

1.概述

WXSS (WeiXin Style Sheets)是一套样式语言，用于描述 WXML 的组件样式。

WXSS 用来决定 WXML 的组件应该怎么显示。

为了适应广大的前端开发者，WXSS 具有 CSS 大部分特性。同时为了更适合开发微信小程序，WXSS 对 CSS 进行了扩充以及修改。

---- 新增了尺寸单位

WXSS 在底层支持新的尺寸单位 `rpx`，开发者可以免去换算的烦恼，只要交给小程序底层来换算即可

---- 提供了全局的样式和局部样式

我们可以写一个 `app.wxss` 作为全局样式，会作用于当前小程序的所有页面，局部页面样式 `page.wxss` 仅对当前页面生效

---- 此外 WXSS 仅支持部分 CSS 选择器

2.内联样式

框架组件上支持使用 `style`、`class` 属性来控制组件的样式。

`style`: 静态的样式统一写到 `class` 中。

`style` 接收动态的样式，在运行时会进行解析，

请尽量避免将静态的样式写进 `style` 中，以免影响渲染速度。

```
<view style="color:{{color}};" />
```

`class`: 用于指定样式规则，其属性值是样式规则中类选择器名(样式类名)的集合，
样式类名不需要带上 `.`，样式类名之间用空格分隔。

```
<view class="normal_view" />
```

注意：静态的样式文件建议统一写到 `.wxss` 后缀文件中。

==案例==

wxml

```
<!--pages/users/users.wxml-->
<view>
    <!-- 动态加载的样式 -->
    <view style="color:{{mycolor}}">
        广州
    </view>
    <!-- 静态引用的样式 -->
    <view class="normal">
        深圳
    </view>
</view>
```

wxjs

```
// pages/users/users.js
Page({
    /**
     * 页面的初始数据
     */
    data: {
        mycolor:"#f00"
    },
})
```

WXSS

```
.normal{
    color: blue;
    font-size: 64rpx;
}
```

提示：在wxss中设置颜色，无需添加引号

```

iPhone 5 100% 16
11:46 100% 🔋
••••• WeChat 11:46 100%
1. <!--pages/users/users.wxml-->
2. <view>
3.   <!-- 动态加载的样式 -->
4.   <view style="color:{{mycolor}}">广州
5.   </view>
6.   <!-- 静态引用的样式 -->
7.   <view class="normal">深
8.   </view>
9. </view>
10. // pages/users/users.js
11. Page({
12.   /**
13.    * 页面的初始数据
14.    */
15.   data: {
16.     mycolor: "#f00"
17.   },
18.   /**
19.    * 生命周期函数--监听页面加载
20.    */
21.   onLoad: function(
22.     options) {
23.   }
24. })
25. /* pages/users/users.wxss */
26. /**
27.  *
28.  */
29. .normal{
30.   color: blue;
31.   font-size: 64rpx;
32. }

```

3.选择器

目前支持的选择器有：

选择器	样例	样例描述
.class	.intro	选择所有拥有 class="intro" 的组件
#id	#firstname	选择拥有 id="firstname" 的组件
element	view	选择所有 view 组件
element, element	view, checkbox	选择所有文档的 view 组件和所有的 checkbox 组件
element element	view text	选择view 组件下的 text组件
::after	view::after	在 view 组件后边插入内容
::before	view::before	在 view 组件前边插入内容

==案例==

wxml

```

<!--pages/users/users.wxml-->
<view>
  <!-- 动态加载的样式 -->
  <view style="color:{{mycolor}}">
    广州
  </view>
</view>

<!-- 静态引用的样式 -->
<view class="normal">东莞</view>

```

```
<!-- 选择器使用 -->
<!-- id选择器 -->
<button id="submit_btn">提交</button>
<!-- 分组选择器 -->
<text>姓名: xxx</text>
<!-- 层级选择器 -->

<view>
    <text>层级选择器</text>
</view>

<!-- 全局样式与局部样式 -->
<view class="container user_view c1">我是一个容器</view>
```

WXSS

```
/* pages/users/users.wxss */

/* 类选择器，分组选择器 */
.normal, text{
    color: blue;
    font-size: 64rpx;
}

/* id选择器 */
#submit_btn{
    color: #ff0;
}

/* 层级选择器 */
view text{
    background: pink;
}

.user_view{
    color: red;
    font-size: 20rpx;
}
```

效果

The screenshot shows the WeChat developer tools interface. On the left is the preview window for an iPhone 5 device, displaying a form with a text input containing '姓名: xxx' and two buttons labeled '提交' (Submit) and '取消' (Cancel). The background of the preview is pink. On the right are two code editors. The top editor shows the WXML code for the page, which includes a view with a color style, a view with a class, and a button with an id. The bottom editor shows the corresponding WXSS code, defining styles for various components like .normal, #submit_btn, and view text.

```

users.wxml
1 <!--pages/users/users.wxml-->
2 <view>
3 ...<!-- 动态加载的样式 -->
4 <view style="color:{(mycolor)}">广州
5 </view>
6 </view>
7 ...<!-- 静态引用的样式 -->
8 <view class="normal">东莞</view>
9 <!-- 选择器使用 -->
10 <!-- id选择器 -->
11 <button id="submit_btn">提交</button>
12 <!-- 元素选择器 -->
13 <button>取消</button>
14 <!-- 分组选择器 -->
15 <text>姓名: xxx</text>
16 <!-- 层级选择器 -->
17 <view>
18 ...<text>层级选择器</text>
19 </view>

users.wxss
1 /* pages/users/users.wxss */
2 /* 类选择器，分组选择器 */
3 .normal{text:
4   color: blue;
5   font-size: 64px;
6 }
7 /* id选择器 */
8 #submit_btn{
9   color: #ff0;
10 }
11 /* 元素选择器 标签选择器 */
12 button{
13   margin-top: 30px;
14 }
15 /* 层级选择器 */
16 view text{
17   background: pink;
18 }

```

4. 尺寸单位

rpx(responsive pixel): 可以根据屏幕宽度进行自适应。

规定屏幕宽为 750rpx。

例如在 iPhone6上，屏幕宽度为 375px，共有750个物理像素，则 $750\text{rpx} = 375\text{px} = 750\text{物理像素}$ ， $1\text{rpx} = 0.5\text{px} = 1\text{物理像素}$ 。

设备	rpx 换算 px (屏幕宽度/750)	px 换算 rpx (750/屏幕宽度)
iPhone5	$1\text{rpx} = 0.42\text{px}$	$1\text{px} = 2.34\text{rpx}$
iPhone6	$1\text{rpx} = 0.5\text{px}$	$1\text{px} = 2\text{rpx}$
iPhone6 Plus	$1\text{rpx} = 0.552\text{px}$	$1\text{px} = 1.81\text{rpx}$

建议：开发微信小程序时设计师可以用 iPhone6作为视觉稿的标准。

注意：在较小的屏幕上不可避免的会有一些毛刺，请在开发时尽量避免这种情况

在日后的小程序开中，我们的统一的单位是 **rpx** --- 不要使用 **px** 作为单位

5. 全局样式

定义在 **app.wxss** 中的样式为全局样式，作用于每一个页面

The screenshot shows a code editor interface with a sidebar on the left displaying a project structure. The structure includes a 'pages' folder containing 'common', 'index', 'logs', 'template', and 'user' subfolders, along with 'utils', 'app.js', 'app.json', and 'app.wxss'. The 'app.wxss' file is currently selected and its content is displayed in the main editor area. The code in 'app.wxss' is as follows:

```
1  /**app.wxss**/
2  .container {
3      height: 100%;
4      display: flex;
5      flex-direction: column;
6      align-items: center;
7      justify-content: space-between;
8      padding: 200rpx 0;
9      box-sizing: border-box;
10 }
11
```

6.局部样式

在 `pages` 的 `wxss` 文件中定义的样式为局部样式，只作用在对应的页面，并会覆盖 `app.wxss` 中相同的选择器。