**What’s the difference between [style] and [ngStyle] in Angular?**

When you need to apply dynamic styles to a HTML element using Angular, there are different options to consider.

The first and most obvious solution is to use the regular style or class HTML attributes along with Angular data-bindings, which looks like this:

<div [style.color]="errorMessageColor">  
 The phone number you entered does not match the expected format  
</div>

The above code would style the div using a color value set on the errorMessageColor property of your component. Thanks to the data binding, whenever that property changes, the div would get a new color.

You can make the above example even more powerful using the ternary operator in your expression to express a conditional styling:

<div [style.color]="hasError ? 'red' : 'black' ">  
 The phone number you entered does not match the expected format  
</div>

The above example assumes that hasError can evaluate to true, which is convenient since in Javascript anything that is not null or undefined would work. As a result, if hasError is something that isn’t null or undefined or false, then the message will show up in red color, otherwise it will be black.

This is great but it can get quite verbose if you want to apply multiple styles to the same element:

<div [style.color]="hasError ? 'red' : 'black' " [style.font]="font" [style.background-color]="hasError ? 'tomato' : 'white' ">  
 The phone number you entered does not match the expected format  
</div>

In that case, using a CSS class is a much better option as you can refactor those CSS properties in one place and possibly reuse them on other HTML elements.

Sometimes, you might need very specific CSS properties based on different conditions though. That’s where ngStyle comes into play. With ngStyle, you can bind to an object that expresses as many conditions and cases as you need:

<div [ngStyle]="currentStyles">  
 The phone number you entered does not match the expected format   
</div>

And then in your component code:

this.currentStyles = {   
 'font-style': this.canSave ? 'italic' : 'normal',   
 'color': this.hasError ? 'red' : 'black',   
 'font-size': this.hasError ? '24px' : '12px'   
};

Now you can express as many different styles as needed. ngStyle is an Angular directive that gives you the flexibility to do this, where as style is a regular HTML property to which you can only bind values one by one. That’s the difference between the two of them.

Note that the same distinction of behavior applies between class and ngClass, where the former is meant to be used for single bindings, and the latter can be used to bind to a decision object similar to the one showed above for ngStyle.