1.函数内部变量调用外部函数，

  
  
错误：var panduan = pandua(jieguo)  
  
2.JS new出来的构造函数属于一个对象没有返回值。

如果var c = a1(4);改为var c = new a1(4);则弹出object  
 <body>

<button onclick="a2()">666</button>

<script type="text/javascript">

var b=0;

function a1(b){

b=b+1;

return b;

};

function a2(){

var c = a1(4);

alert(c);

}

</script>

</body>

3.async异步函数

var asyncReadFile = async function (){

var f1 = await readFile('/etc/fstab');

var f2 = await readFile('/etc/shells');

console.log(f1.toString());

console.log(f2.toString());};

async 表示函数里有异步操作，await 表示紧跟在后面的表达式需要等待结果

<http://www.ruanyifeng.com/blog/2015/05/async.html>

4.ES6  
 (1)一个类必须有constructor方法，如果没有显式定义，一个空的constructor方法会被默认添加.  
 constructor方法默认返回实例对象（即this），完全可以指定返回另外一个对象。  
  
 (2) function add(...values) {  
   
 ES6引入rest参数（形式为“...变量名”），用于获取函数的多余参数，这样就不需要使用arguments对象了。rest参数搭配的变量是一个数组，该变量将多余的参数放入数组中。  
 <http://www.cnblogs.com/huansky/p/5684020.html>  
  
rest参数之后不能再有其他参数（即只能是最后一个参数），否则会报错。

// 报错function f(a, ...b, c) {

// ...

}

[http://es6.ruanyifeng.com/#docs/function#rest](http://es6.ruanyifeng.com/#docs/function)参数  
  
 (3)  
 类名(组件名)一定要用大写开头，否则自定义的组件无法编译，识别不出来。

类中定义render函数要注意两点，见代码注释。

render() {//开头花括号一定要和小括号隔一个空格，否则识别不出来

return <ol>//标签前一半一定要和return一行

{

React.Children.map(this.props.children, function (child) {

return <li>{child}</li>;

})

}

</ol>;

}

在class中使用class的变量或者方法，一定要加个this。如this.handlerclick。

es6 绑定事件需要onClick={this.func1.bind(this)}。

这样func1和bind里面的参数‘this’的作用域才绑定到了一起（注意es5是不需要这个bind(this)的），func1中如果有this.name这类语句，相当于是使用参数‘this’里面的变量值;或者使用箭头函数func1= (e)=> {函数体}

5.foreach,map,reduce区别

  
实际应用：

把"10,-6.1|28,-6.0" 中的10，28组成数组["10","28"]

-6.1,-6.0组成另一个数组["-6.1","-6.0"]



函数中用filter的时候 需要在数组前面加个return来返回值

6.es6 引用赋值  
 var a = {foo:'bar'};

var b = a;

a.foo = 'good';

console.log(b == a)//true  
  
 对象和数组的赋值是引用赋值， a 和 b 只是一个引用符号，其所指向的对象实体不同（比如 a -> object#001, b -> object#002），js的对象（数组）相等判断是根据是否指向同一个对象实体来的确定的

7.constructor  
 constructor(props){

super(props)

}  
这段话是用来创建this的，this由super（）创建的，子组件加这段代码能得到父组件传过来的props  
  
8.JS函数只执行一次  
 var isFirst = true;

function doS(){

if(isFirst){

doSth...//这里写你要做什么

isFirst = false;

}

}

9.处理字符串 split()  
<http://www.w3school.com.cn/jsref/jsref_split.asp>  
"2:3:4:5".split(":") //将返回["2", "3", "4", "5"]

"|a|b|c".split("|") //将返回["", "a", "b", "c"]

10.查询字符串  
 使用substring（）方法  
 倒叙查询方法：<http://www.cnblogs.com/guilongliang/p/5527790.html>

例子：JSON对象属性名以UOD结尾的且值为1改为2，值为-1改为3 该怎么做  


Map+substring+for in  
  
  


11.负数小数点进两位  
 (-0.014 \* 100).toFixed(1)

直接乘100不行

12.两个页面之间跳转传参数  
 <http://www.cnblogs.com/zhu-xingyu/p/5577118.html>  
  
第一个页面  
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>direct1.html</title>

</head>

<body>

<div><a id="a1" href="direct2.html">点击此处</div>

<script>

var a1=document.getElementById("a1");

var person={

store\_id:"nk11",//模拟,上线改为从参数获得

date:201730,//模拟,上线改为从参数获得

version:20170810

};

a1.href=a1.href+"?"+person.store\_id+'|'+person.date+'|'+person.version;

</script>

</body>

</html>  
  
第二个页面  
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>direct2.html</title>

</head>

<body>

<script>

var getInfo=window.location.search.slice(window.location.search.lastIndexOf("?")+1);

var cc = getInfo.split("|")

console.log(getInfo);

</script>

</body>

</html>

13.合并单元格

http://www.cnblogs.com/xianrongbin/archive/2012/10/05/2712624.html

其中，tabObj代表表的Id，即GridView的Id，cellIndex代表合并第几列，beignRow 代表从几行开始，通常使用1，因为0是表头哈。

js代码如下：

//合并单元格算法tabObj代表表的Id，即GridView的Id，cellIndex代表合并第几列，beignRow 代表从几行开始，通常使用1，因为0是表头

const SpanGrid = function(tabObj, cellindex, beginRow) {

var colIndex = cellindex;

var rowBeginIndex = beginRow;

if (tabObj != null) {

var i, j, m;

var intSpan;

var strTemp;

m = 0;

for (i = rowBeginIndex; i < tabObj.rows.length; i++) {

intSpan = 1;

m++;

strTemp = tabObj.rows[i].cells[colIndex].innerText;

for (j = i + 1; j < tabObj.rows.length; j++) {

if (strTemp == tabObj.rows[j].cells[colIndex].innerText &&(tabObj.rows[i].cells[0].innerText == tabObj.rows[j].cells[0].innerText ) ) {

intSpan++;

tabObj.rows[i].cells[colIndex].rowSpan = intSpan;

tabObj.rows[j].cells[colIndex].style.display = "none";

}

else {

break;

}

}

}

i = j - 1;

}

14.JS 获得第几周的星期日期

<http://bbs.csdn.net/topics/392030605>

ISO标准算法  
 function getDate(period,week){

        //period:第几周，week:星期几

        if(/^([1-4]?\d|5[12])$/.test(period) && /^[0-6]$/.test(week)){

            var ymd = new Date(new Date().getFullYear(),0,1);

            ymd.setDate(1 + period \* 7 - ymd.getDay() + week);

            return ymd.getFullYear() +"-"+ (ymd.getMonth() + 1) +"-"+ ymd.getDate();

        }else{

            return "参数错误";

        }

    }

console.log(getDate(0,0));//获取第一周星期一日期

console.log(getDate(3,3));//获取第四周星期四日期

console.log(getDate(10,6));//获取第十一周星期天日期

补充:判断当前日期为第几周 <http://momentjs.cn/docs/#/get-set/iso-week-year/>

15.获取URL参数  
GetRequest() {

var url = location.search;

var theRequest = new Object();

if (url.indexOf("?") != -1) {

var str = url.substr(1);

var strs = str.split("&");

for(var i = 0; i < strs.length; i ++) {

theRequest[strs[i].split("=")[0]]=unescape(strs[i].split("=")[1]);

}

}

return theRequest;

}

16.数组倒叙输出  
 需求:api倒叙排列 data:reverse()

17.JS 对象里掉用函数结果当属性值  
 

18.用for循环生成对象时,外面要包层if 避免每次引用这个对象都计算一遍  


19.JS深复制  
 如果数组里嵌套对象,对象里又嵌套数组,很可能操作的时候会出问题,解决办法是data =JSON.parse(JSON.stringify(data))

这个方法当有属性是undefined或null的时候会被移除掉  
<https://www.zhihu.com/question/23031215>

20.alert函数去掉服务器地址显示

http://blog.csdn.net/u010883006/article/details/77985034



21.断点  
 在代码中空一行写bugger即可

22.error  
 <http://www.jb51.net/article/14481.htm>

Property:   
  name: 错误名   
  number: 错误号   
  message: 错误信息,多同description   
  description: 描述   
 FF Only 属性   
  fileName: 错误发生的文件   
  stack: 错误发生时的调用堆栈

23.MD5  
 参数为0的时候有时候需要传字符串0(根据后端)

24.判断类型  
 Object.prototype.toString.call (data) == "undefined"

不能用typeof 因为type数组和null的结果都是object

25.两个div同步滚动

//编辑器和预览区同步滚动

let preview = document.querySelector('#preview');

let editor = document.querySelector('#editor');

preview.addEventListener('scroll', (e)=> {

e.stopPropagation();

editor.scrollTop = preview.scrollTop;

});

editor.addEventListener('scroll', (e)=> {

e.stopPropagation();

preview.scrollTop = editor.scrollTop;

});

26.遍历tree 找出叶子节点的层级

function getArray(data,name)

{

for (var i in data) {

console.log('i',i);

console.log('datai',data[i].children);

if (data[i].name == name) {

console.log(data[i]);

break;

} else {

getArray(data[i].children, name);

}

}

}

<http://blog.csdn.net/yf275908654/article/details/52215321>

27.