1. 要考虑多种情况 wx:if=”{{data}}” 和wx:else
2. 在做CSS时,考虑是否要定高，数据的可变性会不会对视图发生影响，如会影响，应该使用绝对定位属性。
3. 将数据添加入数组 var a=[]; a.push(1);
4. 获取距离今天的时间  
   for (let i = 0; i < res.data.result.list.length;i++){

let today = new Date ()

let timerToday = today.getTime()

let cha = (timerToday - res.data.result.list[i].stime) / 1000 / 3600 / 24

res.data.result.list[i].stime = Math.floor(cha)

}

返回到原始数据中再进行获取，如果自己设置一个数组，但是其遍历会是数据全部展示出来

5. **“+=”是什么意思？**

int i = 1;

int sum = 0;

sum = sum + i; //这个意思

sum += i; //这是简写

6.**canvas**

fillRect(x, y, width, height)

绘制一个填充的矩形

strokeRect(x, y, width, height)

绘制一个矩形的边框

clearRect(x, y, width, height)

清除指定矩形区域，让清除部分完全透明。

beginPath()

新建一条路径，生成之后，图形绘制命令被指向到路径上生成路径。

closePath()

闭合路径之后图形绘制命令又重新指向到上下文中。

stroke()

通过线条来绘制图形轮廓。

fill()

通过填充路径的内容区域生成实心的图形。

function draw() {

var canvas = document.getElementById('canvas');

if (canvas.getContext){

var ctx = canvas.getContext('2d');

ctx.beginPath();

ctx.moveTo(75,50);------开始位置，移动到

ctx.lineTo(100,75);--------直线画到该位置

ctx.lineTo(100,25); --------直线画到该位置

（绘制直线，需要用到的方法lineTo()。）

ctx.fill();------填充该区域 } }

ctx.arc(75,75,50,0,Math.PI\*2,true); // 绘制圆

ctx.arc(75,75,35,0,Math.PI,false); // 口(顺时针)半圆

ctx.fill();与ctx. stroke()是对应的，一种是填充，一种是描线

小程序中canvas

// canvas.js

Page({

canvasIdErrorCallback: function (e) {

console.error(e.detail.errMsg)

},

onReady: function (e) {

// 使用 wx.createContext 获取绘图上下文 context

var context = wx.createCanvasContext('firstCanvas')

context.setStrokeStyle("#00ff00")

context.setLineWidth(5)

context.rect(0, 0, 200, 200)

context.stroke()

context.setStrokeStyle("#ff0000")

context.setLineWidth(2)

context.moveTo(160, 100)

context.arc(100, 100, 60, 0, 2 \* Math.PI, true)

context.moveTo(140, 100)

context.arc(100, 100, 40, 0, Math.PI, false)

context.moveTo(85, 80)

context.arc(80, 80, 5, 0, 2 \* Math.PI, true)

context.moveTo(125, 80)

context.arc(120, 80, 5, 0, 2 \* Math.PI, true)

context.stroke()

context.draw()

}

})

7.一个事件里有多个处理函数，其关系是平行的，并不是按照顺序从上之下，因此如果要按顺序，则需要包含在前的函数的if条件内，判断函数是否走完，再调用另外一个函数。

8.判断返回要判断其result\_code，存在且为某一值时才能确定内容已经提交到服务器上去了。

9.小程序中radio 加入name要加载radio-group中，form提交才会有数据

10. 后端要求的数据少一个都会没有request返回，所以按照后端要求提交的数据传递

11. splice() 方法向/从数组中添加/删除项目，然后返回被删除的项目。

**注释：**该方法会改变原始数组。

**语法**

arrayObject.splice(index,howmany,item1,.....,itemX)

|  |  |
| --- | --- |
| **参数** | **描述** |
| index | 必需。整数，规定添加/删除项目的位置，使用负数可从数组结尾处规定位置。 |
| howmany | 必需。要删除的项目数量。如果设置为 0，则不会删除项目。 |
| item1, ..., itemX | 可选。向数组添加的新项目。 |

**返回值**

|  |  |
| --- | --- |
| **类型** | **描述** |
| Array | 包含被删除项目的新数组，如果有的话。 |

12. reverse()---字符串反转

13.父元素使用了opacity之后，子元素也会透明化，即使你设置了子元素opacity：1. 正确的做法是在父元素的background:rgba(0,0,0,0.8)

14.小程序的text是没办法绑定事件的

15.change和input的区别：change是指输入完成，input每输入一个都会触发事件

16.typeof(“supper”)判断变量，字符串的类型，输出number,string等

17. pow() 方法可返回 x 的 y 次幂的值。

### 语法

Math.pow(x,y)

18. join() 方法用于把数组中的所有元素放入一个字符串。

元素是通过指定的分隔符进行分隔的。

<script type="text/javascript">

var arr = new Array(3)

arr[0] = "George"

arr[1] = "John"

arr[2] = "Thomas"

document.write(arr.join("."))

</script>

输出：

George.John.Thomas

19. **func.call()与func.apply()与func.bind()**

var func=new function(){this.a="func"}

var myfunc=function(x){

var a="myfunc";alert(this.a); alert(x);}

myfunc.call(func,"var");

myfunc.call(func,a,b)

call用来修改this所对应的域，实际执行还是原来的myfunc方法，只不过this指向改变了，ab为myfunc()的参数

myfunc.apply(func,[a,b])

apply用法与call方法一样，只不过其传入的参数为一个数组

function Animal(name) {

this.name = name;

this.showName = function () {

alert(this.name);

}

}

function Cat(name) {

Animal.call(this, name);

}

var cat = new Cat('Black Cat');

cat.showName();

　　输出结果为：Black Cat

　　Animal.call(this) 的意思就是使用 Animal对象代替this对象，那么 Cat中不就有Animal的所有属性和方法了吗，Cat对象就能够直接调用Animal的方法以及属性了。——实现继承

20. super2 = super2.concat(super1)

let sp = [...that.data.list, ...res.data.result.donationList]

第2句为es6语法，数组的合成

21.引用github中的图片

<https://raw.githubusercontent.com/zxpsuper/picture/master/bird.png>

22.vue永远不要把模块一起复制，重新安装可以避免一些奇怪的报错

23.习惯先拉后推可以避免推送失败。----git小乌龟