**96. Using HostListener to Listen to Host Events**

* -: So we saw two examples for directives now.
* And the better approach definitely is a good approach.
* But it's not very interactive, right? It always gives us a blue background.
* And I want to change this.

**Style if I hover the element**

* I only want to style this blue, or give this a blue background, if I hover over it.
* And if I move my mouse away it should go back to transparent.

**@HostListener**

* *Host listener's just a convenient way of listening to events on that element.*
* So let's improve the better highlight here a bit.
* We need to react to some events occurring on the element the directive sits on.
* And a quick and easy way to do this inside of this directive.
* Is to simply add a new decorator.
* And this is the *host listener decorator.*
* (keyboard tapping) Which needs to be imported from at angular core.
* (mouse clicking) And added to some method we want to execute.
* So the method here could be mouse over.
* (keyboard tapping) Now I will move this from the top of the file below ngOnInit here.
* (mouse clicking) So now this can be triggered whenever some event occurs.
* And that event is specified here as an argument as a string.
* Host listener here takes the argument name as an input and that would be mouse enter let's say.
* (keyboard tapping) That is one of the events supported by the dom element this directive sits on.
* So you have basically all events available.
* You could also use with event binding before.
* So that's my host listener targeting this event.
* And we could also receive the event data here.
* So event data of type event (Keyboard tapping) would be passed to us here.
* So that works.
* (mouse clicking) You can also listen to custom events here and retrieve that data.
* So that's just like the method you execute when you add a click listener or whatever your event is.
* And then pass the method between the quotation marks.
* So that's happening here.
* *Host listener's just a convenient way of listening to events on that element.*

Text

Description automatically generated

* So with that we listen to the mouse entry event.
* We get the event data which I don't need here.
* But what I want to do in this case is I want to change the background color of the element.
* So what I can do is I can copy this code from ngOnInit and comment it out.
* And now set the style here on mouse enter.
* And now I can quickly copy this and add another method.
* (mouse clicking) Which I'll name mouse leave maybe.
* (mouse clicking) Where I will listen to the mouse leave event another official event I can listen to.
* And then I want to set the background color to transparent maybe.
* (keyboard tapping) Now with this in place we should have a reactive directive.
* So now no background color (mouse clicking) and if I hover over it, it gets blue.
* So that is working as intended.
* Now with host listener reacting to user event or to any events.