111assign 合并对象并具有相同属性的对象

var o1 = { a: 1, b: 1, c: 1 };

var o2 = { b: 2, c: 2 };

var o3 = { c: 3 };

var obj = Object.assign({}, o1, o2, o3);

console.log(obj); // { a: 1, b: 2, c: 3 }

注释

*/\*\*  
 \** ***@method*** *超出行显示省略号  
 \** ***@param*** *{string} contextClass 超出行显示省略号  
 \** ***@param*** *{number} [lineNum=2] 第几行需要省略号  
 \** ***@param*** *{number} [lineHeight=28] 行高  
 \** ***@param*** *{number} [safeNum=30] 用于提高dom操作性能，每行安全的字符数，如30，2行，不会检查30\*2以内的字符是否需要填充‘...’  
 \** ***@return*** *null 返回值说明  
 \*/*

多行省略号

*/\*\*  
 \** ***@method*** *超出行显示省略号,在update中调（会将样式自动设置成 height:auto）;  
 \** ***@param*** *{string} contextClass 超出行显示省略号  
 \** ***@param*** *{number} [lineNum=2] 第几行需要省略号  
 \** ***@param*** *{number} [lineHeight=28] 行高  
 \** ***@param*** *{number} [safeNum=30] 用于提高dom操作性能，每行安全的字符数，如30，2行，不会检查30\*2以内的字符是否需要填充‘...’  
 \** ***@return*** *null 返回值说明  
 \*/*dot: **function** (contextClass, lineNum, lineHeight, safeNum) {  
 lineNum = lineNum || 2  
 lineHeight = lineHeight || 28  
 safeNum = lineHeight || 30  
 **let** el = document.getElementsByClassName(contextClass)  
 // console.log(el);  
 **for** (**var** j = 0; j < el.length; j++) {  
 **let** wN = 0 // 字数，图片算一个字  
 el[j].style.height = 'auto'  
 el[j].style.width = 'auto'  
 el[j].style.maxHeight = 'none'  
 // console.log('j', j, 'el[j].offsetHeight', el[j].offsetHeight)  
 **var** pT = parseInt(window.getComputedStyle(el[j]).paddingTop)  
 **var** pB = parseInt(window.getComputedStyle(el[j]).paddingBottom)  
 // console.log('pT', pT, 'pB', pB, lineNum \* lineHeight + pT + pB)  
 **if** (el[j].offsetHeight < lineNum \* lineHeight + pT + pB) {  
 // console.log('< 25', el[j].offsetHeight)  
 **continue** } **else** {  
 // console.log('> 25')  
 // break  
 }  
 **let** str = el[j].innerHTML  
 // console.log(str);  
 **for** (**var** i = 0; i < str.length; i++) {  
 **var** a // 是否是img的标志位  
 **var** resNum  
 **var** res  
 **if** (str[i] === '<' && str.slice(i + 1, i + 5) === 'img ') {  
 // i = i+1  
 res = str.slice(i)  
 // console.log('res', res)  
 resNum = res.indexOf('>')  
 // console.log(resNum.);  
 a = 1  
 i = i + resNum  
 wN++  
 } **else** {  
 wN++  
 }  
 // console.log('str.substr(0, i)', str.substr(0, i + 1))  
 **if** (wN > 2 \* safeNum) {  
 el[j].innerHTML = str.substr(0, i + 1) + '...'  
 // console.log('i=', i, 'el[j].offsetHeight', el[j].offsetHeight) // , 'str', str  
 **if** (el[j].offsetHeight > lineNum \* lineHeight + pT + pB) {  
 // el[j].style.overflow = 'hidden';  
 **if** (a === 1) {  
 a = 0  
 **if** (j === 0) {  
 // console.log('strLast', str.substr(0, i - resNum))  
 }  
 el[j].innerHTML = str.substr(0, i - resNum) + '...'  
 // console.log(t.substr(0, i - resNum) + '....');  
 } **else** {  
 a = 0  
 // console.log('strLast', str.substr(0, i - 1))  
 // el[j].innerHTML = t.substr(0, i);  
 el[j].innerHTML = str.substr(0, i - 1) + '...'  
 }  
 **break** } **else** {  
 a = 0  
 }  
 el[j].innerHTML = str  
 }  
 }  
 }  
}

1111样式

111max

## 可能的值

|  |  |
| --- | --- |
| **值** | **描述** |
| none | 默认。定义对元素被允许的最大高度没有限制。 |
| *length* | 定义元素的最大高度值。 |
| *%* | 定义基于包含它的块级对象的百分比最大高度。 |
| inherit | 规定应该从父元素继承 max-height 属性的值。 |

对象的浅拷贝

function easyClone(Obj) {

var objNew = {};

for ( var i in Obj) {

objNew[i] = Obj[i];

}

return objNew;

其实就是将每个原对象的属性和值复制到新对象上去，当然我们也可以使用Object.assign() 方法可以把任意多个的源对象自身的可枚举属性拷贝给目标对象，然后返回目标对象，同时Object.assign() 也是浅拷贝。

浅拷贝因为没有递归循环检查对象的每个值是否是对象，而是直接进行了赋值，所以如果某个值是对象的时候就会出现问题，所以在一般情况下我们需要用深拷贝来进行备份。

对象的深拷贝

最简单的深拷贝

var b = JSON.parse(JSON.stringify(a))

但是这种深拷贝有一定的局限性，第一：无法复制函数，第二：原形链没了，对象就是object，所属的类没了。

其实简单的深拷贝只需要我们递归调用浅拷贝就可以了：

function deepCopy(obj) {

　　var objNew = objNew || {};

　　for (var i in obj) {

　　　　if (typeof p[i] === 'object') {

　　　　　　objNew[i] = (p[i].constructor === Array) ? [] : {};

　　　　　　deepCopy(obj[i], objNew[i]);

　　　　} else {

　　　　　　　objNew[i] = obj[i];

　　　　}

　　}

　　return objNew;

}

111遍历对象

for(var key in obj){

//只遍历对象自身的属性，而不包含继承于原型链上的属性。

if (obj.hasOwnProperty(key) === true){

keys.push(key);

values.push(obj[key]);

}

}

alert("keys is ："+keys+" and values is ："+values);

111遍历数组

var arr = [1,2,3,4];

arr.forEach(function(value,index,array){

    array[index] == value;    //结果为true

    sum+=value;

    });

console.log(sum);    //结果为 10

111 file

<input type="file" placeholder="" class="col-sm-12" v-on:change="fileReady($event)">

fileReady: **function** (e) {  
 **this**.fileState = **true  
 var** file = e.target.files[0]  
 console.log(e.target)  
},

**var** form = **new** FormData()  
**var** params2 = {  
 couponId: **this**.couponIdModal  
}  
 // form.append("couponId", this.couponIdModal)  
 form.append("excelFilePath", **this**.file)

$.ajax({  
 url:URL+ '/loanBk/coupon/importUsers?'+str,  
 type:'post',  
 // dataType:'json',  
 // contentType:'multipart/form-data; boundary=----WebKitFormBoundarylZiVIC4IH7OVaGwq',  
 headers: pubZ.headers,  
 data: params,  
 processData:**false**, //这个一定要  
 contentType: **false**, //这个一定要  
 success: **function** (result) {  
 **if**(result.code=='200000'){

processData:**false**, //这个一定要  
 contentType: **false**, //这个一定要

异步多ajax，次数不定，jquery ，eval 拼字符 when then

**var** str1 = '$.when('  
 **var** strs = ''  
 **var** str3 = ').then(function(a,b) {\n' +  
 ' var count = 0\n' +  
' console.log(arguments);\n' +  
 ' for(var i = 0; i<arguments.length; i++){\n' +  
 ' console.log("arguments[i]", arguments[i], arguments[i]);\n' +  
 ' console.log("arguments[i][0].code", arguments[i][0].code, arguments[i][0].data);\n' +  
  
 'if(arguments[i][0].code == "200000" && arguments[i][0].data == 1) {\n' +  
 'count++\n' +  
 ' console.log("陈工一个", count);\n' +  
  
 '}\n' +  
'};\n' +  
'if(count == arguments.length){ShowWarningAlert("警告","全部成功")}else{ShowWarningAlert("警告","未全部成功")}\n' +  
'\n' +  
 ' })'  
  
 **var** \_this = **this  
 var** userCouponIdList = []  
 **this**.tableData.forEach(**function** (value) {  
 **if** (value.check == **true**) {  
 /\*userCouponIdList.push(value.userCouponId)  
 strs = strs + 'api\_coupon\_revokeUserCoupon(' + '{userCouponId:'+value.userCouponId+'}' + ', function (result) {\n' +  
  
 ' }),'\*/  
 userCouponIdList.push(value.userCouponId)  
 strs = strs + 'api\_coupon\_revokeUserCoupon(' + '{userCouponId:'+value.userCouponId+'}' + '),'  
}  
 })  
 **var** strLast = str1 + strs +str3  
 console.log('strLast', strLast)  
 **if** (userCouponIdList.length > 0) {  
 eval(strLast)  
 }

api\_coupon\_revokeUserCoupon 方法中一定要返回 jquery AJAX对象

**function** api\_coupon\_revokeUserCoupon(params){  
 **return** $.ajax({  
 url:URL+ '/loanBk/coupon/revokeUserCoupon',  
 type:'patch',  
 dataType:'json',  
 // contentType:'text',  
 headers: pubZ.headers,  
 data: params,  
 });  
}

$.when( $.ajax("test.aspx") ).then(function(ajaxArgs){

alert(ajaxArgs[1]); /\* ajaxArgs is [ "success", statusText, jqXHR ] \*/

});

ajaxArgs是一个数组，是所有请求的详情

eval用 **new** **Function 代替**

//很少有

**var** a = **new** **Function**(param,body);

// 并且body里面只能是string类型.

**if** (userCouponIdList.length > 0) {  
 ShowDeleteAlert('警告', '是否确认批量回收优惠券?', **function** (item) {  
 **var** param = []  
 **var** a = **new** Function(param, strLast)  
 a()  
 // eval(strLast)  
 /\*api\_coupon\_abandonCoupon(userCouponIdList, function (result) {  
 // null  
 })\*/  
 })  
}

浏览器判断

**var** ed = **function** IEVersion () {  
 **var** userAgent = navigator.userAgent // 取得浏览器的userAgent字符串  
 **var** isIE = userAgent.indexOf('compatible') > -1 && userAgent.indexOf('MSIE') > -1 // 判断是否IE<11浏览器  
 **var** isEdge = userAgent.indexOf('Edge') > -1 && !isIE // 判断是否IE的Edge浏览器  
 **var** isIE11 = userAgent.indexOf('Trident') > -1 && userAgent.indexOf('rv:11.0') > -1  
 **if** (isIE) {  
 **var** reIE = **new** RegExp('MSIE (\\d+\\.\\d+);')  
 reIE.test(userAgent)  
 **var** fIEVersion = parseFloat(RegExp['$1'])  
 **if** (fIEVersion == 7) {  
 **return** 7  
 } **else if** (fIEVersion == 8) {  
 **return** 8  
 } **else if** (fIEVersion == 9) {  
 **return** 9  
 } **else if** (fIEVersion == 10) {  
 **return** 10  
 } **else** {  
 **return** 6//IE版本<=7  
 }  
 } **else if** (isEdge) {  
 **return** 'edge'//edge  
 } **else if** (isIE11) {  
 **return** 11 //IE11  
 } **else** {  
 **return** -1//不是ie浏览器  
 }  
}

var menuid=window.parent.CurrentValue;  
var userid=window.sessionStorage.userId;  
温小 2018/3/16 11:05:45  
  
获取iframe外的管理员Id和菜单Id

jquery ajax封装

// 19.1.139 升级消息模板  
**function** api\_sysTemplate\_updateTemplateVersion(params, success, before, after){  
 $.ajax({  
 url:URL+ '/ sysTemplate/updateTemplateVersion',  
 type:'patch',  
 dataType:'json',  
 contentType:'text',  
 headers: pubZ.headers,  
 data: params,  
 success: **function** (result) {  
 **if**(result.code=='200000'){  
 success(result)  
 **switch**(result.data)  
 {  
 **case** 1:  
 window.location.reload();  
 **break**;  
 **case** -1:  
 ShowWarningAlert('警告','操作失败');  
 **break**;  
 **case** -2:  
 ShowWarningAlert('警告','发布状态才能修改');  
 **break**;  
 **case** -3:  
 ShowWarningAlert('警告','validDate, version eventType和contents是必填项');  
 **break**;  
 **case** -4:  
 ShowWarningAlert('警告','如果useType=0,有效开始日期日期小于当前日期');  
 **break**;  
 **default**:  
 ShowWarningAlert('警告','操作失败');  
 }  
 StopLoading();  
 $(".loadingContainer").removeClass('loadingContainer');  
 }  
 **else**{  
 ajaxFail(result);  
 }  
 },  
 beforeSend: **function** () {  
 before()  
 },  
 complete: **function** () {  
 after()  
 },  
 });  
}

拦截，禁止启用按钮

params.data.couponId = **this**.couponIdModal  
paramsStr = JSON.stringify(params)  
api\_coupon\_modifyCoupon(paramsStr, **function** (result) {  
 // null  
}, **function** () {  
 \_this.saveBtnDisabled = **true**}, **function** () {  
 \_this.saveBtnDisabled = **false**})

111tostring

**1、.toString()**可以将所有的的数据都转换为字符串，但是要排除null 和 undefined

例如将false转为字符串类型

|  |  |
| --- | --- |
| 1  2  3  4 | <script>    var str = false.toString();    console.log(str, typeof str);  </script> |

返回的结果为 false，string

看看null 和 undefined能不能转换为字符串[javascript](http://www.100sucai.com/code/2135.html)

|  |  |
| --- | --- |
| 1  2  3  4 | <blockquote style="margin-right: 0px;" dir="ltr"><pre class="html" name="code"><script>    var str = null.toString();    console.log(str, typeof str);  </script> |

结果程序报错

|  |  |
| --- | --- |
| 1  2  3  4 | <script>    var str = undefined.toString();    console.log(str, typeof str);  </script> |

程序也报错

111froala111getSelection

<!-- Include CSS for icons. -->  
<link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.4.0/css/font-awesome.min.css" rel="stylesheet" type="text/css" />  
  
<!-- Include Editor style. -->  
<link href="https://cdnjs.cloudflare.com/ajax/libs/froala-editor/2.5.1/css/froala\_editor.pkgd.min.css" rel="stylesheet" type="text/css" />  
<link href="https://cdnjs.cloudflare.com/ajax/libs/froala-editor/2.5.1/css/froala\_style.min.css" rel="stylesheet" type="text/css" />  
  
<!-- Create a tag that we will use as the editable area. -->  
<!-- You can use a div tag as well. -->  
  
<!-- Include jQuery lib. -->  
<!--<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script>-->  
  
<!-- Include Editor JS files. -->  
<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/froala-editor/2.5.1//js/froala\_editor.pkgd.min.js"></script>  
  
<!-- Initialize the editor. -->

编辑插入东西

insertParams: **function** () {  
 //设置编辑器获取焦点  
 // this.editor.events.focus();  
 $('#editDiv').focus()  
  
 // 获取选定对象  
 **const** selection = getSelection();  
 // 判断是否有最后光标对象存在  
 **if** (**this**.lastEditRange) {  
 // 存在最后光标对象，选定对象清除所有光标并添加最后光标还原之前的状态  
 selection.removeAllRanges()  
 selection.addRange(**this**.lastEditRange)  
 }  
 **var** a = '<u>啊啊啊啊</u>'  
 //插入内容  
 $('#editDiv').froalaEditor('html.insert', 'foo bar', **true**)  
 //记录最后光标停留位置  
 **this**.lastEditRange = selection.getRangeAt(0)  
},

初始化

**var** content = $('#editDiv . fr-element')  
**if** (content.length == 0) {  
 $('#editDiv').html('').froalaEditor({  
 inlineMode: **false**, alwaysBlank: **true**, language: 'zh\_cn',  
 // imageUploadURL: URL + '/picture/imgUpload?token=' + window.sessionStorage.token //上传到本地服务器  
 })  
 $('#editDiv . fr-element').addClass('f-placeholder')  
  
} **else** {  
 content.html('')  
}  
  
$('#editDiv').on('froalaEditor.keyup', **function** (e, editor) {  
 console.log('blur')  
 \_this.lastEditRange = getSelection().getRangeAt(0);  
 // Do something here.  
});  
$('#editDiv').on('froalaEditor.click', **function** (e, editor, clickEvent) {  
 // Do something here.  
 console.log('click')  
 \_this.lastEditRange = getSelection().getRangeAt(0);  
  
  
});

读内容

getEditorHtml:**function**(){  
 //获取编辑框内容  
 **var** activityContent=$("#editDiv . fr-element").html();  
 **if**(activityContent==**false** || activityContent=='<p><br></p>'){  
 ShowNormalAlert('提示','编辑内容为空');  
 }  
 **else**{  
 ShowNormalAlert('编辑内容为',activityContent);  
 }  
}

111input 强制为数字

parseIntNum: **function** (num) {  
 **if**(num===**undefined** || num === **null**){ // **undefined null 不排除会引起**toString报错  
 **return** ''  
 } **else if** (num < 0 || num.toString().trim() === '') {  
 **return** ''  
 } **else** {  
 **if**(isNaN(num)){  
 **return** ''  
 }**else** {  
 **return** Math.floor(Number(num))  
 }  
 }  
},  
parseFix2: **function** (val) {  
 **if**(val===**undefined** || val === **null**){  
 **return** ''  
 } **else if** (val < 0 || val.toString().trim() === '') {  
 **return** ''  
 } **else** {  
 **if**(isNaN(val)){  
 **return** ''  
 }**else** {  
 **return** Number(val).toFixed(2)  
 }  
 }  
},

111file 上传文件后缀名校验

**this**.file = e.target.files[0]  
  
 console.log(e.target.value)  
**var** fileName = e.target.value  
 **var** flag = **false**; //状态  
 **var** arr = ["XLSX","XLS"];  
 //取出上传文件的扩展名  
 **var** index = fileName.lastIndexOf(".");  
 **var** ext = fileName.substr(index+1).toUpperCase();  
 //循环比较  
 **for**(**var** i=0;i<arr.length;i++)  
 {  
 **if**(ext == arr[i])  
 {  
 flag = **true**; //一旦找到合适的，立即退出循环  
 **break**;  
 }  
 }  
 //条件判断  
 **if**(flag)  
 {// document.write("文件名合法");  
 }**else** {  
 ShowWarningAlert('警告','文件名不合法！')  
 }

111上月111近一个月111近一月

autoSetTime: **function** (d) {  
 **var** myDate = **new** Date()  
 **var** endDate = myDate.getTime()  
 **var** endDataStr = pubZ.secondsToDateStr(endDate)  
 **var** thisMonthDay = 0  
 **var** lastMonthDay = 0  
 **this**.$refs.endDate.value = pubZ.secondsToDateStr(endDate)  
 **if** (isNaN(d)) {  
 **var** lastMonthDate = **new** Date()  
 // lastMonthDate.setTime(pubZ.dateStrToSeconds(this.$refs.endDate.value)) // 测试  
 **var** thisMonthFirst = 0  
**var** lastMonthFirst = 0  
 **if**(lastMonthDate.getMonth() > 0 ) {  
 lastMonthDate.setDate(1)  
 thisMonthFirst = lastMonthDate.getTime()  
 lastMonthDate.setMonth(lastMonthDate.getMonth()-1)  
 // lastMonthDate.setDate(1)  
 lastMonthFirst = lastMonthDate.getTime()  
 console.log('当前日期',thisMonthFirst/(24 \* 3600 \* 1000), '上月天数',lastMonthDay)  
 lastMonthDay = (thisMonthFirst - lastMonthFirst)/(24 \* 3600 \* 1000)  
  
 lastMonthDate = **new** Date() //测试结束放开  
 // lastMonthDate.setTime(pubZ.dateStrToSeconds(this.$refs.endDate.value)) // 测试  
 **if**(lastMonthDate.getDate() > lastMonthDay){  
 //现在日期大于上个月天数  
 // console.log('当前月份', lastMonthDate.getMonth())  
 **var** temp = lastMonthDate.getMonth() - 1  
 lastMonthDate.setMonth(temp)  
 lastMonthDate.setMonth(temp)  
  
 lastMonthDate.setDate(lastMonthDay)  
 }**else** {  
 lastMonthDate.setMonth(lastMonthDate.getMonth()-1)  
 }  
} **else** {  
 // 1月 12月都是31天  
 console.log('1月 12月都是31天')  
 lastMonthDate.setFullYear(lastMonthDate.getFullYear() -1)  
 lastMonthDate.setMonth(11)  
 }  
 **this**.$refs.beginDate.value = pubZ.secondsToDateStr(lastMonthDate.getTime())  
  
/\*if(lastMonthDate.getMonth() > 0 ) {  
 lastMonthDate.setMonth(lastMonthDate.getMonth-1)  
}\*/  
 } **else** {  
  
 **var** beginDate = endDate - d \* 24 \* 3600 \* 1000  
 **this**.$refs.beginDate.value = pubZ.secondsToDateStr(beginDate)  
  
 }  
 },

111校验111tostring111trim

inputCheck: **function** (val, type, num) {  
 **if** (type === 'checkSpace') {  
 **if** ((val !== **null** && val !== **undefined**) && val.toString().trim() === '') {  
 **return true** // 空未填返回true  
 } **else** {  
 **return false** }  
 } **else if** ('checkLength') {  
 **if** (val !== **null** && val !== **undefined**) {  
 console.log('checkLength', val.toString().replace(/[\u4e00-\u9fa5]/g,'aaa'))  
 **if**(val.toString().replace(/[\u4e00-\u9fa5]/g,'aaa').length > num){  
 **return true** } **else** {  
 **return false** }  
 } **else** {  
 **return false** }  
 } **else if** ('checkNumLength') {  
 **if** (val !== **null** && val !== **undefined**) {  
 **if**(val.toString().trim().length > num){  
 **return true** } **else** {  
 **return false** }  
 } **else** {  
 **return false** }  
 }  
},  
toString: **function** () {  
 **if** ((val !== **null** && val !== **undefined**)) {  
 **return** val.toString()  
 } **else** {  
 **return** ''  
 }  
},  
trim: **function** () {  
 **if** ((val !== **null** && val !== **undefined**)) {  
 **return** val.toString().trim()  
 } **else** {  
 **return** ''  
 }  
},

isNaN 用法很怪NaN不等于NaN，别用，es5 Number.isNaN可以片段是否是NaN。要判断是否是数字typeof() === ‘number’

111强制保留2位小数正数111保留2位

<input class="col-sm-12" type="text" placeholder="" v-model="amount" :disabled="showLock" v-on:input="parseFloatNum">

// last 为上次正确输入的值  
parseFloatNum: **function** (num, last) {  
 **if**(num===**undefined** || num === **null**){  
 **return** ''  
 } **else if** (num < 0 || num.toString().trim() === '') {  
 **return** ''  
 } **else** {  
 **if**(/^[1-9]\d\*\.?\d{0,2}$/.test(num)){  
 **return** num  
 } **else** {  
 **return** last  
 }  
 }  
},

Temp

1.我的账号模块是没有内容的？

2.课堂审批模块是没有内容的？

3.课堂列表

3.1课堂资源列表，转码中状态文档上没有，补一下。

3.2课堂资源，新建资源没有，

3.3课堂资源，更新资源没有。

4.在线考试

4.1在线考试，考试统计没有

4.2在线考试，学生试卷列表没有

(查看未参加，查看答题，因为是学生试卷列表后面的流程，我也不知到有没有，没有的话你帮我补上)

4.3在线考试，删除考试没有

4.4在线考试，删除考试没有

4.5在线考试，监控考试是不是就是监控互动接口？你是不是定时调的，多久一次好一些

4.6试题答案编辑，哪个接口？没有的话补上 （获取和提交两个接口吧？）