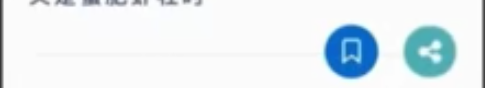
# Template

Template只是一个类似括号的玩意，不显示内容，所以template里面不能加点击事件等。

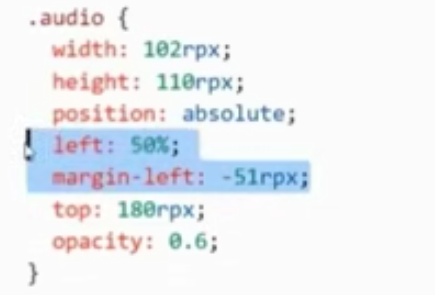


中间线的实现方式



居中的方式

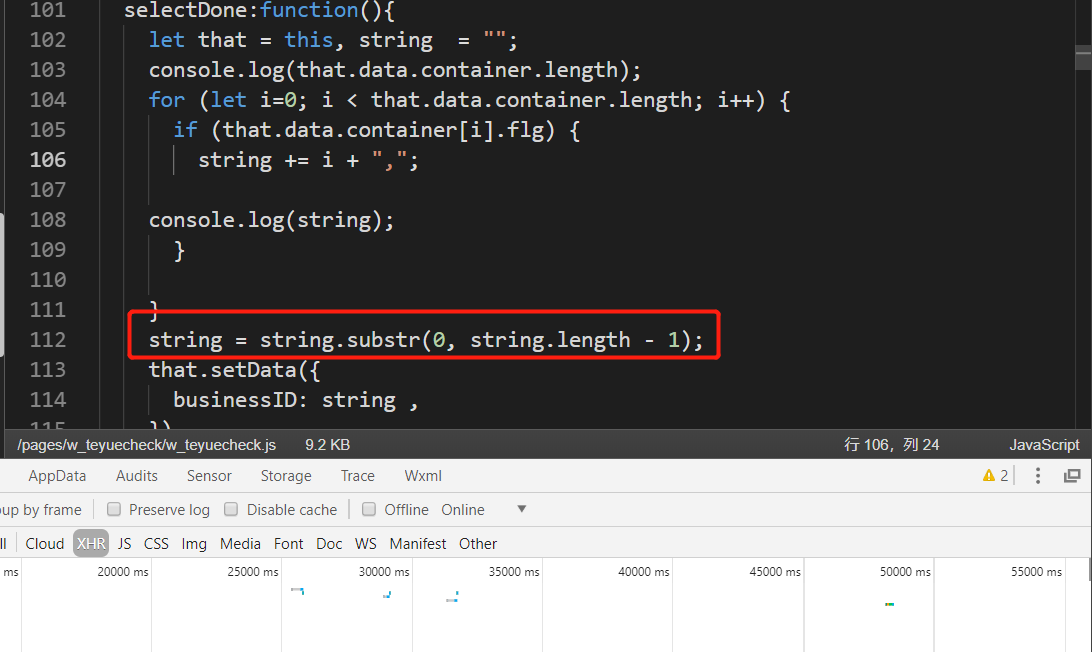
Position：absolute；是让该组件脱离文本



# 把很多元素加入到一个字符串显示的方法

## 删除字符串的最后一个字符

string=string.substr(0,string.length-1)是删除最后一个字符



把很多元素增加到数组里面

# push() 方法

可向数组的末尾添加一个或多个元素，并返回新的长度。

let that = this, arr = [];

console.log(that.data.container.length);

for (let i=0; i < that.data.container.length; i++) {

if (that.data.container[i].flg) {

arr.push(i);

//和字符串不同，这里不能写成arr=arr.push(i);因为push是一个方法，方法里面不能叠加方法

console.log(arr);

}

}

# pop()方法

pop()方法刚好和push()方法相反。pop()方法删除数组的最后一个元素，把数组的长度减1，并且返回它被删除元素的值，如果数组变为空，则该方法不改变数组，返回undefine值。

数组名.pop();

例如：var arr=[a,b,c]

arr.pop();

console.log(arr);

输出结果是[a,b]

# unshift()方法

unshift()方法是向数组的开头添加一个或多个元素，并且返回新的长度。

var arr = [b,c];

arr.unshift(z,x,a);

console.log(arr);

输出结果是[z,x,a b,c]

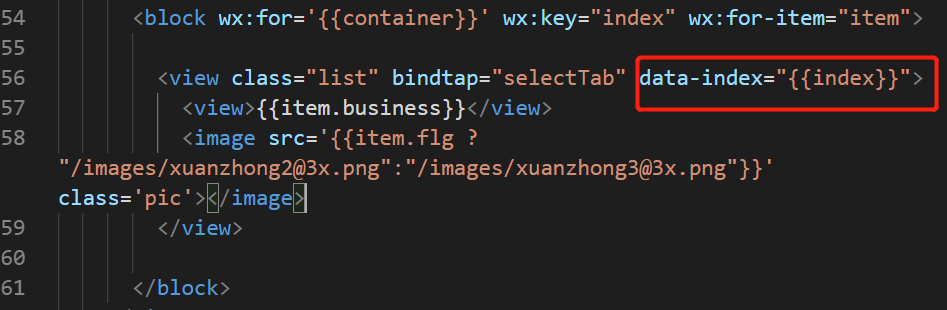
# shift()方法

shift()方法和unshift()方法恰恰相反。该方法用于把数组的第一个元素从其中删除，并返回被删除的值。如果数组是空的，shift()方法将不进行任何操作，返回undefined的值。

# concat() 方法用于连接两个或多个数组。

# 对点击事件进行标记（划重点）

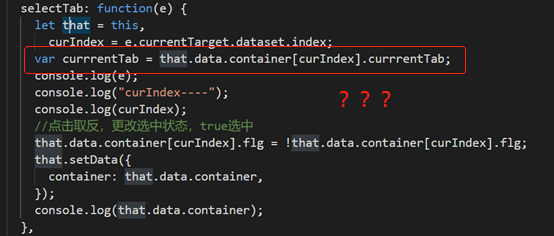
## 对于wx:for的循环数组，



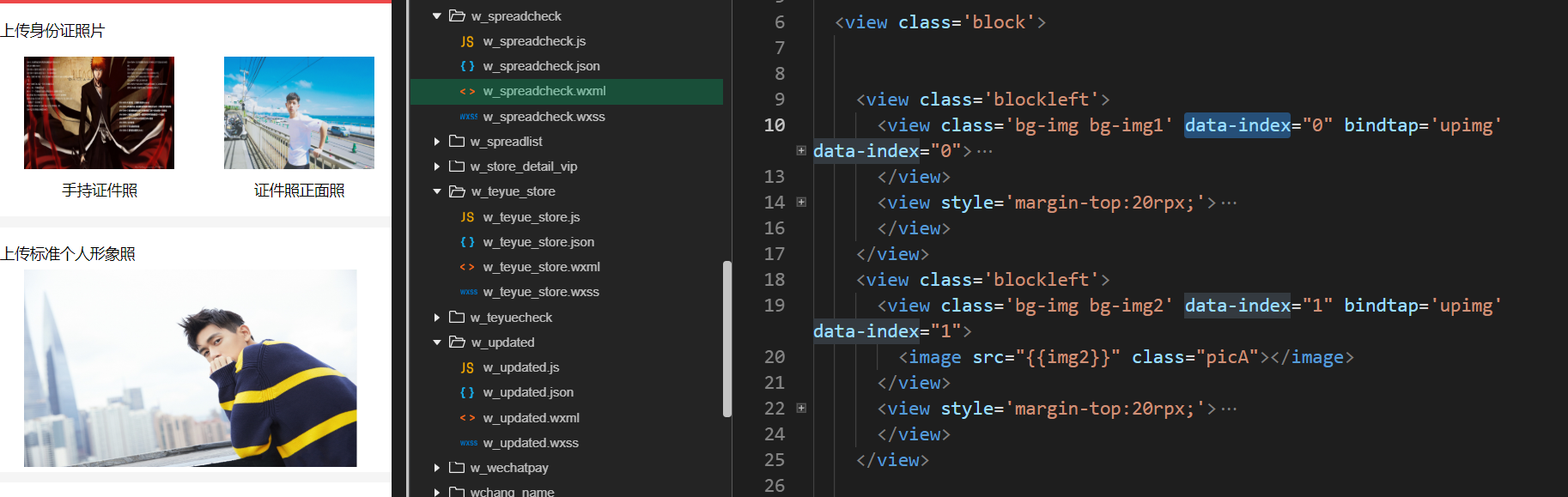
data- 是一个自定义的属性。

例如：data-index会在点击后返回的参数里面,路径如下e.currentTarget.dataset.index;数组的话会自动排序。

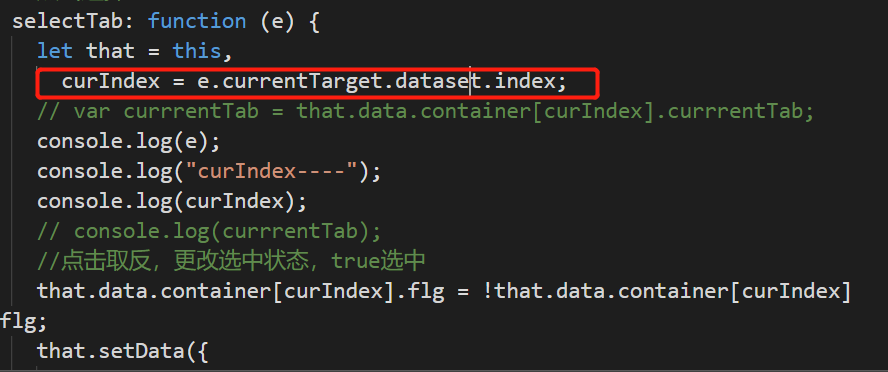
js代码如下

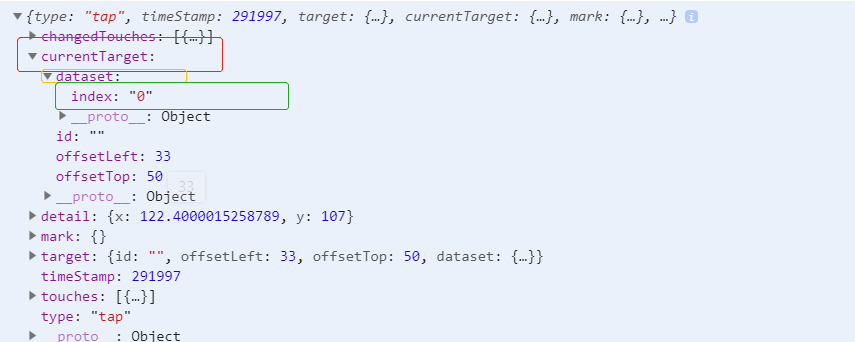


对于一个页面的多个点击事件，可以手动标记，从而可以调用同一个方法，但是不会混乱



点击事件返回值：





自定义的属性在返回值的路径是res.currentTarget.dataset.index

# undefined出现原因：(口诀：一变量二函数一对象)

           1.变量被声明了但是没赋值时

           2.调用函数时，应该提供的参数没提供，则该参数为undefined

           3.函数没有返回值时，默认返回undefined

           4.对象没有赋值的属性

# null出现原因

           1.作为函数的参数，表示该函数的参数不是对象

           2.作为对象原型链的终点

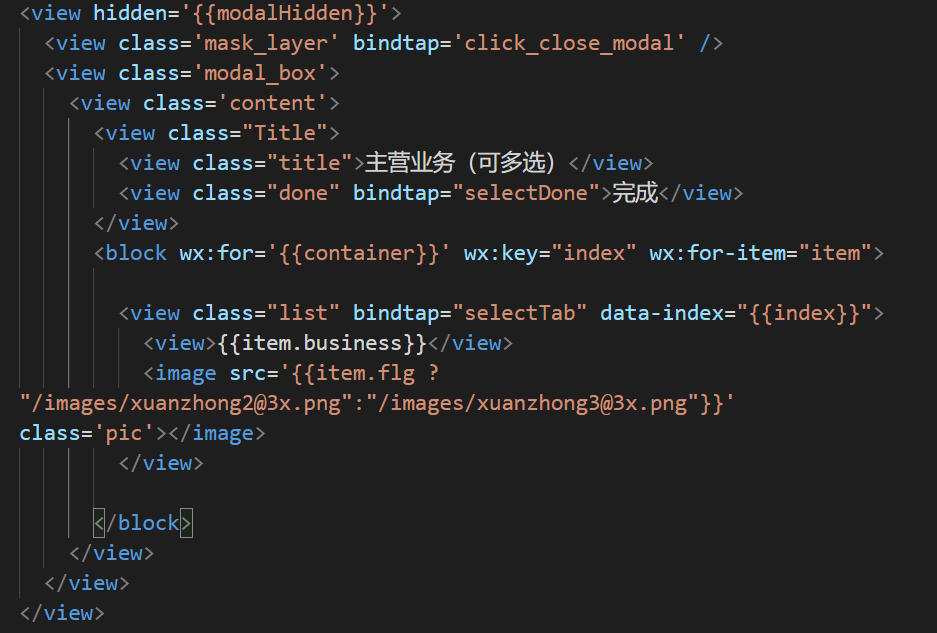
# 多列选择器-picker

## 从后台获取省列表，根据省id获取市列表，根据市id获取区列表

# 弹窗

## 可自定义

思路：可把一个view组件设置hidden属性，当点击需要 显示这个组件时，把hidden属性设置为false，view组件内容既可以展示。例如;



# 分页

# js禁用按钮的的属性

disabled=”disabled”

disabled 属性规定禁用按钮。

被禁用的按钮既不可用，也不可点击。

可以设置 disabled 属性，直到满足某些条件（比如选择一个复选框），才恢复用户对该按钮的使用。然后，可以使用 JavaScript 来清除 disabled 属性，以使文本区变为可用状态。

清除 disabled 属性