Style Precedence

When there are multiple bindings to the same class name or style attribute, Angular uses a set of precedence rules to determine which classes or styles to apply to the element. These rules specify an order for which style and class related bindings have priority. This styling precedence is as follows, from the most specific with the highest priority to least specific with the lowest priority:

1. Template bindings are the most specific because they apply to the element directly and exclusively, so they have the highest precedence.

Binding type	Example
Property binding	`<div [class.foo]="hasFoo">` `<div [style.color]="color">`
Map binding	`<div [class]="classExpression">` `<div [style]="styleExpression">`
Static value	`<div class="foo">` `<div style="color: blue">`

2. Directive host bindings are less specific because you can use directives in multiple locations, so they have a lower precedence than template bindings.

Binding type	Example
Property binding	`host: {'[class.foo]': 'hasFoo'}` `host: {'[style.color]': 'color'}`
Map binding	`host: {'[class]': 'classExpr'}` `host: {'[style]': 'styleExpr'}`
Static value	`host: {'class': 'foo'}` `host: {'style': 'color: blue'}`

3. Component host bindings have the lowest precedence.

Binding type	Example
Property binding	`host: {'[class.foo]': 'hasFoo'}` `host: {'[style.color]': 'color'}`
Map binding	`host: {'[class]': 'classExpression'}` `host: {'[style]': 'styleExpression'}`
Static value	`host: {'class': 'foo'}` `host: {'style': 'color: blue'}`

Precedence and specificity

In the following example, binding to a specific class, as in `[class.special]`, takes precedence over a generic `[class]` binding. Similarly, binding to a specific style, as in `[style.color]`, takes precedence over a generic `[style]` binding.

Precedence and bindings from different sources

Specificity rules also apply to bindings even when they originate from different sources. An element can have bindings that originate from its own template, from host bindings on matched directives, and from host bindings on matched components.

Precedence of bindings and static attributes

Bindings take precedence over static attributes because they are dynamic. In the following case, `class` and `[class]` have similar specificity, but the `[class]` binding takes precedence.

{@a styling-delegation}

Delegating to styles with lower precedence

Higher precedence styles can defer to lower precedence styles using `undefined` values. For example, consider the following template:

Imagine that the `dirWithHostBinding` directive and the `comp-with-host-binding` component both have a `[style.width]` host binding.

If `dirWithHostBinding` sets its binding to `undefined`, the `width` property falls back to the value of the `comp-with-host-binding` host binding.

@HostBinding('style.width') width = ''; // undefined

If `dirWithHostBinding` sets its binding to `null`, Angular removes the `width` property entirely.

@HostBinding('style.width') width = null;