

```

//测试页面
<template>
  <view>
    <button class="test-btn">测试按钮</button>
    <view class="test">
      <eForm />
    </view>
  </view>
</template>
//测试子组件
<template>
  <view>
    <button class="test-btn">测试按钮</button>
    <view class="test">
      <eForm />
    </view>
  </view>
</template>
//测试子组件的子组件
<template>
  <view>
    <button class="test-btn">测试按钮</button>
  </view>
</template>

```

测试一 页面添加非scoped的样式

```

//页面style
<style lang="scss">
  .test-btn{
    background-color: #00ff00;
  }
</style>

```

小程序同app:



h5:



h5默认style启用scoped，所以页面的样式不会影响到子组件的样式

测试二 页面和子组件都添加非scoped样式

```
//页面style
<style lang="scss">
  .test-btn{
    background-color: #00ff00;
  }
</style>
//子组件style
<style lang="scss">
  .test-btn{
    background-color: #0000ff;
  }
</style>
```

小程序同app:



h5:



小程序和APP、H5子组件添加的样式只作用于自身，子组件的子组件样式还是由页面的样式决定。（有无加scoped结果一样）

测试三 子组件添加/deep/

```
//页面style
<style lang="scss">
  .test-btn{
    background-color: #00ff00;
  }
</style>
//子组件style
<style lang="scss">
  .test-btn{
    background-color: #0000ff;
  }
  .test{
    /deep/ .test-btn{
      background-color: #0000ff;
    }
  }
</style>
```

小程序：



h5同app:



小程序/deep/的类选择器不起作用，APP和H5端生效，APP端将页面的非scoped覆盖

测试四 将/deep/的类选择器改成标签选择器

```
//页面style
<style lang="scss">
  .test-btn{
    background-color: #00ff00;
  }
</style>
//子组件style
<style lang="scss">
  .test-btn{
    background-color: #0000ff;
  }
  .test{
    /deep/ button{
      background-color: #0000ff;
    }
  }
</style>
```

小程序、H5、APP相同：



对于小程序/deep/ 标签选择器生效

测试五 页面添加scoped并使用/deep/

```
//页面style
<style lang="scss" scoped>
  /deep/.test-btn{
    background-color: #00ff00;
  }
</style>
//子组件style
<style lang="scss">
  .test-btn{
    background-color: #0000ff;
  }
</style>
```

小程序同app同H5:

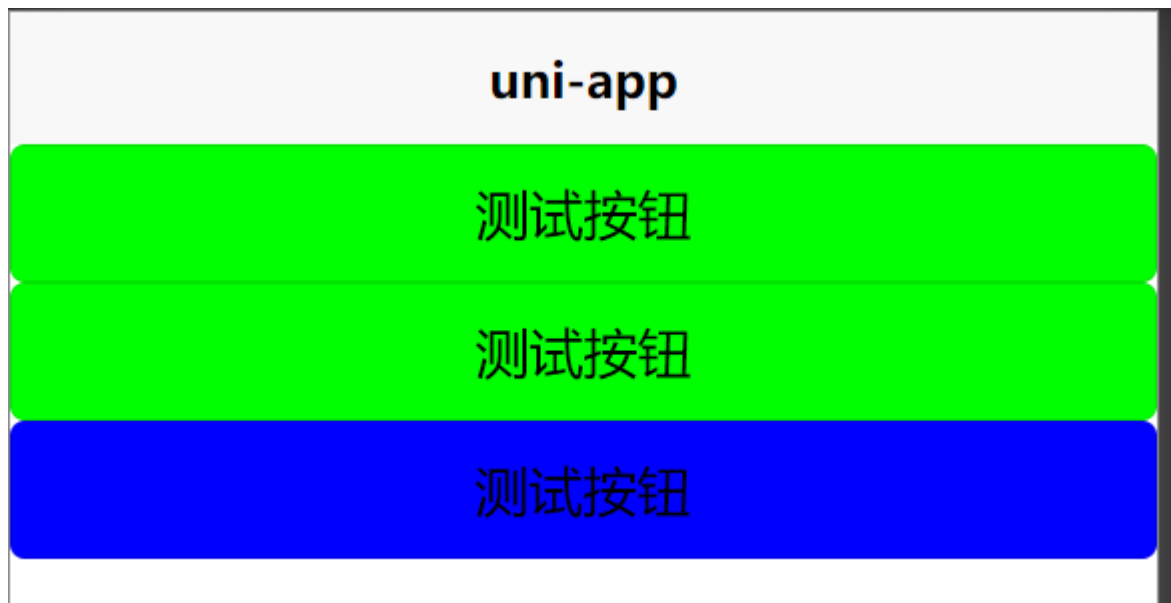


加了/deep/后页面的样式将覆盖所有子组件的样式

测试六 页面和子组件都加/deep/ 同时子组件添加scoped

```
//页面style
<style lang="scss" scoped>
  /deep/.test-btn{
    background-color: #00ff00;
  }
</style>
//子组件style
<style lang="scss" scoped>
  .test-btn{
    background-color: #0000ff;
  }
  .test{
    /deep/ .test-btn{
      background-color: #0000ff;
    }
  }
</style>
```

H5同APP:



小程序:



测试七 页面和子组件都加/deep/ 子组件不添加scoped

```
//页面style
<style lang="scss" scoped>
  /deep/.test-btn{
    background-color: #00ff00;
  }
</style>
//子组件style
<style lang="scss">
  .test-btn{
    background-color: #0000ff;
  }
  .test{
    /deep/ .test-btn{
```



```
background-color: #0000ff;
    }
}
</style>
```

H5和APP同上

小程序:



h5和APP子组件的样式会被直接父组件的/deep/样式覆盖，小程序子组件的样式不加scoped时被页面的/deep/样式影响 加上scoped子组件自身的样式不会被父组件的/deep/样式影响

测试八

```
//页面style
<style lang="scss" scoped>
  .test-btn{
    background-color: #00ff00;
  }
  .test{
    /deep/.test-btn{
      background-color: #ff0000;
    }
  }
</style>
//子组件style
<style lang="scss">
  .test-btn{
    background-color: #0000ff;
  }
  .test{
    /deep/.test-btn{
      background-color: #0000ff;
    }
  }
</style>
```

```
//子组件的子组件样式
<style>
  .test-btn{
    background-color: #0000ff;
  }
</style>
```

小程序同H5同app:



父组件样式覆盖所有子组件的样式，子组件是否添加scoped结果相同

总结：对于scoped H5页面是默认启用使样式不会影响到子组件，使用/deep/可以是父组件的样式穿透到子组件上，子组件的样式由最近的父组件决定

小程序 默认不启用scoped 父组件的样式会直接作用到子组件上，子组件的样式由最外层的父组件（页面）的样式决定，组件的/deep/ 对类选择器无效，标签选择器时生效。小程序页面添加scoped后使用/deep/时对子组件的作用取决于子组件是否带有scoped，有则为子组件自身的样式，没有则父组件的样式覆盖子组件

app 默认不启用scoped 父组件的样式会直接作用到子组件上，子组件的样式由最外层的父组件（页面）的样式决定，加上scoped后使用/deep/时性质基本同H5

结论：H5是默认scoped可以不用手动加，需要在父组件覆盖子组件样式时使用/deep/

小程序和APP的style最好添加scoped属性，需要覆盖子组件样式时可以在页面添加不带scoped的 style去覆盖子组件样式