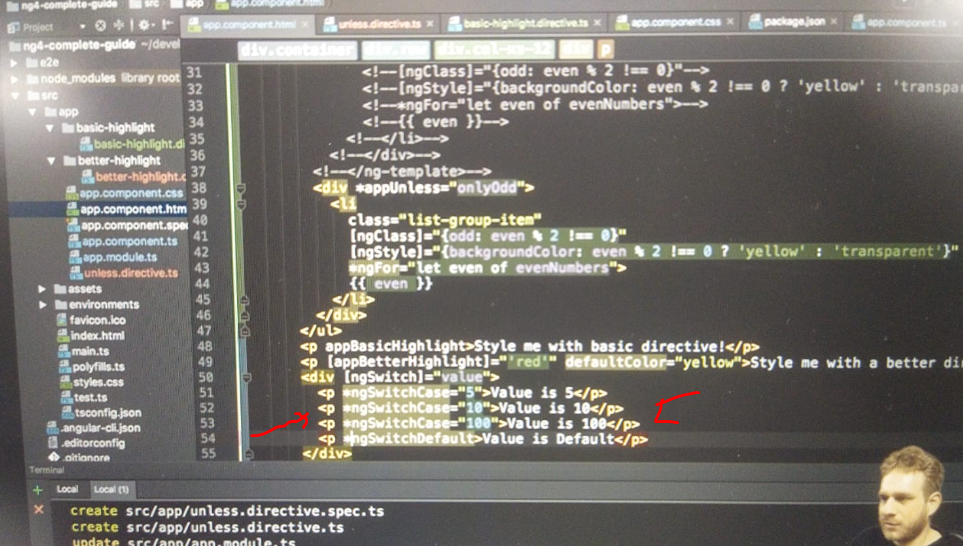
* -: I want to conclude the section with one of our built in structural directive, which might be handy from time to time, ngSwitch.

**ngSwitch – Structural Directive**

* Imagine the case that we have a value like this, which is simply 10.
* Anything you want.
* Now we got a place in our app where this value changes and we got a couple of different messages we want to display for each of these values.
* For this, we can use ngSwitch.
* So, let's add a new div here below our paragraphs with our own attribute directives.
* And ngSwitch is used like this.
* We bind to ngSwitch with property binding, so no star at this point.
* And we bind to value.
* So this is our condition basically, what we want to check.
* And then you might be aware that Switch has a couple of cases we can now cover.
* So let's say we have a paragraph where we say value is five.
* Then we have to same for 10.
* Then we have to same for 100 and maybe for default.

**ngSwitch and ngSwitchCase/ngSwitchDefault:**

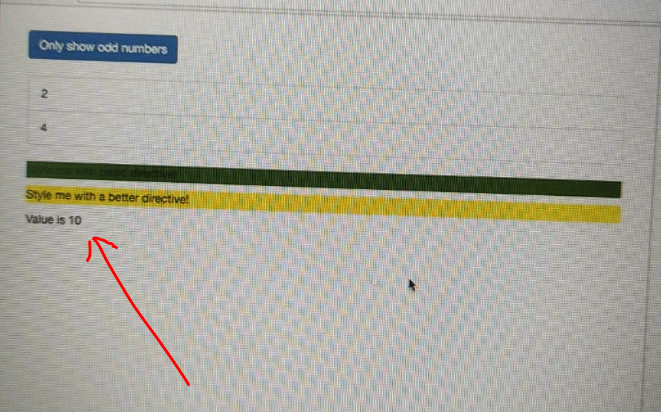
* Now we need to add something to these paragraphs to control which paragraph gets shown because it should only be one of these at a time.
* That something is ngSwitchCase and we simply pass the value for this case here as an argument, for example five.



* And we need to start at the beginning.
* That's super important.
* So here the star comes into play again because behind the scenes this is then all transformed by Angular.
* So now we can use this here for the average use cases too, with 10 and with 100.
* And finally, for the default case, we have ngSwitchDefault, like this.
* Now here we also need to add the star, of course.

Text

Description automatically generated

* With this in place, if we save this we only see 10 here.
* 
* And if I go to my component and I change this to five, like this, then of course we see five.
* So this is ngSwitch, super easy to use, might be useful in some cases where you find yourself creating a lot of ngIf conditions.
* ngSwitch might be the better solution in such cases.
* So that is structural directives finished and that is directives overall finished.
* You learned a lot about them and I hope this knowledge allows you to create awesome directives for your applications in cases where the built in ones aren't everything you need, even though these already cover a lot of the default use cases to be honest.
* So, let's move on.
* Let's add a directive to our recipe book in the next section, and then let's move on in the course.
* See you there.