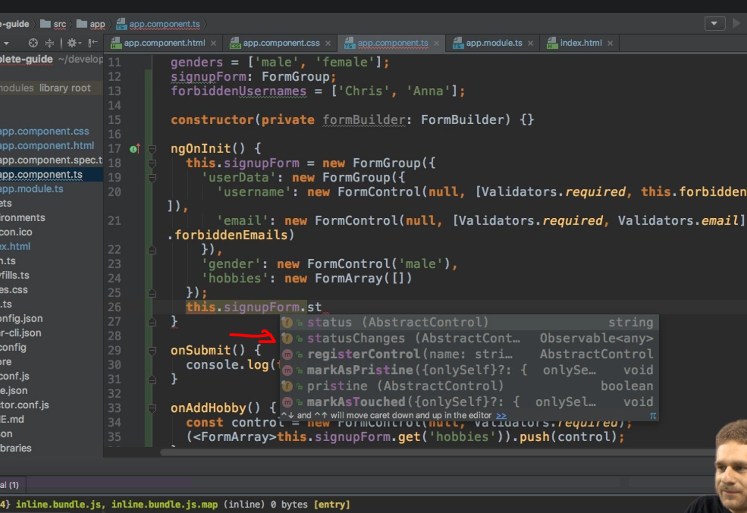
* -: We're almost done with this section.
* We worked a lot with reactive forms now.
* And in the last lecture at the end of it with the asynchronous validator, you saw that ***the status of this input switched from invalid to pending to valid for example.***
* There actually is **a form state you can track** in general.
* I'll do this here on 'ngOnInit'.
* On my signup form and on each control of this form, so on get any control to, you have two observables you can listen to.
* **Status changes and value changes.**



Text

Description automatically generated

* So let's listen to value changes.
* As mentioned, this is an observable so we can subscribe to it.
* And as the first argument, we pass the call back which is executed when new data arrives.
* So this will be the value because we're listening to value changes, and we can lock the value here.

Text

Description automatically generated

* So if we do this, have a look at our form.
* If I type something here you see with every key stroke, the object here is printed, the value of the form actually.

Graphical user interface, text, application, email

Description automatically generated

* This makes sense because value changes is fine whenever we change something about the form, so whenever we type into any input.
* So this is value changes.
* ***And again, you can also call this on individual form control.***
* I'm going to copy this and then comment it out because we also have status changes.

Text

Description automatically generated

* So here we receive the status now.
* Let's view this.
* Here, if I type something you see invalid because the form is invalid.
* If I type something in here where we have a asynchronous validator, you'll see it switch to pending and then after one and a half seconds to valid.

Graphical user interface, text, application, email

Description automatically generated

* So status changes and value changes.
* Two nice hooks, observables, you can listen to it or you can subscribe to if you really want to closely watch what happens in your form or again, on individual control, and want to react to that.