



# Global styles

## ON THIS PAGE



## :global(...)

To apply styles to a single selector globally, use the `:global(...)` modifier:

```
<style>
  :global(body) {
    /* applies to <body> */
    margin: 0;
  }

  div :global(strong) {
    /* applies to all <strong> elements, in any component,
       that are inside <div> elements belonging
       to this component */
    color: goldenrod;
  }

  p:global(.big.red) {
    /* applies to all <p> elements belonging to this component
       with `class="big red"`, even if it is applied
       programmatically (for example by a library) */
  }
</style>
```



If you want to make @keyframes that are accessible globally, you need to prepend your keyframe names with `-global-`.

The `-global-` part will be removed when compiled, and the keyframe will then be referenced using just `my-animation-name` elsewhere in your code.



```
keyframes -global-my-animation-name {  
  /* code goes here */  
}  
</style>
```

## :global

To apply styles to a group of selectors globally, create a `:global {...}` block:

```
<style>  
:global {  
  /* applies to every <div> in your application */  
  div { ... }  
  
  /* applies to every <p> in your application */  
  p { ... }  
}  
  
.a :global {  
  /* applies to every `.b .c .d` element, in any component,  
   that is inside an `.a` element in this component */  
  .b .c .d {...}  
}  
</style>
```

The second example above could also be written as an equivalent `.a :global .b .c .d` selector, where everything after the `:global` is unscoped, though the nested form is preferred.

[✎ Edit this page on GitHub](#)

PREVIOUS

Scoped styles

NEXT

Custom properties