常用技巧

2019年10月2日

19:38

**常用技巧：**

mysql日期转换为字符串 date.format(hiredate,"%Y-%m-%d");

字符串转换为mysql日期 str\_to\_date(hiredate,"%Y-%m-%d");

字符串转换为int int I = Interger.parseInt(str)

转换日期为特定类型字符串 str = simpledateformat(date,"yyyy年MM月dd日[hh:mm:ss]")

设置jQuery对象数组的属性 $("input[name=sub]").prop("要修改的属性",this.checked);

**删除目录2下面的所有文件及其文件夹:**

删除该目录下内容:

String url = "D:/test/1/2";

boolean bol = FileUtils.deleteQuietly(new File(url));

删除父目录下所有内容

String url = "D:/test/1/2"; boolean bol = FileUtils.deleteQuietly(new File(url).getParentFile());

**EL格式化输出数字**

<fmt:formatNumbervalue="${ticket.price}"pattern="#,#00.00#"/>

**jstl格式化字符串日期**

在对字符串日期进行格式化式化输出时，需要进行两次转化：如下

<fmt:parseDate value="20081214121011" pattern="yyyyMMddHHmmss" **var="test"**/>

<fmt:formatDate **value="${test}**" pattern="yyyy-MM-dd HH:mm:ss"/>

**利用layui显示td隐藏内容**

//描述鼠标悬停显示全部内容

$("td").on("mouseenter",function(){

if(this.offsetWidth<this.scrollWidth){

var that=this;

var text=$(this).text();

layer.tips(text,that,{

tips:1,

time:2000

});

}

});

**利用bootstrap定制个性化导航条**

**html:**

<div class="navbar my-navbar" role="navigation">

<div class="navbar-header">

<a class="navbar-brand" href="#">

<img src="img/logo.png" alt="logo">

</a>

<button class="navbar-toggle collapsed" data-toggle="collapse" data-target="#navmenu">

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

</div>

<div id="navmenu" class="collapse navbar-collapse">

<ul class="nav navbar-nav navbar-right">

<li class="active"><a href="#">首页</a></li>

<li><a href="#">案例展示</a></li>

<li><a href="#">相关技术</a></li>

<li><a href="#">友情链接</a></li>

</ul>

</div>

</div>

**css:**

.my-navbar{

background: #2b2996;

}

.my-navbar .navbar-brand{

padding: 10px 15px;

}

.my-navbar .navbar-nav>li>a{

color: #fff;

}

.my-navbar .navbar-nav>li>a:hover,

.my-navbar .navbar-nav>li>a:focus,

.my-navbar .navbar-nav>li.active>a{

color: #009bf7;

background: none;

}

.my-navbar .navbar-toggle{

border: 1px solid #fff;

}

.my-navbar .navbar-toggle .icon-bar{

background: #fff;

}

**如何获取表单的所有input对象**

function form2Json(formId) {

var d = {};

var t = $('#' + formId).serializeArray();

$.each(t, function() {

d[this.name] = this.value;

});

return d;

}

**验证手机电话格式**

<input id="phone" name="phone" label="移动电话"

         data-options="required:true,validType:'phoneAndMobile'"

class="easyui-textbox" style="width:100%">

**集合框架排序:**

list\_ticket.sort(Comparator.comparingInt(Tickets::getTickno).reversed());

Collections.sort(list\_ticket,newComparator<Tickets>(){

@Override

public int compare(Ticketstickets\_1,Ticketstickets\_2){

return (tickets\_2.getTickno()-tickets\_1.getTickno());

}

});

使用lambda表达式代替匿名函数:

Collections.sort(list\_ticket,(Ticketst1,Ticketst2)->t2.getTickno()-t1.getTickno());

**lambda表达式**:(重写方法参数)->主要内容; 用于实现简单的接口,主要内容可以使用大括号括起来,写java语句.

Collections.sort(list\_ticket,(Ticketst1,Ticketst2)->t2.getTickno()-t1.getTickno());

**限制:**

1）-> 运算符具有与赋值运算符 (=) 相同的优先级，并且是右结合运算符。

2）在 is 或 as 运算符的左侧不允许使用 Lambda。

3）适用于匿名方法的所有限制也适用于 Lambda 表达式，如：Lambda 语句无法用于创建表达式树。

4）只有在 Lambda 有一个输入参数时，括号才是可选的，否则括号是必需的，两个或更多输入参数由括在括号中的逗号分隔。

5）有时，编译器难于或无法推断输入类型，如果出现这种情况，显式指定类型

**Predicate的泛型委托:**

例子:

//获取Array中长度不大于3的字符串,返回String[]类型

使用Predicate获取list中字符串长度大于3的元素,使用lambda表达式

Predicate<String>predicate=p->p.length()>3;

list=list.stream().filter(predicate).collect(Collectors.<String>toList());

正则表达式转变为Predicate,选择集合中合法的字符串

Predicate<String>predicate=Pattern.compile("56$").asPredicate();

list=list.stream().filter(predicate).collect(Collectors.<String>toList());

Supplier:

Supplier<Employee>supplier=Employee::new;

::-----方法引用,通过一个方法的名字指向一个方法

**获取滚动条的位置**

var sTop=document.body.scrollTop+document.documentElement.scrollTop;

注释:document.body.scrollTop 和 document.documentElement.scrollTop只有一个生效,另一个值为 0 ;

**省市级联 :**

select-option应用

**select对象的常用事件、属性、方法**

onchange:当选项发生改变时触发

selectedIndex:获取当前被选中的选项的下标

length:获取或设置下拉框选项的数量

text:当前被选中选项的纯文本

value:当前被选中选项的value值

add(new Option(text,value)，null);新增一个选项

**JS部分:**

<script>

//文字下标，二维数组

var cityList=newArray();

cityList['山东省']=['济南市','青岛市','烟台市'];

cityList['河南市']=['郑州市','洛阳市','开封市'];

cityList['湖北市']=['武汉市','宜昌市','荆州市'];

function addCity(){

//获取当前被选中的省份

var province=document.getElementById("province").value;

//找到city,清空下拉列表

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

city.options.length=0;

for(var I in cityList){

if(i==province){

for(var j in cityList[i]){

city.add(new Option(cityList[i][j],cityList[i][j]),null)

}

}

}

}

</script>

多维数组转换为单维数组

<<多维数组转换为单维数组.txt>>

**判断当前页面是否在移动端打开:**

'name' in p1