**72. More on View Encapsulation:**

**Overriding View Encapsulation:**

* -: In the last lecture you learned how angular encapsulates your styles.
* Now you can override this encapsulation though.

Text

Description automatically generated

* If you go back to your code to the server element component you can add something to the ad component decorator.
* It's called encapsulation this property.
* And there as a value you can access a view encapsulation which needs to be imported from at angular core.
* So make sure to add this import and then you can choose between free modes, emulate it as the default.
* So you don't need to add this.
* None, If you add this, would makes sure that if we have a look at our component again and we inspect our server element, that now in here we don't see these strange attributes being added to our elements.
* Therefore this component now does not use view encapsulation.
* The other components still will use it there.
* You still see these attributes but if you now define any styles for this component in these CSS file of this component they will actually get up light globally.
* And I can demonstrate this by going into this CSS file.
* And if I change the label color and the label is in the cockpit not in this component to red, yeah now you will see the label there is overwritten.
* Whilst this label of course still has its custom unique attribute it still is a label in the end.
* And in this server element component we disable encapsulation.
* So there our selectors aren't changed by angular.
* They don't receive their unique selector therefore they are enforced they are used application wide, also affecting other components.
* Might be the behavior you want, chances are that this is not the case but it's important to know that you can change this behavior.
* Now besides None you also could choose native and native uses the ShadowDom technology.
* This should give you the same result as before with emulated but only in browsers which supported.
* Which is why for most cases you want to choose emulated but be aware that you could switch to None or Native year two.
* So I will go back to emulated, which again you wouldn't have to add.
* It's a default anyways.
* With that our label should no longer be red now.
* So that is how you can change it and how view encapsulation works.
* A nice tool ensuring that by default only your component receives the styles you define for it.
* But as you learned it can be overwritten.

**Encapsulation Types:**

**None:**

No view encapsulation will be there.

**Emulated:**

It is default. Where component has view encapsulation.

**Native:**

Supports shadow dom technology. Same like emulated.