# 微信小程序 - Unit02

# 1.WXSS

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

## • 尺寸单位 -- rpx

rpx(Responsive Pixel),可以根据屏幕宽度进行自适应，规定屏幕尺寸为750rpx

|  |
| --- |
| 手机设备 |
| iPhone5(320/750) |
| iPhone6(375/750) |
| iPhone6 Plus(414/750) |

## • 样式导入

样式导入的语法结构是：

@import "wxss样式文件的路径及名称";

# 2.组件

## • <image>组件

<image>组件是图片组件，支持GIF、JPG、PNG、WEB、SVG格式的文件，语法结构如下：

<image
  
 src="图像的URL地址"
  
 lazy-load="是否懒加载"
  
 mode="图像的裁切、缩放的模式"></image>

<image>组件默认的尺寸为300X240

mode常用的模式有：

scaleToFill,缩放模式，不保持横纵比

aspectFit,缩放模式，保持横纵比，使图像的长边显示出来

aspectFill，缩放模式，保持横纵比，使图像的短边显示出来

## • <swiper>组件

<swiper>组件用于实现滑块视图容器，其语法结构是：

<swiper
  
 autoplay="是否自动播放"
  
 indicator-dots="是否显示面板指示点"
  
 indicator-color="指示点正常显示的颜色"
  
 indicator-active-color="指示点被选定的颜色"
  
 circular="是否采用衔接滑动"
  
 interval="自动切换的时间间隔(5000ms)"
  
 duration="切换的时长(500ms)"
  
 easing-function="动画类型">
  
 <swiper-item>...</swiper-item>
  
 ...
  
</swiper>

autoplay属性必须存在否则不会触发轮播图

动画类型包括：

linear:线性动画

easeInCubic,缓入动画

easeOutCubic,缓出动画

easeInOutCubic，缓入和缓出动画

## • <text>组件

<text>组件用于实现文本，语法结构是：

<text selectable="是否可选" decode="是否解码"></text>

示例代码如下：

<text selectable="true">我是可选定的文本(需在真机模式下调试)</text>
  
<text>A&gt;B,B&lt;A</text>
  
<text decode="true">A&gt;B,B&lt;A,Tom&amp;John</text>

微信小程序的解码只支持：&nbsp;、&gt;、&lt;、&amp;及&apos;

以上五个称为HTML实体(HTMLEntity)

## • <icon>组件

<icon>组件用于实现图标，语法结构是：

<icon type="图标类型" size="图标尺寸" color="颜色"></icon>

图标类型包括：

success，表示成功

warn，表示警告

waitng表示等待

clear表示清除

info表示信息

cancel表示取消

download表示下载

search表示搜索

## • <progress>组件

<progress>组件表示进度条，语法结构是：

<progress percent="百分比" show-info="是否显示百分比"
  
 font-size="字号大小"
  
 stroke-width="进条度线的宽度"
  
 activeColor="已完成的进度条颜色"
  
 backgroundColor="未完成的进度条颜色">
  
</progress>

## • <button>组件

<button>组件用于实现按钮，其语法结构是：

<button type="按钮类型(primary|default|warn)"
  
 size="按钮尺寸(default|mini)"
  
 plain="是否镂空"
  
 loading="是否带有加载图标"
  
 disabled="是否禁用"
  
 open-type="开启的微信开放能力">
  
 ...
  
</button>

微信开放能力有：

contact,打开客服会话

share，触发用户转发

getPhoneNumber,获取用户手机号

getUserInfo,获取用户信息

## • <navigator>组件

<navigator>组件为页面链接组件，语法结构是：

<navigator url="当前小程序内的跳转链接" open-type="跳转方式">
  
 ...
  
</navigator>

跳转方式有：switchTab -- 可以跳转到包含在tabBar属性内的页面

默认情况下URL属性只能跳转到没有包含在app.json的tabBar属性内的页面

## • <input>组件

<input>组件用于实现输入框，**该组件为原生组件**，语法结构是：

<input
  
 type="输入框的类型"
  
 password="是否为密码"
  
 placeholder="占位符"
  
 maxlength="最大长度"
  
 focus="是否自动获得焦点"
  
 bindinput="输入时触发的函数"
  
 bindfocus="获取焦点时触发的函数"
  
 bindblur="失去焦点时触发的函数"
  
 bindconfirm="点击完成按钮时触发的函数">
  
</input>

输入框类型包括：

* text，文本输入键盘
* number，数字输入键盘
* idcard，身份证号输入键盘
* digit，带小数点的数字输入键盘

# 3.WXML

## 3.1 数据绑定

page.wxml中的动态数据来源于page.js中的data

### • 内容绑定

<component>{{variable}}</component>

示例代码如下：

### • 属性绑定

<component 属性名称="{{variable}}"></component>

示例代码如下：

<!--page.wxml-->
  
   
<icon type="{{type}}" size="{{size}}"></icon>

//page.js
  
   
Page({
  
 data:{
  
 type:'success',
  
 size:50
  
 }
  
})

### • 样式绑定

<component class="{{variable}}">...</component>
  
   
<component style="{{variable}}">...</component>

<!--page.wxml-->
  
   
<text class="{{class}}">CSS样式类的绑定</text>

//page.js
  
   
Page({
  
 data:{
  
 class:'text'
  
 }
  
})

<!--page.wxml-->
  
   
<view style="{{inline}}">行内样式的绑定</view>

//page.js
  
   
Page({
  
 data:{
  
 inline:'color:red;font-size:24px'
  
 }
  
})

<!--page.wxml-->
  
   
<view style="color:{{color}};font-size:{{size}}">行内样式的绑定</view>

//page.js
  
   
Page({
  
 data:{
  
 color:'#00f',
  
 size:'26px'
  
 }
  
})

## 3.2 列表渲染

<component
  
 wx:for="{{variable}}" wx:key="index"
  
 wx:for-index="索引变量名称" wx:for-item="成员变量名称">
  
</component>

默认的成员变量名称为item，默认的成员下标变量为index

示例代码如下：

//page.js
  
   
Page({
  
 data:{
  
 array:['关羽','赵云','刘备','曹操']
  
 }
  
})

<!--page.wxml-->
  
   
<view wx:for="{{array}}" wx:key="index">
  
 索引值:{{index}}--姓名:{{item}}
  
</view>
  
<view wx:for="{{array}}"
  
 wx:for-item="user"
  
 wx:for-index="key"
  
 wx:key="key">
  
 姓名:{{user}}, 索引值:{{key}}
  
</view>

## 3.3 条件渲染

<component wx:if="{{condition}}">
  
...
  
</component>
  
   
<component wx:if="{{condition}}">
  
...
  
</component>
  
<component wx:else>
  
...
  
</component>
  
   
<component wx:if="{{condition}}">
  
...
  
</component>
  
<component wx:elif="{{condition}}">
  
...
  
</component>
  
...
  
<component wx:else>
  
...
  
</component>

示例代码如下：

<!--page.wxml-->
  
<view wx:if="{{week == 0 || week == 6}}">休息</view>
  
<view wx:else>去上班</view>
  
   
<view wx:if="{{week == 0}}">星期日</view>
  
<view wx:elif="{{week == 1}}">星期一</view>
  
<view wx:elif="{{week == 2}}">星期二</view>
  
<view wx:elif="{{week == 3}}">星期三</view>
  
<view wx:elif="{{week == 4}}">星期四</view>
  
<view wx:elif="{{week == 5}}">星期五</view>
  
<view wx:else>星期六</view>

//page.js
  
   
Page({
  
 data:{
  
 //表示星期的第几天,0表示星期日,依次类推
  
 week:0
  
 }
  
})

示例代码如下：

<!--page.wxml-->
  
   
<view wx:for="{{users}}" wx:for-item="user" wx:key="index">
  
 <text>
  
 姓名:{{user.username}}
  
 性别:{{user.sex ? '男' : '女'}}
  
 <block wx:if="{{user.education==1}}">专科</block>
  
 <block wx:elif="{{user.education==2}}">本科</block>
  
 <block wx:else>研究生</block>
  
 </text>
  
</view>

//page.js
  
Page({
  
 data:{
  
 users:[
  
 {
  
 username:'关羽',
  
 //true表示男,false表示女
  
 sex:true,
  
 //1表示专科,2表示本科,3表示研究生,4表示硕士,5表示博士
  
 education:1
  
 },
  
 {
  
 username:'莫小贝',
  
 sex:false,
  
 education:2
  
 },
  
 {
  
 username:'吕秀才',
  
 sex:true,
  
 education:3
  
 },
  
 {
  
 username:'刘备',
  
 sex:true,
  
 education:3
  
 }
  
 ]
  
 }
  
})

<block></block>组件用于保证正常的流程控制语句的使用，但不会出现多余的、无用的节点。

作业：

1.尝试用循环结构实现轮播图

2.预习微信小程序中的事件及数据的双向绑定

https://developers.weixin.qq.com/miniprogram/dev/framework/view/wxml/event.html