

Angular ngStyle Directive

[Leave a Comment](#) / [3 minutes of reading](#) / [March 9, 2023](#)

[← ngClass Directive](#)

[Angular Tutorial](#)

[ngFor Trackby →](#)

The Angular ngStyle directive allows us to set the many inline style of a HTML element using an expression. The expression can be evaluated at run time allowing us to dynamically change the style of our HTML element. In this tutorial, we learn how to use the ngStyle with an example.

Table of Contents

[ngStyle Syntax](#)

[ngStyle Example](#)

[Change Style Dynamically](#)

[ngStyle multiple attributes](#)

[Specifying CSS Units in ngStyle](#)

[Using object from Controller](#)

[Change Style using Style Property Binding](#)

ngStyle Syntax

```
1  
2 <element [ngStyle]="{'styleNames': styleExp}">...</element>  
3
```

Where

element is the DOM element to which style is being applied

styleNames are style names (ex: 'font-size', 'color' etc). with an optional suffix (ex: 'top.px', 'font-style.em'),

styleExp is the expression, which is evaluated and assigned to the styleNames

We can add more than one key value pairs 'styleNames': styleExp each separated by comma.

In the following example, some-element gets the style of font-size of 20px .

```
1  
2 <some-element [ngStyle]="{'font-size': '20px'}">Set Font size to 20px</some-element>  
3
```

The units (for example px , em) are prefixed to the styleName .

```
1  
2 Syntax:  
3 <element [ngStyle]="{'styleName.unit': widthExp}">...</element>  
4  
5 Example:  
6 <some-element[ngStyle]="{'font-size.em': '3'}">...</some-element>
```

ngStyle Example

Change Style Dynamically

Initialize a variable `color` and add it to your component

```
1  
2 color: string= 'red';  
3
```

And in your template, add the following

```
1  
2 <input [(ngModel)]="color" />  
3 <div [ngStyle]="{'color': color}">Change my color</div>  
4
```

In the above example, we apply `ngStyle` directive to the `div` element. We assign JavaScript object `{'color': color}` to the `ngStyle` directive. The variable `color` is dynamically changed by the user input and it is applied instantly to the `div` element

The following code uses the ternary operator to set the background color to **red** if the `status` variable indicator is set to “error” else **blue**.

```
1  
2 div [ngStyle]="{'background-color':status === 'error' ? 'red' : 'blue' }"></div>  
3
```

ngStyle multiple attributes

We can change multiple style as shown in the following example

```
1
2 <p [ngStyle]="{'color': 'purple',
3               'font-size': '20px',
4               'font-weight': 'bold'}">
5   Multiple styles
6 </p>
7
8
```

The JavaScript object is assigned to the `ngStyle` directive containing multiple properties. Each property name of the object acts as a class name. The value of the property is the value of the style.

Specifying CSS Units in ngStyle

CSS has several units for expressing a length, size etc. The units can be `em`, `ex`, `%`, `px`, `cm`, `mm`, `in`, `pt`, `PC` etc. We prefix the units to the `StyleName` as shown below.

```
1
2 <input [(ngModel)]="size" />
3 <div [ngStyle]="{'font-size.px': size}">Change my size</div>
4
```

Using object from Controller

Create a class as shown below

```
1
2 class StyleClass {
3   'color': string = 'blue';
4   'font-size.px': number = 20;
5   'font-weight': string = 'bold';
6 }
7
```

And in controller initialize the class

```
1
2 styleClass: StyleClass = new StyleClass();
3
```

Then you can refer it in your template as shown below

```
1
2 <div [ngStyle]="styleClass">size & Color</div>
3
```

Change Style using Style Property Binding

We can also change the Styles using the [Style Binding](#).

Further Reading on Angular Styles

- [Angular ngStyle Directive](#)
- [ngClass Directive](#)
- [Angular Global Styles](#)
- [Angular Component Styles](#)
- [View Encapsulation in Angular](#)
- [Angular Directives](#)