

Vue3中deep样式穿透的使用细节及源码解析

Vue3 中可以在 style 标签中使用 :deep() 的方法进行样式穿透，主要是解决在使用第三方的 UI 库（如 element-plus 等）时导致的对其样式设置不生效的问题。

1. scoped属性

在介绍deep样式穿透之前，我们先看一下scoped属性。

在一个 Vue3 组件的 style 标签中可以对其设置 scoped 属性 `<style scoped>`，`</style>`。该属性的作用主要有以下三个方面：

- [Vue 会为该组件 template 中定义的 HTML 元素中添加一个 data-v-{一个hash值} 的属性选择器]
- [在 style 标签中写的 CSS 样式 Vue 都会在样式选择器的最后段添加上 data-v-{一个hash值} 的属性标签]
- [而在使用第三方的 UI 库时，只会为根元素添加 data-v-{一个hash值} 属性，而子元素则不会添加]

说的简单一点，其实就是在 HTML 的标签下添加了一个 属性选择器，可以根据这个属性选择器来保证各个组件的样式不被相互污染，从而实现一个模块化的效果。

举个栗子（element-plus）：

```
1 <template>
2   <div class="main">
3     <el-input class="ipt"></el-input>
4   </div>
5 </template>
6
7 <script setup></script>
8
9 <style scoped>
10   .ipt {
11     width: 300px;
12   }
13 </style>
14
```

在这个栗子中，主要使用了一个 `el-input` 标签，并为其设置了一个宽度。

```

<body>
  <div id="app" data-v-app>
    <div data-v-7a7a37b1 class="main">
      <div data-v-7a7a37b1 class="el-input ipt">
        <!-- input -->
        <!-- prepend slot -->
        <!--v-if-->
        <div class="el-input__wrapper" tabindex="-1">
          <!-- append slot -->
          <!--v-if-->
        </div>
      </div>
    </div>
  </div>

```

那么随之而来就产生一个问题，当使用以下方法修改样式时并不能生效：

```

1  .ipt .el-input__wrapper {
2    background-color: red;
3  }

```

但该选择器确实是添加进去了。

出现这种结果的原因就在于 Vue 将 `[data-v-7a7a37b1]` 属性添加到 `.el-input__wrapper` 之后，而 `.el-input__wrapper` 的标签上并不存在 `[data-v-7a7a37b1]` 属性。那么 deep 样式穿透随之而来。

2. deep 样式穿透

使用方法

```

1  :deep(.ipt .el-input__wrapper) {
2    background-color: red;
3  }

```

```

<!--VITE + VUE + CSS-->
<style type="text/css" data-vite-dev-id="D:/WorkPlace/jt...
  .ipt[data-v-7a7a37b1] {
    width: 300px;
  }
  [data-v-7a7a37b1] .ipt .el-input__wrapper {
    background-color: red;
  }
</style>

```

`:deep()` 函数会把属性选择器放在最前面，那么就可以捕获到啦！

3. 源码解析

目录：core-main/packages/compiler-sfc/src/compileStyle.ts

```

1   export function doCompileStyle(
2     scoped = false,
3   ): SFCStyleCompileResults | Promise<SFCStyleCompileResults> {
4     .....
5     if (scoped) {
6       plugins.push(scopedPlugin(longId))
7     }
8     .....
9   }

```

在这个函数中，如果存在 `scoped` 属性，就会调用 `postcss` 这个插件，这个插件的主要作用就是把 CSS 转换成抽象语法树 (AST) 便于之后的操作。

```

1   function processRule(id: string, rule: Rule) {
2     .....
3     rule.selector = selectorParser(selectorRoot => {
4       selectorRoot.each(selector => {
5         rewriteSelector(id, selector, selectorRoot)
6       })
7     }).processSync(rule.selector)
8   }

```

之后在 `processRule` 函数中调用 `rewriteSelector()` 方法对 CSS 选择器进行重写。

```

1   function rewriteSelector(
2     id: string,
3     selector: selectorParser.Selector,
4     selectorRoot: selectorParser.Root,
5     slotted = false,
6   ) {
7     let node: selectorParser.Node | null = null
8     let shouldInject = true
9     // find the last child node to insert attribute selector
10    selector.each(n => {
11      .....
12      if (n.type === 'pseudo') {
13        const { value } = n
14        // deep: inject [id] attribute at the node before the ::v-deep
15        // combinator.
16        if (value === ':deep' || value === '::v-deep') {
17          if (n.nodes.length) {
18            // .foo ::v-deep(.bar) -> [xxxxxxx] .foo .bar

```

```

19         // replace the current node with ::v-deep's inner selector
20         let last: selectorParser.Selector['nodes'][0] = n
21         n.nodes[0].each(ss => {
22             selector.insertAfter(last, ss)
23             last = ss
24         })
25         // insert a space combinator before if it doesn't already have
one
26         const prev = selector.at(selector.index(n) - 1)
27         if (!prev || !isSpaceCombinator(prev)) {
28             selector.insertAfter(
29                 n,
30                 selectorParser.combinator({
31                     value: ' ',
32                 }),
33             )
34         }
35         selector.removeChild(n)
36     } else {
37         .....
38     }
39     .....
40 }

```

当遇到 `:deep` 时，会将原来的属性选择器添加到前面元素中，即：`.foo ::v-deep(.bar) -> [xxxxxxx]`
`.foo .bar` 通过这种方法就能定位到第三方 UI 库中的 CSS 选择器了。

结语

在 Vue3 中，当使用一些第三方的 UI 库时，由于 Vue3 实现了模块化封装，那么再设置 UI 库的 CSS 样式时有时会出现设置不成功的问题，那么这个时候可以考虑使用 `:deep()` 来进行样式穿透。