一、

循环数组：A:for=“{{数组}}” key=”{{item}}” 视图渲染{{item}}

# 二 改变视图数据

This.setData({}) this.$spliceData({ 'a.b': [1, 0, 5, 6] })

改变数组内数据：

this.setData({

"array[0].qw":"阿拉布坦",

})

数组名要加引号

三。、页面跳转

关闭当前页面

dd.redirectTo({

url: '/pages/index/index'

})

保留当前页面

dd.navigateTo({

url: 'new\_page?count=100'

})

关闭当前所有页面，跳转到应用内的某个指定页面。dd.reLaunch({

url: '/page/index'

})

三、getCurrentPages()函数用于获取当前页面栈的实例，

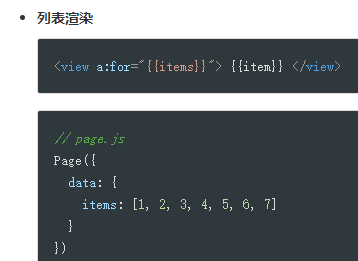
四、条件渲染

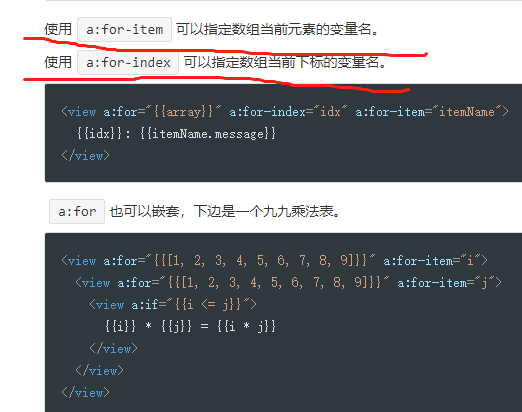
a:if="{{flage==1}}"

<view a:elif="{{flage==2}}">

a:else

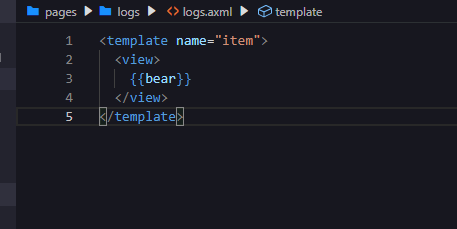
五数组循环



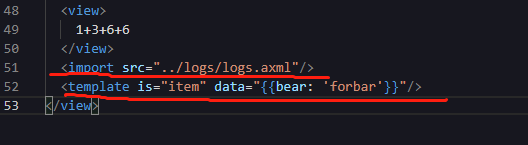


六模板

模板写进一个单独页面



引用时候用important





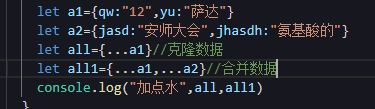
七：点击事件



八：运算符 hidden属性

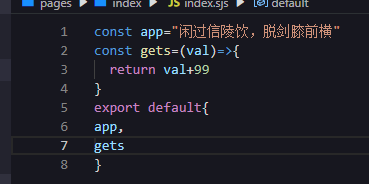
<view hidden="{{flag ? true : false}}"> Hidden </view>

九：展开运算符

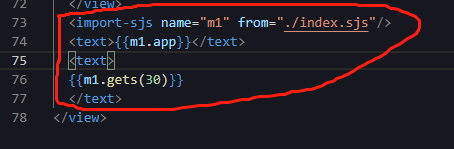


十、脚本语言

文件夹命名app.sjs



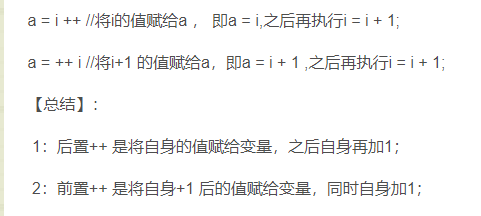
引用



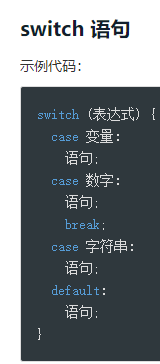
<import-sjs from="./index.sjs" name="test"></import-sjs>

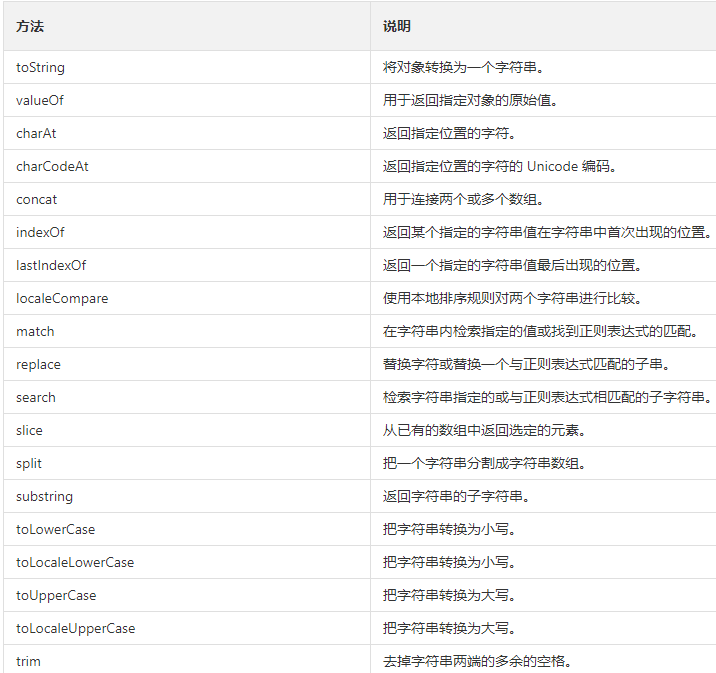
Form 要引入的页面 name：text.变量或者方法

十一、

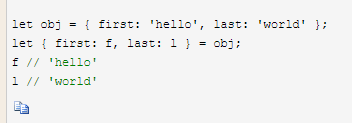


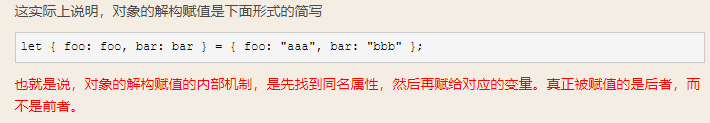
Switch：





十二、对象解构赋值



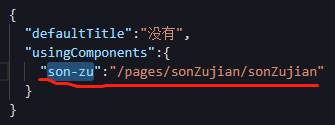


getDate(); 时间函数

十三、

导入外联样式

十四、组件传值

**引入组件**：在父组件的json中配置引入路径

Son-zu写在父组件html中<son-zu></ son-zu>

**父传子： 在父组件的组件标签里写上**<son-zu POI=“{{ao}}”></ son-zu> ao为父组件data中的值，poi为传到子组件中的值，把POI写到子组件props中：props：{

POI：“”

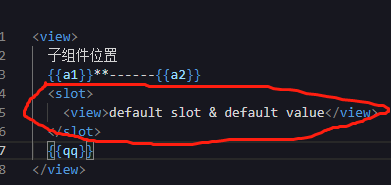
}

在子组件标签中可以直接写{{poi}}渲染出，在函数中调用时候用this.props.poi

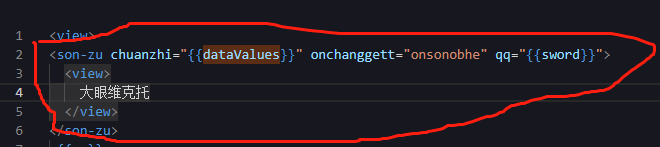
子传父：

**十五、slot 槽位**

**当在子组件中写上<solt>安达市多</solt>时候，**



**solt中的内容会显示到父组件里面，**



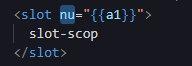
**但是如果父组件中的子组件标签有内荣，solt的内容就会被覆盖**

**十六、named slot**

**子组件当中 <slot name=”header”></slot>**

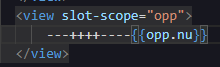
**父组件中 <view =”header”>header</view>**

slot scope

子组件

a1是子组件中data的值，

父组件



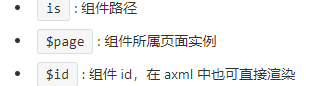
Slot-scope=“opp”是父组件去拿子组件的值，opp.Nu渲染出值

十七 组件的周期

**didMount为渲染后回调，此时页面已经渲染，通常在这里请求服务端数据比较合适。**

**didUpdate 为更新后回调，每次组件数据变更的时候都会调用。**

**didUnmount 为删除后回调，每当组件示例从页面删除的时候会触发此回调。**





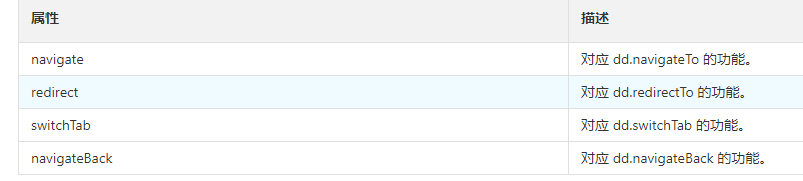
**十八、**

dd.navigateTo({

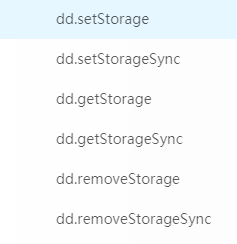
url:'/pages/son/son'

})

**跳转的页面要在app.json；里面配置**



十九、数据存储

**Sync**同步

dd.setStorage({key:"firts",data:{addreee:"北京",spell:"asda555"},success: function() {

dd.alert({content: '写入成功'});

}})

**二十、删除对象 delete op.name**