

CSS Modules 与 scoped 的不一样

原创：邱鸿 大转转FE 昨天

What?

css 的作用域表现。

CSS Modules

是一个CSS文件，其中所有类名和动画名称默认为局部作用域。

使用JS编译原生的CSS文件，使其具备模块化的能力，该文件需要import使用。

Scoped

在vue文件中的style标签上，有一个特殊的属性：scoped。

此样式仅适用于当前组件元素，从而使组件之间的样式不互相污染。

How?

Css modules

实现原理

通过给样式名加hash字符串后缀的方式，实现特定作用域语境中的样式编译后的样式在全球唯一。

具体效果demo

```
// webpack.config.js
{
```

```
test: /\.css$/,
loader: 'style!css-loader?modules&importLoaders=1&localIdentName=[name]__[local]__[hash:base64:5]',
}
```

编译前 - vue 环境

```
<template>
  <div :class="$style.green">demo demo</div>
</template>

<style module>
.green {
  color: green;
}
</style>
```

编译后 - vue 环境

```
<div class="green_3UI7s9iz">demo demo</div>

.green_3UI7s9iz {
  color: green;
}
```

由此可见

1. css module直接替换了类名，排除了用户设置类名影响组件样式的可能性。优点就是不必再担心命名约定。
2. \$style.green 是个变量，即可在js中引用，引用方式为：this.\$style.green。
3. modules 即为启用，localIdentName 是设置生成样式的命名规则。

应用场景

只应用于某个组件，其他组件不适用。

⚠ 注意点

- 在处理动画animation的关键帧keyframes，动画名称必须先写：
- 比如，animation: deni .5s,能正常编译; animation: .5s deni, 则编译异常

- css modules生效方式：
- 需要css-loader配置才能生效。
- 若使用的是style-loader，则需配置更换为vue-style-loader才可生效。

css modules如何解决权重问题？

允许通过重命名或命名空间来封装样式规则，减少对选择器的约束，从而达到不需要特定方法就可舒服的使用类名。

当样式规则耦合到每个组件时，当不再使用组件时，样式也会被移除。

Scoped

实现原理

vue通过在DOM结构以及css样式上加唯一不重复的标记，以保证唯一，达到样式私有化模块化的目的。

具体效果demo

```
// 编译前 demo.vue
<template>
  <div class="demo-c">hello world!</div>
</template>

<style lang="less" scoped>
  .demo-c {
    width: 100%;
    height: 100px;
    background-color: green;
    color: #fff;
  }
</style>

// 编译后-dom
<div data-v-48baf84c="" class="demo-c">hello world!</div>
// 编译后-css
```

```
.demo-c[data-v-48baf84c] {  
  width: 100%;  
  height: 1.333333rem;  
  background-color: green;  
  color: #fff;  
}
```

应用场景

scoped css可以直接在能跑起来的vue项目中使用，且对应的样式只对该组件有效，不被其他组件污染。

⚠ 注意点

“权重加重”的意思：如果我们要去修改这个样式，需要更高的权重去覆盖其样式。

- 由于css样式优先级的特性，scoped处理会造成每个样式的权重加重了。
- 若组件内部包含有子组件，只会给予组件的最外层标签加上当前组件的data属性：
- 一般父组件如果加了scoped，会比已设置过自己样式的子组件内除最外层标签的内层标签的权重低，所以不会影响他们的样式。
- scoped会使标签选择器渲染变慢更多倍。
- 若用户再定义了相同的类名，也有可能影响到组件的样式。
- 具有scoped属性的样式只会应用到当前元素和其子元素。Inline样式仍然比scoped样式优先级高，所以，最好避免使用inline样式。

Why?

目的

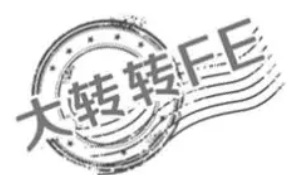
- css 模块化
- css 私有化：不被污染
- css 复用性
- 解决CSS中的全局作用域问题

总结

实现同一目的，比较它们的优缺点，css modules 配置并不麻烦且实现的整体效果更优于scoped css，在此个人更推荐使用css modules。

参考资料

- <https://github.com/css-modules/css-modules>
- <https://cn.vuejs.org/v2/guide/comparison.html%E7%BB%84%E4%BB%B6%E4%BD%9C%E7%94%A8%E5%9F%9F%E5%86%85%E7%9A%84-CSS>
- <https://vue-loader.vuejs.org/guide/css-modules.html#usage>
- <https://github.com/css-modules/webpack-demo>
- <https://webdesign.tutsplus.com/tutorials/solve-your-specificity-headaches-with-css-modules--cms-28973>



一个有趣的前端团队



长按二维码关注
大转转FE