Class and style binding

Use class and style bindings to add and remove CSS class names from an element's class attribute and to set styles dynamically.

Prerequisites

- Property binding

Binding to a single CSS class

To create a single class binding, type the following:

[class.sale]="onSale"

Angular adds the class when the bound expression, onSale is truthy, and it removes the class when the expression is falsy—with the exception of undefined. See styling delegation for more information.

Binding to multiple CSS classes

To bind to multiple classes, type the following:

[class]="classExpression"

The expression can be one of:

- A space-delimited string of class names.
- An object with class names as the keys and truthy or falsy expressions as the values.
- An array of class names.

With the object format, Angular adds a class only if its associated value is truthy.

With any object-like expression—such as object, Array, Map, or Set —the identity of the object must change for Angular to update the class list. Updating the property without changing object identity has no effect.

If there are multiple bindings to the same class name, Angular uses styling precedence to determine which binding to use.

The following table summarizes class binding syntax.

BINDING TYPE	SYNTAX	INPUT TYPE	EXAMPLE INPUT VALUES
Single class binding	[class.sale]="onSale"	boolean | undefined | null	true, false

BINDING TYPE	SYNTAX	INPUT TYPE	EXAMPLE INPUT VALUES
Multi-class binding	[class]="classExpression"	string	"my-class-1 my-class-2 my-class-3"
Multi-class binding	[class]="classExpression"	Record<string, boolean | undefined | null>	{foo: true, bar: false}
Multi-class binding	[class]="classExpression"	Array<string>	['foo', 'bar']

Binding to a single style

To create a single style binding, use the prefix style followed by a dot and the name of the CSS style.

For example, to set the width style, type the following: [style.width]="width"

Angular sets the property to the value of the bound expression, which is usually a string. Optionally, you can add a unit extension like em or %, which requires a number type.

1. To write a style in dash-case, type the following:

```
content_copy<nav [style.background-color]="expression"></nav>
```

2. To write a style in camelCase, type the following:

```
content_copy<nav [style.backgroundColor]="expression"></nav>
```

Binding to multiple styles

To toggle multiple styles, bind to the [style] attribute—for example, [style]="styleExpression".
The styleExpression can be one of:

- A string list of styles such as "width: 100px; height: 100px; background-color: cornflowerblue;".
- An object with style names as the keys and style values as the values, such as {width: '100px', height: '100px', backgroundColor: 'cornflowerblue'}.

Note that binding an array to [style] is not supported.

When binding [style] to an object expression, the identity of the object must change for Angular to update the class list. Updating the property without changing object identity has no effect.

Single and multiple-style binding example

```
nav-bar.component.ts

content_copy@Component({

 standalone: true,

 selector: 'app-nav-bar',

 template: ` <nav [style]="navStyle">

  <a [style.text-decoration]="activeLinkStyle">Home Page</a>

  <a [style.text-decoration]="linkStyle">Login</a>

 </nav>`,

})

export class NavBarComponent {

 navStyle = 'font-size: 1.2rem; color: cornflowerblue;';

 linkStyle = 'underline';

 activeLinkStyle = 'overline';

 /* . . . */

}
```

If there are multiple bindings to the same style attribute, Angular uses styling precedence to determine which binding to use.

The following table summarizes style binding syntax.

BINDING TYPE	SYNTAX	INPUT TYPE	EXAMPLE INPUT VALUES
Single style binding	[style.width]="width"	string | undefined | null	"100px"
Single style binding with units	[style.width.px]="width"	number | undefined | null	100
Multi-style binding	[style]="styleExpression"	string	"width: 100px; height: 100px"

BINDING TYPE	SYNTAX	INPUT TYPE	EXAMPLE INPUT VALUES
Multi-style binding	[style]="styleExpression"	Record<string, string | undefined | null>	{width: '100px', height: '100px'}

Styling precedence

A single HTML element can have its CSS class list and style values bound to multiple sources (for example, host bindings from multiple directives).

(This article is sourced from: https://angular.io/guide/class-binding and the content of the article is here in case of link breakage.)