# WXML语法

微信官方教程如下：https://developers.weixin.qq.com/miniprogram/dev/framework/view/wxml/

## 特点：

1. 布局的方式跟HTML是一模一样的。
2. 标签的名字不再是传统的HTML的了，而是使用微信自己定义的一套，所以写代码的时候完全使用之前写HTML的方式去写，只不过改个标签名就可以了。
3. WXML的语法，和一些模板语法比如Django中的模板语法非常的类似。

## 变量渲染：

1. 使用双大花括号。
2. 获取对象的值，通过下标获取数组中的值。
3. 可以做运算，比如判断，四则运算等。
4. 总而言之一句话：需要使用js中传过来的值，就必须使用双大花括号。

## 条件渲染：

1. 在指定的标签中使用

   ```
   wx:if="条件"
   ```

   ，如果条件成立，则会渲染这个标签。示例代码如下：

   ```
   进入网吧
   ```

2. `else`和`elif`可以直接在`if`下面使用。示例代码如下：

   ```
   出去旅游

   去逛街

   哪里都不去
   ```

   **注意：elif和else只能跟在if后面，否则会报错**

3. 如果需要通过条件来判断是否需要渲染一组标签，那么可以使用`block`。示例代码如下：

```
        出去旅游


        去逛街


        哪里都不去
```

# 列表渲染：

1. 通过语法

   ```
   wx:for="{{列表}}"
   ```

   来渲染一个列表。示例代码如下：

   ```
        {{item}}
   ```

2. 循环中，默认的下标名称是

   ```
   index
   ```

   ，默认的值的名称是

   ```
   item
   ```

   。如果想要修改循环列表的值和下标的名称，那么可以通过

   ```
   wx:for-index
   ```

   和

   ```
   wx:for-item
   ```

   来指定。示例代码如下：

   ```
        {{index}}/{{value}}
   ```

3. 如果想要在循环中渲染多个标签，那么也可以通过

   ```
   block
   ```

   来实现。示例代码如下：

```
    {{index}}
    {{item}}
```

## 九九乘法表案例：

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1×1=1 | | | | | | | | |
| 1×2=2 | 2×2=4 | | | | | | | |
| 1×3=3 | 2×3=6 | 3×3=9 | | | | | | |
| 1×4=4 | 2×4=8 | 3×4=12 | 4×4=16 | | | | | |
| 1×5=5 | 2×5=10 | 3×5=15 | 4×5=20 | 5×5=25 | | | | |
| 1×6=6 | 2×6=12 | 3×6=18 | 4×6=24 | 5×6=30 | 6×6=36 | | | |
| 1×7=7 | 2×7=14 | 3×7=21 | 4×7=28 | 5×7=35 | 6×7=42 | 7×7=49 | | |
| 1×8=8 | 2×8=16 | 3×8=24 | 4×8=32 | 5×8=40 | 6×8=48 | 7×8=56 | 8×8=64 | |
| 1×9=9 | 2×9=18 | 3×9=27 | 4×9=36 | 5×9=45 | 6×9=54 | 7×9=63 | 8×9=72 | 9×9=81 |

wxml文件代码：

```
<view wx:for="{{[1, 2, 3, 4, 5, 6, 7, 8, 9]}}" wx:for-index="index1" wx:for-
item="row" class='row'>
    <view wx:for="{{[1, 2, 3, 4, 5, 6, 7, 8, 9]}}" wx:for-index="index2" wx:for-
item="col" wx:if="{{col <= row}}" class='col'>
      {{col}}*{{row}}={{row*col}}
    </view>
</view>
```

wxssw文件代码：

```
.row{
  display: flex;
  justify-content: start;
  font-size: 10px;
}

.row .col{
  width: 40px;
}
```

## wx:key作用：

如果列表中项目的位置会动态改变或者有新的项目添加到列表中，并且希望列表中的项目保持自己的特征和状态（如 `中的输入内容，` 的选中状态），需要使用 wx:key 来指定列表中项目的唯一的标识符。

wx:key 的值以两种形式提供：

1. 字符串或者数字，代表在 for 循环的 array 中 item 的某个 property，该 property 的值需要是列表中唯一的字符串或数字，且不能动态改变。**在写的时候，直接写这个 property 的名字就可以了，不需要写`item.property`的形式，并且不需要加中括号。**
2. 保留关键字 *this 代表在 for 循环中的 item 本身，这种表示需要 item 本身是一个唯一的字符串或者数字，如：*
   *当数据改变触发渲染层重新渲染的时候，会校正带有 key 的组件，*框架会确保他们被重新排序，而不是重新创建*，以确保使组件保持自身的状态，并且提高列表渲染时的效率。
3. 及时列表中的组件没有发生状态改变，那么也建议使用`wx:key`。因为如果不使用，那么以后重新渲染的时候，就会把之前组件销毁掉，然后重新创建，性能会很低。

如不提供 wx:key，会报一个 warning， 如果明确知道该列表是静态，或者不必关注其顺序，可以选择忽略。

# 模板：

有些时候，一段布局代码我们需要在多个地方使用，那么我们可以将其定义成模板，然后把变量单独抽取出来，通过调用模板的时候再传递过去。示例代码如下：

```
<template name="msgItem">
  <view>
    <text> {{index}}: {{msg}} </text>
    <text> Time: {{time}} </text>
  </view>
</template>
```

调用模板：

```
<import src="../../templates/msgItem.wxml" />
<template is="msgItem" data="{{index:1,msg:"你好！我是睿道课堂！",time:3000年10月22日}}"/>
```

在传递变量的时候第三节：WXML语法，如果是直接从js文件中获取到的，那么可以通过`...item`的方式来展开显示。示例代码如下：

```
Page({
    data: {
        message: {
            index: 1,
            msg: '你好！我是睿道课堂！',
            time: '2018年10月22日'
        }
    }
});
<import src="../../templates/msgItem.wxml" />
<template is="msgItem" data="{{...message}}"/>
```

另外，如果想使用模板的样式文件，也需要在`wxss`文件中导入模板的`wxss`文件。示例代码如下：

```
@import "../../../templates/msgItem.wxss";
```

# include：

如果一段代码是不需要填入动态的数据，那么可以直接使用`include`把这段代码引入到其他的地方。示例代码如下：
header.html中的代码：

```
<view>
这是header.wxml中的内容
</view>
```

index.wxml中的代码：

```
<include src="header.wxml" />
<view>
这里是index.wxml中的代码
</view>
```