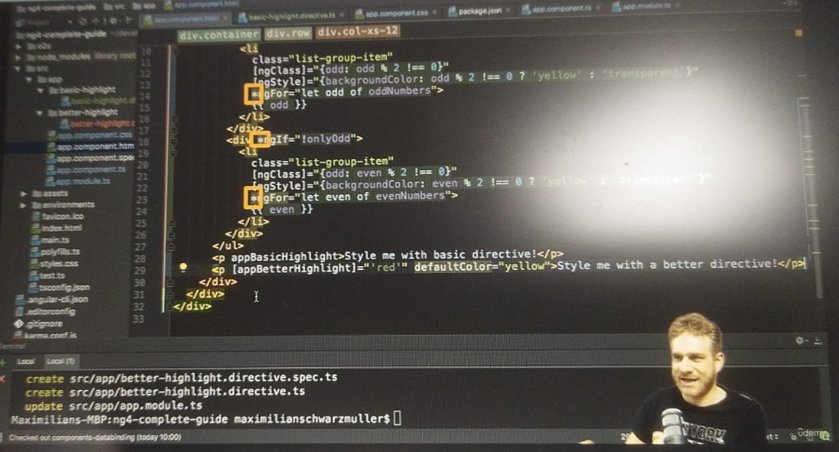
**99. What Happens behind the Scenes on Structural Directives:**

* -: So, now that we had a closer look at attribute directives and created our own one, before creating our own structural directive, let me explain why this star is actually required.



* ***This star indicates to Angular that this is a structural directive.***
* The question is, why does it need to know? Because structural directives with that star actually are just a nicer way for us to to use them, basically.

**Behind the scenes – Structural Directive:**

* Behind the scenes, Angular will transform them into something else, because there is no star in Angular syntax when using, well, directives, or when using property binding or anything like that.
* *There is only* ***property binding, event binding, two-way binding, and string interpolation***, and there is no star operator or something like that.
* So behind the scenes, Angular needs to transform this NGF usage into something where we end up with these tools, property binding and so on.
* And it does; I can write the same list here differently.
* So if I try to write this only art block differently I would end up with the following.
* I would have a ng-template element that is provided by Angular as the NG indicates.
* And inside this element we have the content we conditionally want to render.
* So, in this case here, we would have a div inside of this and in this div, our list item.
* Now, you probably see that it's basically the same but that it's wrapped an ng-template.
* The reason for this is that ng-template is an element which itself is not rendered, but which allows us to define a template in the end for Angular to use once it determines that this template, some element, needs to be rendered, because this condition is true in this case.
* So on the ng-template, we place NGF and now not with the star because this is the form to which it will get transformed due to the star, but instead with the tool we already know, with property binding.
* And here we simply bind, well, not only odd, so the condition.
* So now we're back to the tools we know without the star and that's why we have the star because obviously that's the more intuitive syntax to use directly placing it on the element we want to conditionally render.

Text

Description automatically generated

* But that would be the real content behind the scenes to which it gets transformed.
* Now you don't need to use that, but I find it super important to be aware of this.
* Now therefore, if we save this, we see the list twice, and we see disappear and appear twice, because it works exactly the same.
* It is exactly the same code behind the scenes.
* This is what the star transforms it to.