

提高CSS的编写效率，可以多用这三个CSS伪类

:where

基本使用

`:where()` CSS 伪类函数接受选择器列表作为它的参数，将会选择所有能被该选择器列表中任何一条规则选中的元素。

以下代码，文本都会变成 yellow 颜色

```
1 :where(div p) span {
2     color: yellow;
3 }
4
5 <div class="test-div">
6     <span>哈哈</span>
7 </div>
8 <p class="test-p">
9     <span>哈哈</span>
10 </p>
```

使用场景

其实 `:where()` 的功能本来就有，只不过有了它之后，实现起这些功能来就更加方便快捷~接下来就来讲讲它的组合/叠加功能

我们来看下面的这段 css 代码

```
1 div a:hover,
2 li a:hover,
3 .cla a:hover,
4 .aa .bb a:hover,
5 [class^='bold'] a:hover{
6     color: yellow;
7 }
```

我们可以使用 `:where()` 来简化这个写法，使用它找出 div li .cla 这三种选择器，选择器可以是标签，也可以是类名，也可以是选择器表达式

```
1 :where(div, li, .cla, .a .b, [class^='bold']) a:hover {
2     color: yellow;
3 }
```

再看看使用 `:where()` 的组合，完成一些功能，我们看以下的代码

```
1 .dark-theme button,
2 .dark-theme a,
3 .light-theme button,
4 .light-theme a{
5     color: pink;
6 }
```

我们完全可以使用 `:where()` 简化这个写法

```
1 :where(.dark-theme, light-theme) :where(button, a) {
2     color: pink;
3 }
```

优先级

`:where()` 的优先级是 0，我们可以看下面代码

```
1 .test {
2     color: yellow;
3 }
4 :where(.test) {
5     color: pink
6 }
```

最后字体颜色是 yellow

兼容性

全绿~

浏览器兼容性

[Report problems with this compatibility data on GitHub](#)

	🖥️					📱					
	Chrome 🌐	Edge 🌐	Firefox 🌐	Opera 🌐	Safari 🌐	Chrome Android 🌐	Firefox for Android 🌐	Opera Android 🌐	Safari on iOS 🌐	Samsung Internet 🌐	WebView Android 🌐
<code>:where()</code>	✓ 88	✓ 88	✓ 78	✓ 74	✓ 14	✓ 88	✓ 79	✓ 63	✓ 14	✓ 15.0	✓ 88
Support for forgiving selector list	✓ 88	✓ 88	✓ 82	✓ 74	✓ 14	✓ 88	✓ 82	✓ 63	✓ 14	✓ 15.0	✓ 88

Tip: you can click/tap on a cell for more information.

:is

`:is()` 跟 `:where()` 可以说一模一样，区别就是 `:is()` 的优先级不是0，而是由传入的选择器来决定的，拿刚刚的代码来举个例子

```
1 div {
2   color: yellow;
3 }
4 :where(.test) {
5   color: pink
6 }
7
8 <div class="test">哈哈</div>
```

这要是 `:where()`，那么字体颜色会是 yellow，因为它的优先级是 0

但是如果是 `:is()` 的话，字体颜色会是 pink，因为 类选择器 优先级比 标签选择器 优先级高~

```
1 :is(.test) {
2   color: pink
3 }
4 div {
5   color: yellow;
```



```
6 }
7
8 <div class="test">哈哈</div>
```

兼容性

全绿~

浏览器兼容性

[Report problems with this compatibility data on GitHub](#) 

											
	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android
<code>:is()</code>	✓ 88 ...	✓ 88 ...	✓ 78 ...	✓ 74 ...	✓ 14 ...	✓ 88 ...	✓ 79 ...	✓ 63 ...	✓ 14 ...	✓ 15.0 ...	✓ 88 ...
Support for forgiving selector list	✓ 88	✓ 88	✓ 82	✓ 74	✓ 14	✓ 88	✓ 82	✓ 63	✓ 14	✓ 15.0	✓ 88

Tip: you can click/tap on a cell for more information.

:has

基本使用

举一个场景例子，我们看以下代码，一个容器中，图片是可以显隐的，我想要实现：

- 图片显示时，字体大小为 12px
- 图片隐藏时，字体大小为 20px

```
1 <div class="container">
2   哈哈哈哈哈
3   <img class="test-img" v-if="showImg"></img>
4 </div>
```

如果按照以前的做法，就是使用 动态class 的方式去玩完成这个功能，但是现在有 `:has()` 可以通过 css 的方式去完成这件事~

```
1 .container {
2     font-size: 20px;
3 }
4 .container:has(img) {
5     font-size: 12px;
6 }
7
8 或者
9 .container:has(.test-img) {
10     font-size: 12px;
11 }
```

组合使用

现在又有两个场景

- 判断容器有没有**子img**，有的话字体设置为 12px（上面的例子是后代选择器，不是子选择器）
- 判断容器有没有一个小相邻的img，有的话设置字体颜色为 red

我们可以这么去实现：

```
1 .container:has(>img) {
2     font-size: 12px;
3 }
4
5 .container:has(+img) {
6     color: red;
7 }
```

再来一个场景，当我 hover 到 子img 上时，我想要让 container 的字体变粗，可以这么去使用~

```
1 .container:has(>img:hover) {
2     color: red;
3 }
```

兼容性

还是有一些浏览器不支持

浏览器兼容性

[Report problems with this compatibility data on GitHub](#)

	<div>🖥️</div>					<div>📱</div>					
	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android
<code>:has()</code>	✓ 105	✓ 105	✗ 103 🚩	✓ 91	✓ 15.4	✓ 105	✗ No	✓ 72	✓ 15.4	✓ 20.0	✓ 105

Tip: you can click/tap on a cell for more information.