# 12月

## 12-20

除了button设置disable之后，不能使用click，其他的需要off click

一个 a 没有 href 也没有 tabindex 就不会被 tab 到

HTML 4 认为没有 href (但是有 name) 的 a 是命名锚点

HTML 5 认为没有 href 的 a 是链接占位符

(反正对浏览器来说行为都是一样的

datatable：tostring为0

追加操作需要判断是否已经把原值清空

## 12-19

放弃融合：$ git merge --abort

重新设置：$ git reset --merge

提交融合错误：

1、放弃本地修改：$ git reset --hard origin/master

或者$ git reset --hard ORIG\_HEAD

2、手动修改，删掉其中冲突的部分，然后运行如下命令

$ git add filename

$ git commit -m mark

1、事件冒泡：清除click事件冒泡、为body

2、添加中间变量，blur时判断变量；mousedown设置变量

$td.text("<CarNo>")：text自动转换<> &lt;&gt;

## 12-18

### 请求：

1、初始化请求

2、发送请求

3、获取反应数据

var request = (HttpWebRequest)WebRequest.Create("http://192.168.0.7/api/Huo/SeachHuo");

request.Headers.Set("userid", "3244");

DateTime nowtime = DateTime.Now;

long unixTime = (nowtime.Ticks - new DateTime(1970, 1, 1).Ticks) / 10000000 - 8 \* 60 \* 60;

request.Headers.Set("nowtime", unixTime.ToString());

request.Headers.Set("token", Common.DESEncrypt.HexToMd5("C0EFD0CFB11FB62B" + Common.DESEncrypt.HexToMd5(nowtime.ToString("yyyyMMddHHmmss"), 16).ToUpper(), 32).ToUpper());

request.Timeout = 180000;//两分钟  
request.ReadWriteTimeout = 180000;//两分钟  
request.KeepAlive = false;

var data = Encoding.ASCII.GetBytes(postData.ToString());

request.Method = "POST";

request.ContentType = "application/json";

request.ContentLength = data.Length;

using (var stream = request.GetRequestStream())

{

stream.Write(data, 0, data.Length);

}

var response = (HttpWebResponse)request.GetResponse();

var responseString = new StreamReader(response.GetResponseStream()).ReadToEnd();

HttpRequest

Browser

Header：NameValueCollection

TagBuilder

标签：属性、文本、特性、类

--把字符串展示为html：MvcHtmlString（string）

TagBuilder tag = new TagBuilder("ul");

foreach (string str in list)

{

TagBuilder itemTag = new TagBuilder("li");

itemTag.SetInnerText(str);

tag.InnerHtml += itemTag.ToString();

}

tag.MergeAttribute("href", pageUrl(i)); //定义标签的特性

tag.InnerHtml = i.ToString();//文本

tag.AddCssClass("btn btn-default"); //添加类

result.Append(tag.ToString());

return new MvcHtmlString(tag.ToString());

InnerHtml 这是一个以HTML字符串来设置元素内容的属性。赋给这个属性值将不进行编码，意即可以将它嵌入HTML元素

SetInnerText(string) 设置HTML元素的文本内容，string参数将被编码，以使它安全显示

AddCssClass(string) 对HTML元素添加一个CSS的class

MergeAttribute(string,string,bool) 对HTML元素添加一个标签属性。第一个参数是标签属性名称，第二个是它的值。bool参数指定是否替换已存在的同名标签属性

string result = string.Format("This is the message:<p>{0}</p>", html.Encode(msg));

File.ReadAllText(url,Encoding)

File.WriteAllText(url,text,Encoding)

### postition：

|  |  |
| --- | --- |
| absolute | 生成绝对定位的元素，相对于 static 定位以外的第一个父元素进行定位。  元素的位置通过 "left", "top", "right" 以及 "bottom" 属性进行规定。都变成块级元素 |
| fixed | 生成绝对定位的元素，相对于浏览器窗口进行定位。  元素的位置通过 "left", "top", "right" 以及 "bottom" 属性进行规定。 |
| relative | 生成相对定位的元素，相对于其正常位置进行定位。原来的空间保留  因此，"left:20" 会向元素的 LEFT 位置添加 20 像素。 |
| static | 默认值。没有定位，元素出现在正常的流中  （忽略 top, bottom, left, right 或者 z-index 声明）。 |
| inherit | 规定应该从父元素继承 position 属性的值。 |

top\left有默认值

## 12-16

http://www.haorooms.com/post/checkandselect

### select：

val：返回选中option中的value值，如果没有则返回text的值。

text：返回所有option中的text，有换行

有预选值的：selected="selected"

$("select option:eq(1)").prop("selected","selected"/true); //不会修改html

$("select option:eq(1)").attr("selected","selected"/true);//true会自动替换为selected

[autofocus](http://www.w3school.com.cn/tags/att_select_autofocus.asp)\[disabled](http://www.w3school.com.cn/tags/att_select_disabled.asp)\

js：selectedIndex=2；从0开始 ；；get返回dom元素

<optgroup label="Swedish Cars">

<option value ="volvo">Volvo</option>

<option value ="saab">Saab</option>

</optgroup>

## checkbox

<input type="checkbox" id="chb" value="123"/>

val：value为空时，返回on

checked：attr返回undefined，prop返回true/false；设置或获取

$("#chb").prop("checked",""/false/true/任意值);

只要有checked就选中，不管值是多少

可以使用removeAttr

:checked：匹配选中的input元素

:disabled--:enable

$("input:enabled").prop("disabled",true);不能有空格

全选：

$("#checkAll").click(function() {

$('input[name="subBox"]').attr("checked",this.checked);

});

## 12-15

charset=gb2312: 中文编码

charset=utf-8：国际标准编码

<link href="http://www.chinawutong.com/css/WTMap.css?t=1603171332" rel="stylesheet" />

<script src="http://www.chinawutong.com/js/WTMap.min.js?t=1603261015" charset="gb2312"></script>

### 下拉框

：一个输入框、一个文本框<url class="dropdown-menu">

效果： 1、输入值，出现文本框；失去焦点隐藏文本框

2、点击文本框，选中值，并出现在文本框中

其他：上下键选中，enter输出

为输入框绑定：keyup和blur

文本框绑定：mousedown

<input name="endTime" id="endTime" readonly="" onclick=" WdatePicker({ startDate: '%y-%M-01', dateFmt: 'yyyy-MM-dd', alwaysUseStartDate: true, readOnly: true, minDate: '#F{$dp.$D(\'startTime\')}' })" value="">

### 获取元素位置方法

offset:top/left 相对于文档

position: 相对于偏移父元素

width/innerWidth/outerWidth

### 获取属性

ES5中定义两个枚举的属性名称的属性：Object.keys(object),它返回一个数组，这个数组由可枚举的自由属性的名称组成；Object.getOwnPropertyNames(object)，它返回所有自由属性的名称，而不仅仅是可枚举的属性。

### CSS：

行内：style="width:100px;height:100px;background:red;"

内嵌： <style type="text/css">

#div{width:100px;height:100px;background:red;}

</style>

外部：<link rel="stylesheet" type="text/css" href="ccss.css">

### ul

：list-style-type：none

list-style:square inside url('/i/arrow.gif');

margin：0//有默认高度

### 圆角、阴影、

border-radius：5px

border:1px solid #ddd;

box-shadow: 0 6px 12px rgba(0,0,0,.175); box-shadow: h-shadow v-shadow blur spread color inset;

rgba()最后一个是透明：1-0 opacity blur:模糊距离，边到中间的距离 阴影相等于，在本身的下面设置一个同样的背景标签

:first-letter 向文本的第一个字母添加特殊样式。 1

:first-line 向文本的首行添加特殊样式。 1

:before 在元素之前添加内容。 2

:after 在元素之后添加内容。 2

### 冒号和双冒号

css3为了区分伪类和伪元素，伪元素采用双冒号写法。

常见伪类——:hover,:link,:active,:target,:not(),:focus。

选择元素的第一个子元素

常见伪元素——::first-letter,::first-line,::before,::after,::selection。

::before和::after下特有的content，用于在css渲染中向元素逻辑上的头部或尾部添加内容。

这些添加不会出现在DOM中，不会改变文档内容，不可复制，仅仅是在css渲染层加入。

所以不要用:before或:after展示有实际意义的内容，尽量使用它们显示修饰性内容，例如图标。

<style type="text/css">

.phoneNumber::before {

content:'\260E';

font-size: 15px;

}

</style>

<p class="phoneNumber">12345645654</p>

content用来定义插入的内容，content必须有值，至少是空。

通过attr()调用当前元素的属性，比如将图片alt提示文字或者链接的href地址显示出来。

url("https://www.baidu.com/img/baidu\_jgylogo3.gif");url/uri用于引用媒体文件。

counter()调用计数器，可以不使用列表元素实现序号功能。配合counter-increment和counter-reset属性使用：h2:before { counter-increment: chapter; content: "Chapter" counter(chapter) ". " }

对于CSS2之前已有的伪元素，比如:before，单冒号和双冒号的写法::before作用是一样的。

### 悬浮

注意： 在CSS定义中，a:hover 必须被置于 a:link 和 a:visited 之后，才是有效的。

注意： 在 CSS 定义中，a:active 必须被置于 a:hover 之后，才是有效的。

:link 未访问

visited

:hover 悬浮

active 正在活动链接

### 缩放、旋转

transform: scale(2);图片缩放 scale(x，y);

transform:rotate(-30deg);

scale3d(x,y,z)

rotate3d(x,y,z,angle) 定义 3D 旋转。

rotateX(angle) 定义沿着 X 轴的 3D 旋转。 测试

rotateY(angle) 定义沿着 Y 轴的 3D 旋转。 测试

rotateZ(angle) 定义沿着 Z 轴的 3D 旋转。

### 兼容模式和快速模式

国内的厂商一般把其他内核叫做“高速浏览模式”，而Trident则是“兼容浏览模式”，用户可以来回切换。

但是IE内核速度比较慢，所以现在的浏览器都使用一个叫Webkit的内核，这个内核浏览速度比IE快很多，因为加入了新的内核，所以现在的浏览器体积都比较大，但是因为国内的大部分用户都是用IE，所以很多网站都是针对IE开发的，用其他的浏览器核心会显示不正常，所以出现了双核浏览器，平时使用Webkit内核，即极速模式，遇到显示不正常的网页就换成IE内核，即兼容模式。

Rendering Engine”，可大概译为“渲染引擎”，不过我们一般习惯将之称为“浏览器内核”。负责对网页语法的解释（如标准通用标记语言下的一个应用HTML、JavaScript）并渲染（显示）网页。 所以，通常所谓的浏览器内核也就是浏览器所采用的渲染引擎，渲染引擎决定了浏览器如何显示网页的内容以及页面的格式信息。

Webkit：苹果公司自己的内核，Safari浏览器

Trident(IE内核)

Gecko(Firefox内核)

### 追加类、移除类

this.className += " iehover";

this.className = this.className.replace(new RegExp(" iehover\\b"), "");

viewer.js

http://www.dowebok.com/192.html

### 触发事件：

trigger()方法触发事件后，会执行浏览器默认操作。

例如：

$("input").trigger("focus");

以上代码不仅会触发为<input>元素绑定的focus事件，也会使<input>元素本身得到焦点（这是浏览器的默认操作）。

如果只想触发绑定的focus事件，而不想执行浏览器默认操作，可以使用jQuery中另一个类似的方法——triggerHandler()方法。

$("input").triggerHandler("focus");

该方法会触发<input>元素上绑定的特定事件，同时取消浏览器对此事件的默认操作，即文本框只触发绑定的focus事件，不会得到焦点。

js 及jq 点击别的标签触发 a 标签点击事件

用js 实现：

只需在触发事件中直接加入下列代码即可，其中ID即为a 标签的ID

document.getElementById("ID").click()

### 触发a的点击事件

用jq实现时；

可在a 标签中加入一个span标签或其他标签：

<a ><span id='test'></span></a>

然后在触发事件中加入如下代码即可：

$("#test").click();

（1）change事件 触发事件必须满足两个条件：

a）当前对象属性改变，并且是由键盘或鼠标事件激发的（脚本触发无效）

b）当前对象失去焦点(onblur)

（2）keypress 恩，还好。。。。。就是能监听键盘事件，鼠标复制黏贴操作他就无能为力的赶脚了。。。。。

（3）propertychange（ie）和input事件 //不能触发js动态修改的

input是标准的浏览器事件，一般应用于input元素，当input的value发生变化就会发生，无论是键盘输入还是鼠标黏贴的改变都能及时监听到变化。

propertychange，只要当前对象属性发生改变。

## 12-12

### checkbox

$("#checkbox-id").checked

$('#checkbox-id').is(':checked')

$('#checkbox-id').attr('checked')

js 属性tagName：大写字母

### 属性”与“特性

JQ1.6之后，“属性”与“特性”做了区别，属性指的是“name，id”等等，特性指的是“selectedIndex, tagName, nodeName”等等。

### 事件

change：焦点离开时触发事件 ---当用于 select 元素时，change 事件会在选择某个选项时发生。当用于 text field 或 text area 时，该事件会在元素失去焦点时发生。

focus：获得焦点时触发事件

keyup：按键弹起时触发事件

keypress：按键按下时触发事件（先响应事件，再显示输入结果（获得的是上一次结果），可能被输入法拦截）

propertychange：属性改变时触发事件（不管是获得焦点还是value改变等）ie有效

### 点击事件的执行顺序：

单击(click)：mousedown，mouseout，click；

双击(dblclick)：mousedown，mouseout，click ， mousedown，mouseout，click，dblclick；

mousedown->（other）blur->mouseup->click

## 12-11

### jq删除元素：

remove() - 删除被选元素（及其子元素）

empty() - 从被选元素中删除子元素

### 添加

append() - 在被选元素的结尾插入内容

prepend() - 在被选元素的开头插入内容 $("p").append(txt1,txt2,txt3); // 追加新元素

after() - 在被选元素之后插入内容

before() - 在被选元素之前插入内容

var txt1="<p>Text.</p>"; // 以 HTML 创建新元素

var txt2=$("<p></p>").text("Text."); // 以 jQuery 创建新元素

var txt3=document.createElement("p");

txt3.innerHTML="Text."; // 通过 DOM 来创建文本

### 同胞元素：

siblings() 所有同胞元素；不包括自己--$("h2").siblings("p");可加参数进行筛选

next() 下一个同胞元素

nextAll() 所有跟随的同胞元素

nextUntil() 返回介于两个给定参数之间的所有跟随的同胞元素；$("h2").nextUntil("h6");

prev() 同上；前面

prevAll()

prevUntil()

### 过滤：

first(), 被选元素的首个元素

last() 被选元素的最后一个元素

eq() 索引号从 0 开始

filter() 不匹配这个标准的元素会被从集合中删除，匹配的元素会被返回。 $("p").filter(".intro");

not() 与filter相反

private void dgvData\_RowStateChanged(object sender, DataGridViewRowStateChangedEventArgs e)

{

e.Row.HeaderCell.Value = (e.Row.Index + 1).ToString();

}

### insert into table 插入多条数据

方法1：

insert into `ttt`

select '001','语文' union all

select '002','数学' union all

select '003','英语';

方法2：

INSERT INTO tab\_comp VALUES(item1, price1, qty1),

(item2, price2, qty2),

(item3, price3, qty3);

方法3：

INSERT INTO tab\_comp(item1, price1, qty1) SELECT item1, price1, qty1 FROM tab\_cc;

string.Join(",",list)

F1

### 获取数据库中所有的表名

[select](https://www.baidu.com/s?wd=select&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dBmHuBmvFWmH--m1-Buymk0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnH6vrjbsrj04) name [from](https://www.baidu.com/s?wd=from&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dBmHuBmvFWmH--m1-Buymk0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnH6vrjbsrj04) sys.tables order by name go

### 读取excel内容

string strConn

= "Provider=Microsoft.Ace.OleDb.12.0;" + "data source=" + FileFullPath + ";Extended Properties='Excel 12.0; HDR=NO;IMEX=1'";//此连接可以操作.xls与.xlsx文件

OleDbConnection conn = new OleDbConnection(strConn);

conn.Open();

DataSet ds = new DataSet();

OleDbDataAdapter odda = new OleDbDataAdapter(string.Format("SELECT \* FROM [{0}$]", SheetName), conn); //("select \* from [Sheet1$]", conn);

odda.Fill(ds, SheetName);

conn.Close();

return ds.Tables[0];

### OpenFileDialog

OpenFileDialog openFile = new OpenFileDialog();

openFile.Filter = "\*.xls|\*.xlsx";//文件类型筛选文本文件(\*.txt)|\*.txt|所有文件(\*.\*)||\*.\*

openFile.RestoreDirectory = true;//记录目录

if (openFile.ShowDialog() == DialogResult.OK)

{

DataTable dt = GetExcelToDataTableBySheet(openFile.FileName, "Sheet1");

### DataGridView添加序号

private void dgvData\_RowStateChanged(object sender, DataGridViewRowStateChangedEventArgs e)

{

e.Row.HeaderCell.Value = (e.Row.Index + 1).ToString();

}

#### 列

dgvData.Columns：所有列

dataGridView1.Columns[0].DataPropertyName= "id"；绑定列

HeaderText：行头

#### 行

## 12-10

### 设置option选中：

document.getElementById("sel")[2].selected=true;

var val = document.all.Item.options[document.all.Item.selectedIndex].text

var i=document.getElementById( 'sel' ).options[document.getElementById( 'sel' ).selectedIndex].value;

$(".selector").find("option[text='pxx']").attr("selected",true);

$(".selector").find("option:selected").text();

类库中不能使用Session，可以使用Cache代替；

### a标签滚动条不跳转：

<a href="####" class="vClicks">

onclick="javascript:void(0)"

((Model.User)Session["userinfo"]).ID

box{color:red !important;},加上一个“!important”就优先于正常的CSS规则。

Encoding.ASCII.GetBytes==Encoding.Default.GetBytes

HttpCookie

??后面不能是int

if (location.href.toString().indexOf("/Main/Index")>-1) {

$("#side-menu a:eq(0)").addClass("active");

}

2.成功方案<更适用于非button标签>：

//亲测可用

$(".edit").prop("onclick",null).off("click");//jQuery1.7+

$(".edit").attr('onclick','').unbind('click');//jQuery-1.7

3. button标签可用如下简便方式

$(".edit").prop("disable",true);//使用disable属性就可以禁止click触发

<button>居中，需要包含在等宽的div中（margin:0 auto或改为inline），a只需要设置text-align:center就行

### 获取class：

var p\_class = $("p").attr("class"); //获取p元素的class

$("p").attr("'class", "high"); //设置p元素的class为 "high"

$("#nm\_p").addClass("another"); // 追加样式

1 $("p").removeClass("high"); //移除p元素中值为"high"的class

果要把p元素的两个class都删除，就要使用两次removeClass()方法，代码如下：

1 $("p").removeClass("high").removeClass("another");

Query提供了更简单的方法。可以以空格的方式删除多个class名，代码如下：

1 $("p").removeClass("high another");

外，还可以利用removeClass()方法的一个特性来完成同样的效果。当它不带参数时，就会将class的值全部删除，JQuery代码如下：

1 $("p").removeClass(); //移除p元素的所有class

换样式 JQuery中有一个方法toggle()，JQuery代码如下：

1 toggleBtn.toggle(function(){

2 //元素显示 代码③

3 }, function(){

4 //元素隐藏 代码④

5 })

给标签绑定事件，后来重现加载标签，没有绑定

jQuery注册事件：on("click",function(){ method(arguments); })

程序再本地可以，发布到IIS上出现错误：不能找到Entities，引入EntityFramework.SqlServer

指定的架构无效。错误: WTTCModel.ssdl(2,2) : 错误 0152: 未找到具有固定名称“S

### TimeSpan：

时间间隔 格式：00:00:00.0000036 （天.时：分：秒.纳秒）

TimeSpan ts0 = new TimeSpan(36); //将时间间隔ts0初始化为36刻度数

TimeSpan ts1 = new TimeSpan(4, 20, 33); //将时间间隔ts1初始化为4小时20分33秒

TimeSpan ts2 = new TimeSpan(2,10,36,45); //将时间间隔ts2初始化为2天10小时36分45秒

TimeSpan ts3 = new TimeSpan(6, 12, 56, 45,23); //将时间间隔ts3初始化为6天12小时56分45秒23毫秒 6.12:56:45.0230000

Days 天

Hours 小时

Minutes 分

Seconds 秒

Milliseconds 毫秒（取纳秒前三位）

Ticks 纳秒

TotalDays 总天

TotalHours

TotalMinutes

TotalSeconds

TotalMilliseconds

DateTime.Now.Ticks ：636483204398117985 此属性的值为自 0001 年 1 月 1 日午夜 12:00 以来所经过时间以 100 毫微秒为间隔表示时的数字。

DateTime：2017/12/8 9:00:39

TimeSpan：736670.09:00:39.8078032

### CSS选择器

类 .class

id #id

所有 \*

元素 p什么都不加

同一 ,

内部子元素 >

后面第一个元素 +

内部所有元素 空格

属性 [] [name=value]等于= 包含~= 匹配开头|=

: 单冒号(:)用于CSS3伪类，双冒号(::)用于CSS3伪元素。

未被访问 :link

已被访问 :visited

已激活 :active

指针悬浮 :hover

获取焦点 :focus

获取首字母 :first-letter

获取首行 :first-line

获取第一个子元素:first-child

获取前面的 :before

获取后面的 :after

CSS3

获取前面有 element~element

# 11月

## 第五周

## 12-2

首先.map文件是何许人也，min.map文件是在压缩文件时，为还原原有未压缩js文件中相关变量而存在的伴侣文件，也就是文件的变量名替换对应信息、变量所在位置等元信息数据文件，一般这种文件和min.js主文件放在同一个目录下。如果发现不存在.min.map文件的话，jquery会检测浏览器是否支持source map功能，如果支持的话，那就去下载source map文件，比如chrome浏览器支持这种操作；支持这种功能的前提是使用官网的min.js文件，比如我使用的貌似不是官网的文件就会一直的报错。

这种报错只是在使用jquery.min.js的时候会遇到这种报错，使用非压缩文件的时候不会，因为这个问题产生的原因是在压缩的时候出现的。网上提供的解决方法有几种：

1.使用非压缩的JS文件，但是非压缩的和.min.js相比影响传输速度，所以要权衡考虑；

2.在官网下载.min.map文件；

3.chrome浏览器关闭下载source map选项,具体操作位chrome->开发者选项->右下角的设置->关闭source map js的功能，比如我现在用的是这种，不过很不好的一点是只能在我目前的浏览器上不会报错，换到了其他人的浏览器依然会报错，因此还需要用其他的方式去修改；

4.在压缩的js中删除min.map文件引用,即删除注释//@ sourceMappingURL=jquery-1.10.2.min.map，网上说这是最优方案；

background-image: url(bgimage.gif);

background-repeat，

它是取值：

repeat :???? ?默认值。背景图像在纵向和横向上平铺

no-repeat :?? 背景图像不平铺

repeat-x :??? 背景图像仅在横向上平铺

repeat-y :??? 背景图像仅在纵向上平铺

背景图片的位置控制

background-position : length（百分数 | 由浮点数字和单位标识符组成的长度值）??| position （top | center | bottom | left | center | right）

去掉ul前面的空格：padding：0 margin：0

实心：list-style-type: none;

input[type=radio], input[type=checkbox]{

margin-right:5px;

cursor:pointer;

}

jquery里我们要获取某个元素的相邻元素时，可以用到的命令有三个：

next():用来获取下一个同辈元素。

prev():用来获取上一个同辈元素。

siblings()：用来获取所有的同辈元素。

cursor其他取值

auto ：标准光标

default ：标准箭头

pointer, hand ：手形光标

wait ：等待光标

text ：I形光标

vertical-text ：水平I形光标

no-drop ：不可拖动光标

not-allowed ：无效光标

help ：帮助光标

all-scroll ：三角方向标

move ：移动标

crosshair ：十字标

$("input:checkbox").removeAttr("checked");

$("input:checkbox").prop("checked","checked");

this.checked=!this.checked;

jObject[0].removeAttr("checked");

<hr>横线 margin-top：20 bottom：20

按钮

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>CSS制作按钮</title>

<style type="text/css">

.btn{

width:100px;

height:100px;

background:#ccc;

border-radius:50%;

box-shadow:5px 5px 10px #000; /\*设置外阴影\*/

}

.btn:active{

background:#bbb;

box-shadow:5px 5px 10px #000 inset; /\*设置内阴影\*/

}

.btn div{

font-size:30px;

font-family:"微软雅黑";

color:blue;

float:left;

margin-top:28px;

margin-left:20px;

}

</style>

</head>

<body>

<div class="btn">

<a href="###">

<div>开始</div>

</a>

</div>

</body>

</html>

三角框

<style>

.arrow{

width:0;

height:0;

border-top:50px solid #000; /\*设置并显示上边框\*/

border-bottom:none; /\*不设置下边框\*/

border-left:50px solid transparent; /\*设置但隐藏左边框\*/

border-right:50px solid transparent; /\*设置但隐藏右边框\*/

}

.arrow:hover{

border-top:none; /\*鼠标经过时，不设置上边框\*/

border-bottom:50px solid #000; /\*鼠标经过时，设置并显示下边框\*/

}

</style>

</head>

<body>

<div class="arrow"></div>

</body>

radio对其、position、float

float：后面的元素

如果要全部替换的话，JS 没有提供replaceAll这样的方法。使用正则表可以达成Replace 的效果

str.replace(/word/g,"Excel")

## 12-1

取消冒泡（传播）：

var e=event||window.event;

e.stopPropagation();

或：执行完语句之后return false;

取消默认行为：

event.preventDefault()

IE:window.event.returnValue = false;

event属性：

altKey, button, cancelBubble, clientX, clientY, ctrlKey, fromElement, keyCode, offsetX, offsetY, propertyName, returnValue, screenX,

screenY, shiftKey, srcElement, srcFilter, toElement, type, x, y

获取对象的属性：

1、for/in：访问所有的属性，包括原型链；可以加判断hasOwnProperty()

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]);

}

}

2、jQuery.each(arr,function(){i,val}{})

var e=event||window.event;

e.stopPropagation();

var str="";

jQuery.each(e,function(i,val){str+=i+":"+val+"\r\n";})

alert(str)

select col.name,pro.value from sys.syscolumns col

left join sys.extended\_properties pro

on col.colid=pro.minor\_id and pro.major\_id=OBJECT\_ID('huo')

and col.id=OBJECT\_ID('huo')

select col.name,pro.value from sys.syscolumns col

left join sys.extended\_properties pro

on pro.minor\_id=col.colid and pro.major\_id=object\_id('huo')

where col.id=object\_id('huo')

配车：发送中 接单成功，则关闭时，刷新页面。

转换为枚举：Sex sexEnum = (Sex)1 ;//这样强制转换就可以了。0不用转换

不能使用：@item.ReceiptState==1?"是":"否"

@item.SendDate.Date---2017/12/1 0:00:00

prop() 方法设置或返回被选元素的属性和值。

当该方法用于返回属性值时，则返回第一个匹配元素的值。

当该方法用于设置属性值时，则为匹配元素集合设置一个或多个属性/值对。

注意：prop() 方法应该用于检索属性值，例如 DOM 属性（如 selectedIndex, tagName, nodeName, nodeType, ownerDocument, defaultChecked, 和 defaultSelected）。

提示：如需检索 HTML 属性，请使用 attr() 方法代替。

提示：如需移除属性，请使用 removeProp() 方法。

tr：点击事件，this为window

this指向所在函数的拥有者,f1的拥有者是谁?是window对象.

使用jquery获取父元素或父节点的方法

jq 判断父类标签的类型

lat维度 lng经度

JSON.parse(jsonstr); //可以将json字符串转换成json对象 JSON.parse(parent)[0].tagName

JSON.stringify(jsonobj); //可以将json对象转换成json字符串

## 11-29

### 强制把内容转换为整数:

一、（int）适合简单数据类型之间的转换，C#的默认整型是int32（不支持bool型）。

二、int.Parse（string sParameter）是个构造函数，参数类型只支持string类型，Parse就是把String类型转换成int,char,double....等,也就是\*.Parse(string) 括号中的一定要是string类型。

三、Convert.ToInt32（）适合将Object类型转换为int型；Convert可以提供多种类型的转换,也就是Convert.\*()括号中可以为很多种类型(包括string)。

四、Convert.ToInt32（）和int.Parse()对于空值（null）的处理不同，Convert.ToInt32(null)会返回0而不会产生任何异常，但int.Parse(null)则会产生异常。

五、对于数字的精确度不同：

（1）、Convert.ToInt32(double value) 如果 value 为两个整数中间的数字，则返回二者中的偶数，即 4.5 转换为 4，而 5.5 转换为 6。

（2）、int(4.6) = 4，int转化其他数值类型为int时没有四舍五入，强制转换(截取整数部分)。

### 获取便签的内容

text() - 设置或返回所选元素的文本内容

html() - 设置或返回所选元素的内容（包括 HTML 标记）

val() - 设置或返回表单字段的值

### jquery事件多次绑定

搜索绑定两次：使用$(“#btnSub”).click(function(){})，当这个js文件多次加载时，就会绑定多次click事件；解决方案：off(“click”).on(“click”,function(){})

### ~或/：表示根网址

LINQ to Entities 不识别方法“System.Object get\_Item(System.String)”，因此该方法无法转换为存储表达式：只能传入基本类型，或方法

Session["keyword"].ToString()

delUser(model) ：不能使用；delUser(22,lgcs18)--lgcs18未定义：字符串一定加引号，否则是变量。

window.location.reload();//强制刷新

## 11-28

head link、meta、script、style、title、base

title\meta\script\base\link

### <!DOCTYPE html>

＜meta http-equiv="参数" content="参数变量值"＞

http-equiv顾名思义，相当于http的文件头作用，它可以向浏览器传回一些有用的信息，以帮助正确和精确地显示网页内容，与之对应的属性值为content，content中的内容其实就是各个参数的变量值。

6、content-Type(显示字符集的设定)

<meta http-equiv="X-UA-Compatible" content="IE=edge" />：这是个是IE8的专用标记,用来指定IE8浏览器去模拟某个特定版本的IE浏览器的渲染方式（比如人见人烦的IE6），以此来解决部分兼容问题，例如模拟IE7的具体方式如下：< meta http-equiv = "X-UA-Compatible" content = "IE=EmulateIE7" />

<meta name="viewport" content="width=device-width, initial-scale=1" />：

layout viewport：

layout viewport 是网页的所有内容，他可以全部或者部分展示给用户。

visual viewport：

visual viewport 就是当前显示给用户内容的窗口，你可以拖动或者放大缩小网页。

width 设置layout viewport 的宽度，为一个正整数，使用字符串”width-device”表示设备宽度

initial-scale 设置页面的初始缩放值，为一个数字，可以带小数

minimum-scale 允许用户的最小缩放值，为一个数字，可以带小数

maximum-scale 允许用户的最大缩放值，为一个数字，可以带小数

height 设置layout viewport 的高度，这个属性对我们并不重要，很少使用

user-scalable 是否允许用户进行缩放，值为”no”或”yes”, no 代表不允许，yes代表允许

百度统计：http://tongji.baidu.com/open/api/more?p=guide\_overview

<script src="//hm.baidu.com/hm.js?d37dd9f0891a6066b9d878cccf7fa892"></script>

### position：

absolute：生成绝对定位的元素，相对于 static 定位以外的第一个父元素进行定位。元素的位置通过 "left", "top", "right" 以及 "bottom" 属性进行规定。

fixed：生成固定定位的元素，相对于浏览器窗口进行定位。元素的位置通过 "left", "top", "right" 以及 "bottom" 属性进行规定。

relative：生成相对定位的元素，相对于其正常位置进行定位。因此，"left:20" 会向元素的 LEFT 位置添加 20 像素。

static 默认值。没有定位，元素出现在正常的流中（忽略 top, bottom, left, right 或者 z-index 声明）。

inherit 规定应该从父元素继承 position 属性的值。

{0:X2}：表示转换为十六进制

strHex.Substring(8, 16)：位置，长度

### 登陆：

1、验证验证码

2、验证用户名、密码

3、根据是否保存：设置用户cookie

4、设置用户Session

5、设置权限Session

URL - 统一资源定位器

Web 浏览器通过 URL 从 web 服务器请求页面。

### URL 编码

URL 编码使用 "%" 其后跟随两位的十六进制数来替换非 ASCII 字符。

URL 不能包含空格。URL 编码通常使用 + 来替换空格。

JavaScript 中使用 encodeURI() 函数，ASP 中使用 Server.URLEncode() 函数。

JavaScript 函数将空格编码成 %20 。

### URI

URI，是uniform resource identifier，统一资源标识符，用来唯一的标识一个资源。

而URL是uniform resource locator，统一资源定位器，它是一种具体的URI，即URL可以用来标识一个资源，而且还指明了如何locate这个资源。

而URN，uniform resource name，统一资源命名，是通过名字来标识资源，比如mailto:java-net@java.sun.com。

也就是说，URI是以一种抽象的，高层次概念定义统一资源标识，而URL和URN则是具体的资源标识的方式。URL和URN都是一种URI。

总结一下：URL是一种具体的URI，它不仅唯一标识资源，而且还提供了定位该资源的信息。URI是一种语义上的抽象概念，可以是绝对的，也可以是相对的，而URL则必须提供足够的信息来定位，所以，是绝对的，而通常说的relative URL，则是针对另一个absolute URL，本质上还是绝对的。

URI抽象结构：scheme://host:port/path?query#fragment

DNS：（Domain Name Server域名服务器）DNS就是进行域名解析的服务器。域名的最终指向是IP。

### 打开一个网页实际上是进行如下步骤：

1.在浏览器里输入网址；

2.浏览器根据域名查找到对应的IP地址；

3.浏览器向web服务器发送HTTP请求；

4.服务器根据收到的请求返回响应给浏览器，显示内容。

### 公网、外网：

内、外网是相对于防火墙而言的，在防火墙内部叫做内网，反之就是外网。

在一定程度上外网等同于公网，内网等同于私网。

地址为如下3个区域就是处于私网：

1：10.\*.\*.\*

2：172.16.\*.\*至172.31.\*.\*

3：192.168.\*.\* （\*为0到255之间的任意数字）

以上3个区域外的就是处于公网之中了。

### 公网地址

公网地址是指在因特网上直接可达的地址，如果你有一个公网地址，那就意味着你不但能访问别人，还能被人访问。公网地址是架设各种服务器必不可少的条件。

### 私网地址

私网地址产生的一个原因是因为公网地址非常缺乏，大家不得不使用同一个公网IP地址上网，这就是共享上网的由来。同样，私网地址仍然可以被再次共享。

私有地址（Private address）属于非注册地址，专门为组织机构内部使用。

### cmd命令

#### 定位：

输入盘符 如C: 然后回车，相当于进了C盘(c盘一般进的是桌面目录)

输入 cd 目录名 然后回车(目录名是c盘中的一级目录名，也可为c盘中的目录路径)。如cd mm即是进入了mm文件夹。

#### 获取本机ip地址：

ipconfig

#### 获取本机服务器和地址：

nslookup；监测网络中DNS服务器是否能正确实现域名解析的命令行工具

D:\>nslookup

默认服务器: public1.114dns.com

Address: 114.114.114.114

> www.baidu.com 输入网址获取对应的ip

服务器: public1.114dns.com

Address: 114.114.114.114

非权威应答:

名称: www.a.shifen.com

Addresses: 111.13.100.92

111.13.100.91

Aliases: www.baidu.com

#### ping：

利用它可以检查网络是否能够连通，用好它可以很好地帮助我们分析判定网络故障。

Ping按回车即可看到详细说明。

+本机ip：

+Ping网关IP：

+Ping远程IP：可以通过域名得到ip

### 特殊符号

http://www.jb51.net/onlineread/htmlchar.htm

http://www.w3school.com.cn/html/html\_entities.asp

雪花：&#10052;

圈C：&copy; &#169;

圈R：&reg; &#174;

心（红桃）：&hearts; &#9829;

×：&hearts;

÷：&divide;

## 11-27

<http://blog.csdn.net/lhc1105/article/details/46285111>

<https://www.cnblogs.com/kissdodog/archive/2013/04/16/3025315.html>

问题：

设一个老师的生日为M月N日.把M告诉学生甲(即学生甲知道老师出生的月份),把N告诉学生乙(即学生乙知道老师出生的日).现有如下日期是公开的:3月4日,3月8日,3月10日,6月4日,6月7日,9月1日,9月10日,12月1日,12月2日,12月8日，其中有一个是老师的生日.

现有学生甲与学生乙的对话:

学生甲:如果我不知道的话,你一定不知道.

学生乙:本来我不知道的,现在我知道了.

学生甲:那我也知道了.

现在问:老师的生日是哪一天?

e10ad c3949ba59abbe56e057f20f883e

## 第四周

## 11-25

DbParameter：using System.Data.Common;

Expression：System.Linq.Expressions

DbQuery<T> ：System.Data.Entity.Infrastructure

JavaScriptSerializer：System.Web.Script.Serialization;

：Newtonsoft.Json;

：System.Runtime.Serialization.Json; 需要引用System.Runtime.Serialization

HttpContext：System.Web

Ioc：Inversion of Control，即“控制反转”

log4net是一个功能著名的开源日志记录组件。

分析器错误消息: 创建 spring/context 的配置节处理程序时出错: 未能加载文件或程序集“Spring.Web.Mvc5”或它的某一个依赖项。系统找不到指定的文件。

源错误:

行 10: <sectionGroup name="spring">

行 11: <!--Spring声明容器-->

行 12: <section name="context" type="Spring.Context.Support.MvcContextHandler, Spring.Web.Mvc5" />

设置xml生成操作为嵌入的资源

在应用程序配置文件中找不到名为“Entities”的连接字符串。

把App.Config中的Entities数据库连接字符串，复制到Web.config中

我们验证用户信息真实的情况下，我们直接通过Json输出了 Status="y"，但是我们并没有存储用户信息，今天我们来修改一下这个方法存储用户信息，我们通过两种方式，首先通过Session存储用户信息，然后把用户非关键信息加密后写入Cookies，当Session过期的时候，我们就通过Cookies重新获取用户并存储用户信息，这是大致的流程 很多朋友会说这也不是很安全，是的，没有绝对的安全，软件是没有绝对的100%安全的，我们只能在我们需求的情况下做到自己最好的安全。

https://mail.qq.com/cgi-bin/frame\_html?sid=anftojc8klZA5Wsx&r=199f1a2ba3226ed45057635d65708027

https://mail.qq.com/cgi-bin/frame\_html?sid=t6qsLbi0Z6O15mgy&r=f7c6d4c673d6aba0230319719ca6c4c3

https://mail.qq.com/cgi-bin/loginpage?s=session\_timeout&from=&r=22c2dedbec5bb0ec157af06e61b55b37

https://mail.qq.com/cgi-bin/frame\_html?sid=a3YcL7q2W5\_J7WM5&r=4516b12ab916c55676d8e8f1e5f8319a

out userModel)

对控制器类型“LoginController”的操作“Index”的当前请求在下列操作方法之间不明确:

类型 WTTC.Controllers.LoginController 的 System.Web.Mvc.ActionResult Index()

类型 WTTC.Controllers.LoginController 的 System.Web.Mvc.ActionResult Index(System.String, System.String, System.String)

bool b = Session["ValidateCode"].ToString() == imgVerify;Session["ValidateCode"]=null

需使用百度BD09坐标，如使用其他坐标（ WGS84、GCJ02）进行展示，需先将其他坐标转换为BD09，详细说明请参考坐标转换说明，请勿使用非官方的转换方法！！

### 百度地图接口

#### 创建

var map = new BMap.Map("container");

var point = new BMap.Point(116.404, 39.915);

进行初始化，BMap.Map.centerAndZoom()方法要求设置中心点坐标和地图级别。

map.centerAndZoom(point, 15);

panTo()方法将让地图平滑移动至新中心点，如果移动距离超过了当前地图区域大小，则地图会直跳到该点。

map.panTo(new BMap.Point(116.409, 39.918));

#### 覆盖物：信息窗口

var opts = {

width: 250, // 信息窗口宽度

height: 100, // 信息窗口高度

title: "" // 信息窗口标题

}

var infoWindow = new BMap.InfoWindow("<div> <p><span style='color:red'>车牌号</span>：123456</p> <button>发送订单</button></div>", opts); // 创建信息窗口对象

map.openInfoWindow(infoWindow, map.getCenter()); // 打开信息窗口

#### 设置覆盖物的文字标识：

var label = new BMap.Label("我是文字标注哦",{offset:new BMap.Size(20,-10)});

marker.setLabel(label);

获取覆盖物的信息：

marker.addEventListener("click",getAttr);

function getAttr(){

var p = marker.getPosition(); //获取marker的位置

alert("marker的位置是" + p.lng + "," + p.lat);

}

#### 设置点的新图标：

//创建小狐狸

var pt = new BMap.Point(116.417, 39.909);

var myIcon = new BMap.Icon("http://lbsyun.baidu.com/jsdemo/img/fox.gif", new BMap.Size(300,157));

var marker2 = new BMap.Marker(pt,{icon:myIcon}); // 创建标注

map.addOverlay(marker2); // 将标注添加到地图中

#### 文本：

var opts = {

position : point, // 指定文本标注所在的地理位置

offset : new BMap.Size(30, -30) //设置文本偏移量，文本右上角相对于坐标点

}

var label = new BMap.Label("欢迎使用百度地图，这是一个简单的文本标注哦~", opts); // 创建文本标注对象

label.setStyle({

color : "red",

fontSize : "12px",

height : "20px",

lineHeight : "20px",

fontFamily:"微软雅黑"

});

map.addOverlay(label);

### sql自增字段赋值

set IDENTITY\_INSERT [SYS\_MODULE] on//开启自增插入

set IDENTITY\_INSERT [SYS\_MODULE] off

### p是块级，span是行内

### button居中

两种办法：

1、外套一个长宽相等的div 水平居中 margin: 0 auto； 垂直居中：这个div的父元素加上 display：flex， align-items：center；

2、将这个button 加个样式 display：block；重复上面的俩个居中代码就好了。

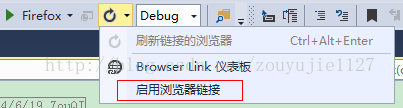
其实都是一个道理的，多敲俩下就知道了

## 11-24

## 11-23

如何干掉Browser Link功能？还我干净的页面代码。

你可以去掉选中 “Enable Browser Link” 选项， “ Browser Link” 功能就会被关闭



你也可以在web.config 文件中进行如下配置：

1. 设置“vs:EnableBrowserLink” 为“false ““. 则可以禁用Browser Link.

**<appSettings>**

**<add** key="vs:EnableBrowserLink" value="false"**/>**

**</appSettings>**

2、在web.config文件中设置debug= false，也可以禁用Browser Link

**<compilation** debug="true"**/>**

POST http://localhost:5300/api/Web/ALL 500 (Internal Server Error)

获取不到值：细心，多仔细看是否是id、id是否匹配、是否加#

var strContent = $("#txtContent").val();

var strTitle = $("txtTitle").val();

请求不到：

$.post("/api/Web/ALL", //地址方法错误

JSON.stringify({ title: strTitle, content: strContent }),

function (data) {

alert(data);

})

不能使用：http://localhost:5300/api/Web/ALL不存在 、

ApiController：get/post请求的用法

方法不能是静态方法；

参数不能是自动匹配；使用[FormData]或dynamic、Model类

public static string ALL(string title = "", string content = "", Dictionary<string, object> json = null)

dynamic需要指定post类型是application/json

[System.Web.Http.HttpPost]

public string ALL(dynamic title)

dynamic的属性，类型是dynamic {Newtonsoft.Json.Linq.JValue}

d.title→Convent.ToString(d.title)

$.ajax({

url: "../api/Web/AllMsg",

type: "POST",

contentType: 'application/json',

data: JSON.stringify({ title: strTitle, content: strContent }),

success: function (data) {

alert(data);

}

});

using 命名空间指令只能应用于命名空间；“WTTC.Comm.JPushHelper”是一个类型，而不是命名空间

页面嵌入C#：

@{WTTC.Comm.JPushHelper.AllMsg("标题", "内容", null); }

只在开始调用一次，返还结果；不会动态调用（像点击事件）：

按钮灰色：最好是根据判断不显示按钮；

或者去掉绑定事件、

<button id="btnPoint" disabled="disabled">

<button id="btnPointMsg" disabled>

## 11-22

#### 权限管理

为什么：一些功能一些用户不能用。方便管理

怎么用：树列表

网页bug：首页，用户虽然没有权限，但是都可以看到：运单、结算、

职位管理→管理员权限：选空数据，点击保存之后，提示正在保存，之后不能使用

没有权限，强制重新登陆，有点流氓。

职位权限：当前网点添加用户，只能使用当前网点设置的权限。

权限： 根据会员id得到用户角色，根据角色得到action

huiyuan→UserRole→RoleAction→Action

总部可以管理所有网点的会员、组织架构

下属网点 只能管理自己网点的下属会员

新增职位：Role

设置权限：/api/mulu/UpdateMulu //选空数据，点击保存之后，提示正在保存，之后不能使用

树列表：User/getMuluByRole

根据网点信息，获取：分 网点（可以开单、签收的）、分拨中心、分公司、总部（有清分的公司和无清分的公司） 、 专线、公司

专线的网点相当于分拨中心

#### 首页表头

设置上面进行反馈；帮助是新手教程

消息是用车消息

#### 专线和公司的区别：

组织架构：

* + - 专线--只有网点
    - 公司--机构列表、添加分公司、分拨中心、网点；右边有机构树状分布

#### 公司管理

##### 组织架构

查看：网点列表，自己网点和自营网点、下属网点 ZZSearchWangDian

操作：修改、停用 只能时自己网点

添加 分网点只能添加合作网点

##### 职位权限

可以设置所有的权限；分拨公司的权限

##### 员工管理

查看：只能查看本网点下的员工。

修改：不能修改隶属组织和登陆名

##### 修改密码

修改会员密码

##### 系统设置

运单打印设置与网点关联

打印机与会员关联、PrinterInfo

没有上级的则设修改公司的为不可见：data.parentid != 0 $(".com").hide();

总公司才可以看到：

公司信息

服务协议：Agreement

公司公告：gonggao

#### 物流公司

##### 组织架构

机构列表：网点是分拨中心、分拨中心是分公司

添加分公司、添加分拨中心：

添加网点：上门提货区域、送货上门区域、支持业务（收货、落货）

##### 线路运价

#### 多表连接

LEFT JOIN 最少的数据放在最前

一对多，则多为主；单复制

（会员第一次登陆）引导测试有问题：

## 11-21

#### 30天考试

ADO.NET的五个对象：

Connection、Commond、DataAdapter、DataReader、DataSet

DateTime可不可以使用null，nullable

三层：

BLL（Business Login Layer）业务逻辑层、

DAL（Database Access Layer）数据逻辑层、

UI（UserInterface）用户界面层、

Model

string str=null与str=''

一个是什么都没有，一个是[空字符串](https://www.baidu.com/s?wd=%E7%A9%BA%E5%AD%97%E7%AC%A6%E4%B8%B2&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4mHf4mWmsuW6vP10znhu-0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHcsPjDLPHfs)，区别大了

你定义了一个变量的引用str

string str=null 把这个引用指向了一个null，没有地址没有值的地方。

string str="" 把这个引用指向了一个地址，地址里面存的是空的字符。

Parameter ：

SqlParameter para2 = new SqlParameter("@OutVal", SqlDbType.VarChar, 10);

para2.Direction = ParameterDirection.Output;

testCommand.Parameters.Add(para2);

#### 公司管理

公司信息：修改公司信息 Company/Edit

公司协议：Company/Agreement

修改：先判断是否有数据，没有添加；有则修改。

id 序号

companyid 公司编号

AgreementContent 协议内容

#### 获取存储过程数据：

select text from syscomments where id=object\_id('SearchWangDian1')

连接为一个字符串，没有换行

sp\_helptext SearchWangDian1

有换行

#### 网点：修改/停用

可以操作：当前网点是总部（可以修改所有）、是合作网点、是登陆网点

不能：自营、网点类型为空

筛选：返回总数

同一公司、未删除；下属网点或自营或本网点

可以看到所有自营、下属、本网点；只能操作下级网点、合作网点、本网点

#### 员工管理： Searchhuiyuan2 （网点）

可以修改本网点所有员工

bll\_user.getuserListByPage(0, "", wangdianid,companyid, 1, 20)

Searchhuiyuan2 idStr,username,wangdianid,copmpanyid,intPageNo,intPageSize

先根据网点、公司筛选用户；再根据id、登陆名筛选 ；

职位权限是跟用户关联的；

## 11-20

#### 登陆：

会员表：登陆时，主要判断 是否删除、禁用；登陆成功自动修改登陆失败次数、上次登录时间、当前登录时间

是否删除 isDelete

是否禁用 state

公司引导 skipGuid 1

登陆失败次数 failCount 最后登录失败时间 faillogintime //当前程序没有用

登陆成功：则设失败次数0、登陆时间、登陆次数加1 logincount

lastlogintimt 最后登陆时间、lastlogintime 上次登陆时间

是否离职 lev 离职时间 levtime

真实姓名 truename

注册时间 regtime

企业 companyid、网点 wagndianid、添加人 parentid

主题样式 cssType

网点：//登陆判断网点是否未启用、删除

添加人 ownid，0是管理员 网点的上级网点为parentid

状态 state 0未启用 1启用 2删除

级别 lever

判断网点类型 branch\_com 0网点 1分公司 2分拨中心

wd\_Type 直营、加盟、合作

是否可以发货（开单） wd\_fahuo、落货（签收） wd\_shouhuo

到货、发货查询电话 arriveHuoTel、fahuoTel

获取权限

通过用户id 获取角色Role

通过角色 获取角色功能RoleAction

通过角色功能 获取功能Action

角色表 Role

角色名、添加时间、添加人、所属网点、

状态：是网点、还是用户

公司id

是管理员，还是普通用户：isManage

角色权限表 RoleAction

角色id、功能id

菜单表 Action

名称、地址、上级id、等级

排序、主栏目

状态：显示在左侧列表、不显示、未启用

管理员：adminaction

分公司权限：BranchCom\_Action

分拨中心：FBCenter\_Action

网点：WD\_Action

#### 首页 Main/Index \_MyLayout3

主页信息：个人信息、网点信息

Html.Action()//获取信息

网点信息：联系人、联系电话、已累计承接运单数、已收货款、拥有线路

待处理运单：运单装车、车辆到站、车辆出站、待签收

加跳转地址/运单数量

回单管理：发站开单、到货签收、到站待寄送、

运单列表、发站收取、客户收取

结算管理

发货收款 最近一月未结算/已结算的 费用/跳转地址

到货收款

货款结算

\_MyLayout3模板

表头

新手教程

设置

消息

退出

表左列

给一级菜单加第一个下属列表地址

\_MyLayout模板

#### 公司注册：RegCompany/reg

添加公司

添加网点

添加会员

添加角色 系统管理员

添加用户角色

添加角色 公司总部

添加角色功能 RoleAction

登陆加载的组织架构

登陆没有添加记忆功能

## 第三周

### 11-17

zTree中的url不是直接写在标签上

实现滚动条：margin-left+1，定时执行

错误：发送邮件，点击添加 未发送成功.txt 文件，发送删除；则原来输入的地方，发送成功后，不能删除不存在的文件

新代码与老需求不兼容

#### 加载树权限：

网点等于0 ，加载所有

1、判断网点是不是总公司 ：branch\_com null总公司 0 网点 1分公司 2分拨中心

1.1公司 判断公司是不是清分，不是的话从adminaction == 1中，去掉 130 清分对账 /QingFen/Index

1.2其他：分别根据 branch\_com、wd\_action、fbcenter\_action

1.2.1 网点 判断是否有发货（发站开单）、收货 （"客户签收" "已签收订单"）

2、判断公司类型 ComPanyType 0:普通公司 1:物流专线公司

2.1 1专线公司，换url list.page = "/SpecialWanngDian/Splaceorder";

"发站开单" "/SpecialWanngDian/Splaceorder";

"开单查询" "/SpecialWanngDian/Splaceorder";

"运单列表" "我开的单" "/AllOrsder/IndexByZZ"

"组织架构" "/SpecialWanngDian/WangDianZZ/Index";

"车辆到站" "/SpecialWanngDian/ArrivalZZ/Index";

"发站外包" "/SpecialWanngDian/waibaoZZ/index";

"已外包运单" "/SpecialWanngDian//waibaoZZ/OutOrderlist";

"运单装车" "/SpecialWanngDian/loadingZZ/Index";

"车辆出站" "/SpecialWanngDian/DepartureZZ/Index";

"员工管理" "/SpecialWanngDian/UserMangerZZ/userList";

"网点收益" "/SpecialWanngDian/xiaoshouZZ/Index";

"收益详情" "/SpecialWanngDian/xiaoshouZZ/detial";

"网点结算" "/SpecialWanngDian/caiwuZZ/Index";

"缴纳费用统计" "/SpecialWanngDian/caiwuZZ/DotRank";

"公司利润" "/SpecialWanngDian/ComprofitZZ/Index";

"下级网点战报" "/SpecialWanngDian/zhanbaoZZ/Index";

"打款记录" "/SpecialWanngDian/dakuanZZ/Index";

"营业额报表" "/SpecialWanngDian/OrderTogetherZZ/Index";

"网点营业额报表" "/SpecialWanngDian/WDOrderTogetherZZ/Index";

"发车量统计报表" "/SpecialWanngDian/CarTogatherZZ/Index";

"网点发车量统计" "/SpecialWanngDian/WDCarTogatherZZ/Index";

3、筛选所有的一级树，添加下属 0-3

133 发站运输费用核销 /SpecialWanngDian/fzysfyhxZZ/Index 75 6

134 到站运输费用核销 /SpecialWanngDian/dzysfyhxZZ/Index 75 7

#### 添加员工：

1、添加会员

调用UserManger中的UpdateHuiyuan，传入id，roleid：不大于0添加；大于0，修改

hy\_model.cssType = 8;

对发送的请求数据进行过滤

parentid是添加员工的id

2、添加用户角色

添加角色：

state 0：用户角色，1：网点角色

isMange 1：公司总部或网点管理员，0:普通

网点属于分拨中心

分公司属于 分公司/总部

分拨中心属于 分公司/总部

修改不能修改隶属公司、不能修改网点的上门提货区域、送货

//总部、分公司不能收货，不能落货 //分拨中心不能收货，可以落货

一共有102个菜单：state 1-29(附属页，) 0-73

网点、分拨中心、分公司都没有的：

公司信息

公司公告

服务协议

公司利润

网点收益

网点结算

网点利润

线路运价

客户下单列表

GPS登陆

仓库管理

收益详情

缴纳费用统计

清分对账

分公司没有：已签单订单

网点没有：打款记录

#### 添加网点：WangDianController/addCom

478-479 1、添加网点（wangdian）

categoory = -1

state = 1

branch\_com=0

ownerid=开单人id

lever=3

wd\_type=直营

2、添加角色(Role) 4个角色：网点系统管理员|开单员|财务人员|运营人员

state=0 0：用户角色，1：网点角色

isMange=1 1：公司总部或网点管理员，0:普通

全部调用4次

2.1、添加角色对应的 角色菜单（RoleAction）1 categoory == -1---FBCenter\_Action == 1 p.WD\_Action == 1

1---p.BranchCom\_Action == 1

选择 支持业务-收货：发站开单

选择 支持业务-落货：客户签收

4个 2.2、选择了上门提货区域：添加提货表 tihuo flag 0:表示提货 1：表示送货

4个 2.3、选择了送货上门区域：添加提货表 tihuo

提货表：省市县镇 公司、网点 重货、轻货、最低、免费价格 添加时间

11-16

组织架构：exec SearchWangDian1 各个网点；总公司可以看到所有网点，其他网点只能看到本网点

wangdian：ownerid开单人 companyid公司 state状态，启动/未/删除

--categoory网点类型：-1分拨、0分、1网点

wd\_type：网点类型：直营、加盟、合作

branch\_com：null总公司 0 网点 2 分拨中心 1分公司

branch\_com == null || (model\_wd.parentid == 0 && model\_wd.branch\_com == 0))//总公司

#### 添加分公司：Permission/WangDianController/addCom

476

1、添加网点（wangdian）categoory = -1

state = 1 branch\_com=1 ownerid=开单人id lever=2

2、添加角色(Role) 4个角色：分公司系统管理员|开单员|财务人员|运营人员

state=0 0：用户角色，1：网点角色

isMange=1 1：公司总部或网点管理员，0:普通

3、添加角色对应的 角色菜单（RoleAction） categoory == -1---FBCenter\_Action == 1 p.WD\_Action == 1

1---p.BranchCom\_Action == 1

#### 添加分拨中心：WangDianController/addCom

477 1、添加网点（wangdian）categoory = -1

state = 1 branch\_com=2 ownerid=开单人id lever=2

2、添加角色(Role) 4个角色：分拨中心系统管理员|开单员|财务人员|运营人员

state=0 0：用户角色，1：网点角色

isMange=1 1：公司总部或网点管理员，0:普通

3、添加角色对应的 角色菜单（RoleAction）1 categoory == -1---FBCenter\_Action == 1 p.WD\_Action == 1

1---p.BranchCom\_Action == 1

### 11-16

#### 添加分公司、分拨中心、网点、

查看添加公司、网点、分拨中心 用户的操作

Role：对应网点表、公司表、上级网点

UserRole：用户权限，对应会员表和权限表

RoleAction：对应权限表和菜单表

Action：菜单

#### 修改发送邮件系统：

问题：发送完成的邮件从文件中删除。

使用File类进行操作。使用FileStream，FileMode.Truncate会把文件清空，但往里面写东西，之后，还是空的。

using (FileStream stream = new FileStream("未发送成功的邮件.txt", FileMode.Truncate, FileAccess.ReadWrite, FileShare.Read))

{

byte[] bytes = new byte[stream.Length];

stream.Read(bytes, 0, (int)stream.Length);

string str = Encoding.Default.GetString(bytes);

bytes = Encoding.Default.GetBytes(str.Replace(o.ToString() + ',', ""));

stream.Write(bytes, 0, bytes.Length);

}//每次都是清空数据

string str=File.ReadAllText("未发送成功的邮件.txt");

string strNew=str.Replace(o.ToString() , "");

File.WriteAllText("未发送成功的邮件.txt", strNew);

#### 查找用户点击 已签收运单，跳转到登陆页的问题：

当加载母版页时，会查看是否有这个权限，没有则调换到login（不区分大小写）：--快速监视-不能监视有lambda的数据

\_MyLayout.cshtml 296

var com = '@Common.StringHelper.getColumn(ViewContext.RouteData.Values["controller"].ToString(),ViewContext.RouteData.Values["action"].ToString(), ((List<Model.UserAction>)Session["userMulu"]).OrderBy(c=>c.lev).ToList())';

if (com == "-1") {

loadUrl("/login");

}

#### 缓存/文件操作：看了一部分，还没有具体测试

HashTable：System.Collections

MemoryCache：.NET4.0引入，using System.Runtime.Caching;

expiration呼气、截至、满期

MemoryCache.Default.Add(item, policy);

MemoryCache.Default[key]

CacheItemPolicy：缓存项政策

AbsoluteExpiration：Gets or sets a value that indicates whether a cache entry should be evicted after a specified duration.获取或设置一个值，该值指示在指定的持续时间之后是否应取消缓存条目。

SlidingExpiration：Gets or sets a value that indicates whether a cache entry should be evicted if it has not been accessed in a given span of time.获取或设置一个值，该值指示在给定时间内未访问缓存条目是否应该被逐出。

Priority：Gets or sets a priority setting that is used to determine whether to evict a cache entry.获取或设置一个优先级设置，用于确定是否将高速缓存条目。

Cache：System.Web.Caching.Cache

HttpRunTime.Cache和HttpContext.Current.Cache都是该类的实例

perform a request or hit F5 to record the reload。执行请求或单击F5来记录重新加载。

### 11-15

修改发送邮件系统：问题--获取数据时，需要把字符串前后的空格删除

查看用户权限

UserRole：用户权限

Role：权限

RoleAction：权限

GIT分支

### 11-14

修改发货人、发货电话enter键选取问题，同步到物流公司、物流专线

测试树状图

Linq Queryable Join

public static IQueryable<TResult> Join<TOuter, TInner, TKey, TResult>(

this IQueryable<TOuter> outer, //要联合的外部数据

IEnumerable<TInner> inner, //要联合的内部数据

Expression<Func<TOuter, TKey>> outerKeySelector,

Expression<Func<TInner, TKey>> innerKeySelector,

Expression<Func<TOuter, TInner, TResult>> resultSelector

);

合并数据 left join，中间两个参数两个关联的键，最后一个是返回的类型，不能是数据库Model

listAction = dal\_userRolr.LoadEntities(c => c.userid == userId)

.Join(

dal\_roleAction.LoadEntities(c => true),

z => z.roleid,

t => t.roleid,

(z, t) => new { actionid = t.actionid })

.Join(

dal\_action.LoadEntities(y => true),

r => r.actionid,

p => p.id,

(r, p) => new Model.UserAction() {

addtime = p.addtime, id = p.id, lev = p.level, name = p.name,

parentid = p.parentid, remark = p.remark, sort = p.sort, page = p.url,

columns = p.columns, state = p.state })

.OrderBy(z => z.sort).ToList();

Entity FrameWork

DataBase-First：添加ADO.NET实体数据模型

查看映射关系、与数据库同步：在设计页面点击表或空白处

创建表、视图、存储过程：空白处右击-从数据库更新模型--添加、刷新、删除

三大模型：

概念模型：Conceptual Model 可以被应用程序直接使用的类。 CSDL描述（Conceptual Schema Definition Language）

存储模型：Storage Model与数据库系统直接对应的类。 SSDL描述（Storage Schema Definition Language）

概念-存储模型映射：Conceptual-Storage Mapping 解决概念模型与存储模型类对应的问题。 MSL(Mapping Specification Language)

参初始化部件“Microsoft.CodeAnalysis.Diagnostics.DiagnosticAnalyzerService”时引发了异常。

解决方法：

win+r，输入regedit，在这个目录下HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\FipsAlgorithmPolicy，把Enabled的值改为0就可以了

### 11-13

排序：

插入排序：把当前值与前面的值比较，小于的话，把当前值与前面所有的值比较，插入到大于与小于之间；等于则插在后面

希尔排序：

选择排序：

简单选择排序：从数组中选出最小（大）的值，排在第一位；然后从剩下的选择一个最小（大）的值排在第二位....

二元选择排序：选出最大和最小值

冒泡排序：

ztree

初始化：$.fn.zTree.init(jQuery对象：树在哪里显示，设置，节点)

节点：name,open,children[{name,children[{name,children}]},]

或id、pId（父id)、name需要setting中data{simpleData:{enable:true }}

isParent：父节点（显示，默认是有子类为父节点，否则为子节点）

icon:".png"节点图标 iconOpen/iconClose

iconSkin:自定义图标

.ztree li span.button.diy02\_ico\_open{...}

.ztree li span.button.diy02\_ico\_close{...}

.ztree li span.button.diy03\_ico\_docu{...}

自定义字体：

font:{'background-color':'black', 'color':'white'}

name:"<span style='color:blue;margin-right:0px;'>view</span>.<span style='color:red;margin-right:0px;'>nameIsHTML</span>"

view:{fontCss:getFont, nameIsHTML:true//name是html}

function getFont(treeId, node) {

return node.font ? node.font : {};

}

超链接：

url、target、click

设置：

view: { showLine: false} 展示线 -+线

showIcon:false 不展示节点前面的图标

data{key:{title:"t"}}节点提示信息：view.showTitle:true可以（默认是true）

callback：

beforeCollapse Function(treeId, treeNode). 父节点折叠之前的事件回调函数，并且根据返回值确定是否允许折叠操作

实现单击：view{dblClickExpand:false}

callback{onClick:onClick}

function onClick(e, treeId, treeNode) {

var ztree = $.fn.zTree.getZTreeObj("treeDemo");

ztree.expandNode(treeNode);

}

尚未对布局页“~/Views/Shared/\_MyLayout.cshtml”调用“RenderBody”方法。

Unexpected token i in JSON at position 2：json格式错误

json中键或值都必须用双引号引起来

## 第二周

### 11-12 星期天

聚会、睡觉、

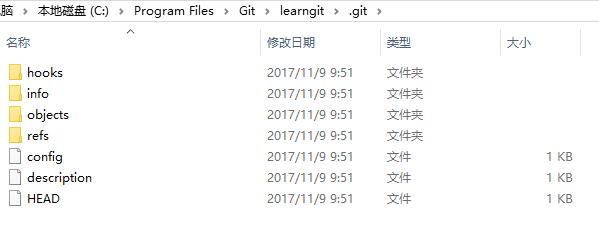
### 11-11

讨论WTMS界面：开单管理、车辆管理、财务管理、统计报表、公司管理

### 11-10

查看WTMS程序

### 11-9 GIT

流程：

本地：设置登录名、邮件地址--初始化文件为git--添加提交文件--提交

远程：克隆远程数据--添加远程连接--add添加文件--commit提交--push到服务器

查看状态status、查看日志log、操作记录reflog、对比不同内容diff、创建目录、定位目录、回退reset、删除文件、撤销、

打断点，VS崩溃，是安装文件与VS系统有冲突。删除最近安装的文件

GIT

1、登录名

$ git config --global user.name "Your Name"//tow dashs

$ git config --global user.email "email@example.com"

git config命令的--global参数，用了这个参数，表示你这台机器上所有的Git仓库都会使用这个配置

2、版本库 repository

创建空目录

2.1 选一个合适的地方

$ mkdir learngit //创建目录 mkdir：make directories

$ cd learngit //定位到目录，支持相对目录。./.. cd：change directory

$ pwd //获取当前目录，相对路径 pwd：Print working directory

/Users/michael/learngit

2.2 通过git init把这个目录变成git可以管理的仓库

$ git init //输入回车自动出现下面的

Initialized empty Git repository in C:/Program Files/Git/learngit/.git/

--ls -ah：ls是linux下的list，列出目录下的文件

2.3 把文件添加到版本库

所有的版本控制系统，都只能察觉到txt文件的变动，而不能察觉到图片、视频的变动，只能知道大小改变，而不知道具体变动

word是用二进制格式的，因此不能跟踪word文件的变动。

而文本有编码，建议使用UTF-8，这样利于控制，解决冲突

2.3.1 把文件放到仓库目录下

$ git add readme.txt

2.3.2 把文件提交到仓库

$ git commit -m "wrote a readme file" //-m之后是本次提交的说明

结果：

[master (root-commit) 31b92dd] wrote a readme file

1 file changed, 2 insertions(+)//一个文件被修改，2行插入

create mode 100644 readme.txt

使用commit，则第一步，设置用户名和邮箱是必须的

commit提交add（2.3.1）中所有的文件

3、查看修改状态

查看状态：git status

结果

On branch master

Changes not staged for commit://修改而没有提交

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

modified: readme.txt

no changes added to commit (use "git add" and/or "git commit -a")

查看修改内容：git diff readme.txt

diff --git a/readme.txt b/readme.txt

index d8036c1..0e9af4f 100644

--- a/readme.txt

+++ b/readme.txt

@@ -1,2 +1,2 @@

-Git is a version control system.

+Git is a distributed version control system.

Git is free software.

\ No newline at end of file

格式是Unix通用的diff格式

添加之后，再查看状态

git add readme.txt

git status

结果：

On branch master

Changes to be committed://将被提交

(use "git reset HEAD <file>..." to unstage)

modified: readme.txt//改动的文件

———————————时光穿梭—————————————

4、回退

查看日志

git log //不能使用大写

返回：

$ git log

commit ad73b1197b323a1adf4530fa1cc278506cc7894e

Author: DGZ <2395247901@qq.com>

Date: Thu Nov 9 10:30:56 2017 +0800

append GPL

commit 8aa3df6662386f79a5f8f1aa711a0196e9af60aa

Author: DGZ <2395247901@qq.com>

Date: Thu Nov 9 10:26:32 2017 +0800

add distributed

commit 31b92dd97e5bd6caa321189777459cc05a010222

Author: DGZ <2395247901@qq.com>

Date: Thu Nov 9 10:08:39 2017 +0800

wrote a readme file

简便：$ git log --pretty=oneline

结果：

ad73b1197b323a1adf4530fa1cc278506cc7894e append GPL

8aa3df6662386f79a5f8f1aa711a0196e9af60aa add distributed

31b92dd97e5bd6caa321189777459cc05a010222 wrote a readme file

回退：HEAD表示当前版本，HEAD^表示上一个版本，HEAD~100表示100

$ git reset --hard HEAD^

结果：

HEAD is now at 8aa3df6 add distributed

获取当前文件

$ cat readme.txt

结果：

Git is a distributed version control system.

Git is free software.

回退到上一版本：

$ git reset --hard ad73b1197b3//只需要前几位就行

HEAD is now at ad73b11 append GPL

有一个head指针，指向版本

获取操作记录：

$ git reflog //需要先cd到文件夹中

结果：

版本号 操作

ad73b11 HEAD@{0}: reset: moving to ad73b1197b3

8aa3df6 HEAD@{1}: reset: moving to HEAD^

ad73b11 HEAD@{2}: commit: append GPL

8aa3df6 HEAD@{3}: commit: add distributed

31b92dd HEAD@{4}: commit (initial): wrote a readme file

5、暂存区

工作区：电脑中看到的文件，如learngit

版本库repository：工作区中隐藏的.git

版本库中有暂存区stage或index，还有自动创建的分支master，以及指向master的一个指针HEAD

git add把文件添加到暂存区

git commit是将暂存区中的内容提交到当前分支

6、管理修改

git add 之后提交，则中间发生的修改不会提交上去

$ git diff HEAD -- readme.txt//版本库和HEAD(当前版本)比较

结果

diff --git a/readme.txt b/readme.txt

index 43b377b..9c3b6c2 100644

--- a/readme.txt

+++ b/readme.txt

@@ -1,4 +1,4 @@

Git is a distributed version control system.

Git is free software distributed under the GPL.

Git has a mutable index called stage.

-Git tracks changes.

\ No newline at end of file

+Git tracks changes of files.

\ No newline at end of file

7、撤销

checkout：

git checkout -- readme.txt //--很重要，没有--，就变成了“切换到另一个分支”的命令

把readme.txt文件在工作区的修改全部撤销，这里有两种情况：

一种是readme.txt自修改后还没有被放到暂存区，现在，撤销修改就回到和版本库一模一样的状态；

一种是readme.txt已经添加到暂存区后，又作了修改，现在，撤销修改就回到添加到暂存区后的状态。

总之，就是让这个文件回到最近一次git commit或git add时的状态。

reset：既可以回退版本，也可以把暂存区的修改回退到工作区。

$ git reset HEAD readme.txt

返回：

Unstaged changes after reset:

M readme.txt

8、删除文件

rm file //rm：remove 从文件夹中删除

git rm file //从版本库中删除

git commit

误删，返回：git checkout -- file //用版本库中的版本替换工作区的版本

HEAD严格来说不是指向提交，而是指向master，master才是指向提交的，所以，HEAD指向的就是当前分支

当我们创建新的分支，例如dev时，Git新建了一个指针叫dev，指向master相同的提交，再把HEAD指向dev，就表示当前分支在dev上：

分支

创建分支：

git checkout -b dev

结果

Switched to a new branch 'dev'

git checkout命令加上-b参数表示创建并切换，相当于以下两条命令：

$ git branch dev //创建分支

$ git checkout dev //切换分支

查看分支

$ git branch

结果

\* dev--当前分支

master--其它分支

提交分支

$ git commit -m 'branch text'

结果

[dev d98d42a] branch text

1 file changed, 1 insertion(+)

切换到主分支，则文件内容没有了；

合并分支

$ git merge dev

结果

Updating 5272231..d98d42a

Fast-forward

readme.txt | 1 +

1 file changed, 1 insertion(+)

删除分支

$ git branch -d dev

结果

Deleted branch dev (was d98d42a).

分支冲突

使用合并则变为

<<<<<<< HEAD

Create new branch is quick AND simple

=======

Create new branch is quick AND simple

>>>>>>> feature1

修改，之后提交（不用合并）

查看分支合并结果

$ git log --graph --pretty=oneline --abbrev-commit

结果

\* 7aced15 (HEAD -> master) conflict fixed

|\

| \* 0635f31 (feature1) AND simple

\* | dd7d46e & simple

|/

\* d98d42a branch text

\* 5272231 第一次提交

删除分支

存储

创建

git stash

查看

git stash list

恢复

git stash apply stash@{} 恢复到最近的一次

删除

git stash drop stash@{}

git log退出方法

英文状态下按Q

#### 远程仓库：

##### 先有本地文件，后有远程库

1、（先进入项目文件夹）通过命令 git init 把这个目录变成git可以管理的仓库

git init

2、把文件添加到版本库中，使用命令 git add .添加到暂存区里面去，不要忘记后面的小数点“.”，意为添加文件夹下的所有文件

git add .

3、用命令 git commit告诉Git，把文件提交到仓库。引号内为提交说明

git commit -m 'first commit'

4、关联到远程库

git remote add origin 你的远程库地址

如：

git remote add origin https://github.com/cade8800/ionic-demo.git

4.1取消本地目录下关联的远程库：

git remote remove origin

5、获取远程库与本地同步合并（如果远程库不为空必须做这一步，否则后面的提交会失败）

git pull --rebase origin master

6、把本地库的内容推送到远程，使用 git push命令，实际上是把当前分支master推送到远程。执行此命令后会要求输入用户名、密码，验证通过后即开始上传。

git push -u origin master//第一次加u

##### 先有远程库后有远程文件：克隆

$ git clone git@github.com:michaelliao/gitskills.git

克隆与pull的区别：

* 克隆直接从远程中拉所有的文件：拉仓库到本地；
* pull则只是拉提交的文件，到自己的仓库中。

### 11-8 jQuery UI

1、for in ：for( i in event){str+=(i+" "+event[i]+"\n")}

event属性：target/srcElement目标元素、type事件类型、keyCode

var d=$('#dialog').data('ui-dialog');

var content='';

for(var item in d)

{

content+=(item+' '+d[item]+'\r\n');

};

alert(content);

2、加载页面之后执行

window.ready(function(){ })

document.onload(function(){ })

$(function(){ })

$(document).ready( function(){} )

jQuery UI

1、初始化： 在一个元素上面调用一个插件，就初始化一个部件

$( "#elem" ).progressbar();默认选项

改默认值：$( "#elem" ).progressbar({ value: 20 });

2、初始化之后可以调用一个方法，传入一个参数：$( "#elem" ).progressbar( "value", 40 );

支持链式调用

3、公共方法：

option：修改当前值。

修改：$( "#elem" ).progressbar( "option", "value", 30 );

获取：$( "#elem" ).progressbar( "option", "value" );

多个：$( "#elem" ).progressbar( "option", { value: 100, disabled: true});

disable：显示为禁用

$( "#elem" ).progressbar( "disable" );等同于设置disable为true

enable 方法是 disable 方法的对立面。

$( "#elem" ).progressbar( "enable" );

destroy 销毁它，生命周期的终止。

或者通过 .remove()，也可以通过 .html() 或 .empty() 来移除元素，小部件会自动销毁。

widget 一些小部件生成包装器元素，或与原始元素断开连接的元素。在下面的实例中，widget 将返回生成的元素。在进度条（progressbar）实例中，没有生成的包装器，widget 方法返回原始的元素。

4、事件：用于在状态改变时通知您

$( "#elem" ).bind( "progressbarchange", function() {

alert( "The value has changed!" );

});

$( "#elem" ).progressbar({

change: function() {

alert( "The value has changed!" );

}

});

1、创建小部件（扩展小部件）：widget--$.widget(命名空间.部件名，父部件的构造函数，扩展的对象【属性】)

创建：

$.widget( "custom.superDialog", $.ui.dialog, {} );

$.widget( "custom.superDialog" ，

,$.ui.dialog, {

red: function() {

this.element.css( "color", "red" );//改变小部件的文本颜色

}

});

调用：

$( "<div>I am red</div>" ).dialog().superDialog( "red" );

$( "<div>I am red</div>" ).dialog();//显示对话框，$()里的是对话框中显示的文字，必须是html；或选择器

2、覆盖父类、原型同名方法，使用这个custom.superDialog，会重写前面有的custom.superDialog

$.widget( "custom.superDialog", $.ui.dialog, {

open: function() {

console.log( "open" );

}

});

3、调用父类的同名方法：\_super() 和 \_superApply() 在父部件中调用了同样的方法

等同于Function.prototype.call() 和 Function.prototype.apply() 方法

因此，\_super() 接受一个参数列表，\_superApply() 接受一个数组作为参数。

$.widget( "custom.superDialog", $.ui.dialog, {

open: function() {

console.log( "open" );

// Invoke the parent widget's open().

return this.\_super();

}

});

4、重定义部件：传递 $.widget() 一个已有的小部件名称和构造函数即可。

$.widget( "ui.dialog", $.ui.dialog, {

open: function() {

console.log( "open" );

return this.\_super();

}

});

5、多态性：父部件不能调用子部件上的方法

var dialog = $( "<div>" ).superDialog();

// This works.

dialog.superDialog( "close" );

// This doesn't.

dialog.dialog( "close" );

6、更改实例的方法

var dialogInstance = $( "<div>" ).dialog().data( "ui-dialog" );

// Override the close() method for this dialog

dialogInstance.close = function() {

console.log( "close" );

};

// Create a second dialog

$( "<div>" ).dialog();

// Select both dialogs and call close() on each of them.

// "close" will only be logged once.

$( ":data(ui-dialog)" ).dialog( "close" );

7、 $( "#dialog" ).dialog({ autoOpen: false });//设置默认是false，则dialog("open")才能打开；且把#dialog元素隐藏

$( "#opener" ).click(function() {

$( "#dialog" ).dialog();//默认打开，设置了autoOpen：false，则需要显示调用open

});

插件方法调用

1、插件调用：把方法名称以字符串，传递到插件调用方法中；

如果需要额外的参数，则在后面再写一个参数

$( ".selector" ).dialog( "option", "height" );

2、获取实例：每个小部件的每个实例都是使用 jQuery.data() 存储在元素上。

var dialog = $( ".selector" ).data( "ui-dialog" );//获取实例部件

dialog.close();//调用实例部件上的方法

在 jQuery UI 1.11 中，使用instance获取实例

$( ".selector" ).dialog( "instance" ).close();

3、返回类型

插件调用的方法返回一个jQuery对象，

var dialog = $( ".selector" ).dialog();

dialog.dialog( "close" );

而实例上调用，返回undefined

dialog.data( "ui-dialog" ).close();

例外的是，返回小部件相关信息的那些方法

$( ".selector" ).dialog( "isOpen" )//返回true/false

$.widget( "custom.progressbar", {

\_create: function() {

var progress = this.options.value + "%";

this.element

.addClass( "progressbar" )

.text( progress );

}

});

1、命名空间只能有一层。如very.custom.progressbar，就不行

2、this.element是一个包含一个元素的jQuery对象；

3、this.options是一个包括所有插件选项的键名/键值的哈希

当我们调用 jQuery.widget()，它通过给 jQuery.fn（用于创建标准插件的系统） 添加函数来扩展 jQuery。所添加的函数名称是基于您传给 jQuery.widget() 的名称，不带命名空间 - "progressbar"。

4、回调：\_trigger() 方法有三个参数：回调名称，一个启动回调的 jQuery 事件对象，以及一个与事件相关的数据哈希。

5、您也可以在不遍历插件方法的情况下创建一个实例，通过选项和元素直接调用构造函数即可：

var bar = $.custom.progressbar( {}, $( "<div></div>" ).appendTo( "body") );

6、插件有构造函数和原型的最大好处是易于扩展插件。通过添加或修改插件原型上的方法，我们可以修改插件所有实例的行为。

7、ui 命名空间被官方的 jQuery UI 插件保留。

8、\_destroy() 取消引用。通过 .destroy() 方法被调用的，.destroy() 方法是在插件实例绑定的元素从 DOM 上移除时被自动调用的，所以这可被用于垃圾回收。基本的 .destroy() 方法也处理一些常用的清理操作，比如从小部件的 DOM 元素上移除实例引用，从元素上解除绑定小部件命名空间中的所有事件，解除绑定所有使用 \_bind() 添加的事件。

API

1、addClass方法：

addClass(className[,duration][,easing][,complete])

或addClass(className [,options]) options:dutation,easing,complete,children,queue,

2、easings:linear不变的速度；swing jQuery默认

3、百叶窗特效：blind

参数：direction up/down/left/right/vertical/horizontal

$( "#toggle" ).toggle( "blind",{direction:"left"});

下方元素会填充当前位置。

4、反弹：bounce 向上反弹

参数：distance 最大反弹距离，像素为单位，默认20

times 反弹的次数，默认加半个。默认为5

$( "#toggle" ).toggle( "bounce", { times:1 }, "slow" );

5、剪辑特效：clip

参数：direction vertical默认上下压缩/horizontal两边压缩

$( "#toggle" ).toggle( "clip" ,{direction:"horizontal"});

6、降落特效：drop 单个方向滑入

参数：direction up、down、left默认、right。

7、动画特效：effect

effect(effect [,options] [,duration] [,complete])

effect String 一个字符串，指示要使用哪一种特效。

options Object 特效具体的设置和 easing。

duration Number 或 String 一个字符串或一个数字，指定动画将运行多久。 400

complete Function() 一旦动画完成时要调用的函数。

.effect( options ) options:effect,duration,easing,complete,queue,

8、爆炸特效：explode

参数：pieces 爆炸裂开的碎片，默认9。只能时开方值，1取1；3取4

9、

1、方法重载

addClass、focus、hide、position、removeClass、show、toggle、toggleClass

2、选择器

$( "div:data(foo)") 匹配一个通过 .data( "foo", value ) 存储数据的 div。

3、元素可聚集：下面类型的元素如果未被禁用，则是可聚焦的（focusable）：

input、select、textarea、button 和 object。

锚如果带有 href 或 tabindex 属性，则是可聚焦的（focusable）。

area 元素如果在一个已命名的 map 内，且带有 href 属性，且有一个可见的图像使用了该 map，则是可聚焦的（focusable）。所有 其他完全基于 tabindex 属性和可见度的元素是可聚焦的（focusable）。

小部件

折叠面板 accordion

<h>标题，<div>同级的标签是显示的详情

1、样式：ui-accordion外层容器，ui-accordion-header标题，ui-accordion-content内容

2、属性：

active 激活面板

bool：设置所有，需collapsible为true；integer：激活单个，0开始

$( ".selector" ).accordion({ active: 2 });

初始化后设置：$( ".selector" ).accordion( "option", "active", 2 );

获取：var active = $( ".selector" ).accordion( "option", "active" );

animate 动画

boolean：false禁用

number：默认动画时间

string：动画

object：easing或duration

collapsible 是否全部可折叠

true

disabled 禁用accordion，不能折叠

默认false

event 触发折叠的事件类型

默认click dblclick：doubleclick双击

header 标题选择器，类型selector

通过调用find查找子元素

$( ".selector" ).accordion({ header: "h3" });

heightStyle 面板高度

auto：和最高的保持一致

fill：基于父类

content：高度和内容保持一致

{ heightStyle: "fill" }

icons 图标

设置为false不显示

header：不活动时的样式

activeHeader：活动时的样式

2、方法：

调用：$( ".selector" ).accordion( "enable" );

destroy 释放

disable 禁用

enable 启用

option(optionName) 获取关联值

option(optionName，value) 设置关联值

option(object) 设置多个值

refresh 更新

widget 返回一个包括accordion的jQuery对象

3、事件

activate(event ,ui) 面板被激活后触发（动画完成后）

event：事件类型 type时accordionactivate

ui：object (jQuery对象)newHeader：激活标题，oldHeader，newPanel激活面板，oldPanel

当正在折叠或者打开时都处于折叠状态。

accordion不会触发事件，可以使用create

绑定：

1、初始化时绑定：$().accordion({ activate=function(event,ui){ } })

2、绑定：$().on("accordionactivate",function(event,ui){})

beforeActivate(event,ui) 面板被激活前触发

事件类型：accordionbeforeactivate

event：事件类型 type时accordionbeforeactivate

ui：object (jQuery对象)newHeader：将激活标题，oldHeader，newPanel将激活面板，oldPanel

当正在折叠或者打开时都处于折叠状态。

create(event,ui) 创建时触发

事件类型：accordioncreate

ui：header激活的标题， panel激活的面板

自动完成部件：autocomplete

属性

appendTo： 类型selector 菜单应该被附加到哪一个元素上

为null时，附加到ui-front或body上

autoFocus： boolean true筛选后的第一条自动获取焦点。默认false

delay： integer 按键和执行搜索之间的延迟，毫秒。默认300

disabled： boolean true禁用

minLength： 必须输入的最小字符数，默认1

position： 菜单的位置

source： 数据源

Array：字符串数组（相等于label/value一样）、[{label:,value:}] label显示再下拉框中，value显示再input中；只提供value

string：url，输入foo，则get请求为url+?term=foo

Function(Object request, Function response(Object data))

request：带有一个term属性，表示文本输入的值

response：回调函数，参数时建议给用户的数据

$( "#autocomplete" ).autocomplete({

source: function( request, response ) {

var matcher = new RegExp( "^" + $.ui.autocomplete.escapeRegex( request.term ), "i" );

response( $.grep( tags, function( item ){

return matcher.test( item );

})

);

}

});

方法：

close：关闭

destroy：移除

disable：禁用

enable：启用

option(optionName)：要获取的选项的名称

option()：获取一个包含键/值对的对象，

option(optionName,value)：设置

option(options)：设置一个或多个选项

search([value])：触发 search 事件，如果该事件未被取消则调用数据源。当被点击时，可被类似选择框按钮用来打开建议。当不带参数调用该方法时，则使用当前输入的值。可带一个空字符串和 minLength: 0 进行调用，来显示所有的条目。

事件

change(event,ui)：输入域发生改变时触发，ui从菜单中选择的条目，否则属性为null

close

create

focus

open

response：搜索完成

search

select

### 11-7 html

--1、basefont 规定页面上的默认字体颜色和字号：---不赞成使用。请使用样式取代它。

color rgb(x,x,x)/#xxxxxx/colorname 规定文档中的默认文本颜色。

face font\_family 规定文档中的默认字体。

size number 规定文档中的默认字体大小。

2、bdo 设置文本方向。只能设置内部的文本，标签则不能

dir ltr/rtl 定义文字的方向(ltr--left to right默认)

--3、big 大号字体效果

4、blockquote 块引用：进行缩进（文本都会从常规文本中分离出来，经常会在左、右两边进行缩进（增加外边距），而且有时会使用斜体。）

<blockquote>

here is a long quotation here is a long quotation

<p>here is a long quotation here is a long quotation</p>

</blockquote>

5、body 定义文档的主体。必需的元素

有一些默认属性，但不赞成使用

6、<br> 换行符。<br />：xhtml 注释：请使用 <br> 来输入空行，而不是分割段落。

？clear属性

7、button

type button/reset/submit 规定按钮的类型。

value text 规定按钮的初始值。可由脚本进行修改。

name button\_name 规定按钮的名称。

autofocus autofocus 规定当页面加载时按钮应当自动地获得焦点。

formaction/ url 覆盖 form 元素的 action 属性。注释：该属性与 type="submit" 配合使用。

formenctype/formmethod/formnovalidate/formtarget

8、canvas 定义图形，比如图表和其他图像；只是图形容器，您必须使用脚本来绘制图形。

<canvas id="myCanvas"></canvas>

height/width 在style中设置宽度和高度不支持

画板和画纸的默认宽高是300\*150

1、 width 和 height 属性是设定画板和画纸的宽高，

2、 style样式 里设定的是仅画板的宽高，画纸的宽高还是为默认值300\*150，

画方形 var canvas=document.getElementById('myCanvas');

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

ctx.fillStyle='#FF0000';

ctx.fillRect(0,0,80,100);

画线条 cxt.moveTo(10,10);//定义起始位置

cxt.lineTo(150,50);

cxt.lineTo(10,50);

cxt.strokeStyle="red";//设置画笔颜色

cxt.stroke();

画圆形 cxt.fillStyle="#FF0000";//填充而不是指定画笔颜色

cxt.beginPath();

cxt.arc(70,18,15,0,Math.PI\*2,true);

cxt.closePath();

cxt.fill();

void ctx.arc(x, y, radius, startAngle, endAngle, anticlockwise);

Parameters

x

圆弧中心（圆心）的 x 轴坐标。

y

圆弧中心（圆心）的 y 轴坐标。

radius

圆弧的半径。

startAngle

圆弧的起始点， x轴方向开始计算，单位以弧度表示。

endAngle

圆弧的终点， 单位以弧度表示。

anticlockwise 可选

可选的Boolean值 ，如果为 true，逆时针绘制圆弧，反之，顺时针绘制。

图片 var img=new Image()

img.src="flower.png"

cxt.drawImage(img,0,0);

渐变 var grd=cxt.createLinearGradient(0,0,175,50);

grd.addColorStop(0,"#FF0000");

grd.addColorStop(1,"#00FF00");

cxt.fillStyle=grd;

cxt.fillRect(0,0,175,50);

10、cite 所包含的文本对某个参考文献的引用，比如书籍或者杂志的标题；以斜体显示

11、col 元素为表格中的三个列规定了不同的对齐方式

12、<dl> 标签用于结合 <dt> （定义列表中的项目）和 <dd> （描述列表中的项目）

13、del删除、ins下划线

<del>二十</del> <ins>十二</ins>

14、div 定义文档中的分区或节（division/section）。块级元素：换行

15、embed 定义嵌入的内容，比如插件。

<embed src="helloworld.swf" />

height pixels 设置嵌入内容的高度。

src url 嵌入内容的 URL。

type type 定义嵌入内容的类型。

width pixels 设置嵌入内容的宽度。

16、<footer> 标签是 HTML 5 中的新标签。

17、form 用于为用户输入创建 HTML 表单，用于向服务器传输数据。块级元素

18、frameset 框架集，用来组织多个窗口（框架）。

cols pixels/%/\* 定义框架集中列的数目和尺寸。

rows pixels/%/\* 定义框架集中行的数目和尺寸。

rows="150,300,150"：除非浏览器窗口正好是 600 像素高，否则浏览器会按比例分配

cols="100, \*"：生成一个宽为固定的 100 像素的列，然后再生成另一个框架列，该列会占据框架集中其余所有的空间。

cols="100, 3\*，\*"：

19、frame 窗口，不能和body一起使用

frameborder 0/1 规定是否显示框架周围的边框。

longdesc URL 规定一个包含有关框架内容的长描述的页面。

marginheight pixels 定义框架的上方和下方的边距。

marginwidth pixels 定义框架的左侧和右侧的边距。

name name 规定框架的名称。

noresize noresize 规定无法调整框架的大小。

scrolling yes/no/auto 规定是否在框架中显示滚动条。

src URL 规定在框架中显示的文档的 URL。

<frameset cols="500,500,1000">

<frame src="frame\_a.htm" />

<frame src="frame\_b.htm" />

<frame src="frame\_c.htm" />

</frameset>

noframes 元素可为那些不支持框架的浏览器显示文本

1、<h1-h6>

2、head 定义文档的头部，它是所有头部元素的容器。必需的元素。<base>, <link>, <meta>, <script>, <style>, 以及 <title>。

title 定义文档的标题，它是 head 部分中唯一必需的元素。

link 链接一个外部样式表 <link rel="stylesheet" type="text/css" href="theme.css" />

meta 元素可提供有关页面的元信息（meta-information）

<meta name="keywords" content="HTML,ASP,PHP,SQL">

<meta http-equiv="charset" content="iso-8859-1">

http-equiv content-type/expires/refresh/set-cookie 把 content 属性关联到 HTTP 头部。

name author/description/keywords/generator/revised/others 把 content 属性关联到一个名称。

scheme some\_text 定义用于翻译 content 属性值的格式。

base

href URL 规定页面中所有相对链接的基准 URL。

target 在何处打开页面中所有的链接。

\_blank 在新窗口中打开被链接文档。

\_self 默认。在相同的框架中打开被链接文档。

\_parent 在父框架集中打开被链接文档。

\_top 在整个窗口中打开被链接文档。

framename 在指定的框架中打开被链接文档。

位于 head 元素内部

为页面上的所有链接规定默认地址或默认目标。通常情况下，浏览器会从当前文档的 URL 中提取相应的元素来填写相对 URL 中的空白。使用 <base> 标签可以改变这一点。浏览器随后将不再使用当前文档的 URL，而使用指定的基本 URL 来解析所有的相对 URL。这其中包括 <a>、<img>、<link>、<form> 标签中的 URL。

3、meter 度量给定范围（gauge）内的数据

<meter value="0.6">60%</meter>

form form\_id 规定 <meter> 元素所属的一个或多个表单。

high number 规定被视作高的值的范围。

low number 规定被视作低的值的范围。

max number 规定范围的最大值。

min number 规定范围的最小值。

optimum number 规定度量的优化值。

value number 必需。规定度量的当前值。

4、nav HTML 5 中的新标签。定义导航链接的部分

style中定义height/width

5、hr 水平线<hr />

6、html 限定了文档的开始点和结束点，在它们之间是文档的头部 <head>和主体<body>。

7、i 斜体 <i></i>

8、iframe 创建另一个文档的内联框架（即行内框架

frameborder 0/1 规定是否显示框架周围的边框。

longdesc URL 规定一个包含有关框架内容的长描述的页面。

marginheight pixels 定义框架的上方和下方的边距。

marginwidth pixels 定义框架的左侧和右侧的边距。

name name 规定框架的名称。

noresize noresize 规定无法调整框架的大小。

scrolling yes/no/auto 规定是否在框架中显示滚动条。

src URL 规定在框架中显示的文档的 URL。

width/height

9、img 向网页中嵌入一幅图像

<img src="/i/eg\_tulip.jpg" alt="上海鲜花港 - 郁金香" />

alt text 规定图像的替代文本。

src URL 规定显示图像的 URL。

width/height

10、ul无序列表（有） ol有序列表（li前面有序号）

ol

reversed reversed 规定列表顺序为降序。(9,8,7...)

start number 规定有序列表的起始值。

type 1/A/a/I/i 规定在列表中使用的标记类型。

ul type disc/square/circle 不赞成使用。请使用样式取代它。规定列表的项目符号的类型。

style="list-style-type:square"

li

11、output

<form oninput="x.value=parseInt(a.value)+parseInt(b.value)"> 0

<input type="range" id="a" value="50">100

+<input type="number" id="b" value="50">

=<output name="x" for="a b"></output>

</form>

12、p 段落

13、object param 添加一个对象和属性

14、pre 显示预格式化的文本，

标签中的特殊符号被转换为符号实体： "&lt;" 代表 "<"，"&gt;" 代表 ">"

15、progress 标签标示任务的进度（进程）下载条

max number 规定任务一共需要多少工作。

value number 规定已经完成多少任务。

<progress></progress>什么都不写，则是一个不断滚动的条。只有<progress>则忽略后面的内容。

16、q 短的引用，一般加引号 Quotes

17、ruby

18、script 定义一段javascript代码，

<script type="text/javascript">

async async 规定异步执行脚本（仅适用于外部脚本）。

charset charset 规定在外部脚本文件中使用的字符编码。

defer defer 规定是否对脚本执行进行延迟，直到页面加载为止。

src URL 规定外部脚本文件的 URL。

19、section 定义一个节，html5中的

20、<audio controls>

<source src="horse.ogg" type="audio/ogg">

<source src="horse.mp3" type="audio/mpeg">

Your browser does not support the audio element.

</audio>

1、span 行内元素

2、sub 下标

3、style 定义样式信息。type 属性是必需的，定义 style 元素的内容。唯一可能的值是 "text/css"。

style 元素位于 head 部分中。

4、sup 上标

5、table

border pixels 规定最外层的边框的box

cellpadding pixels/% 规定单元边沿与其内容之间的空白。

cellspacing pixels/% 规定单元格之间的空白。

frame 规定外侧边框的哪个部分是可见的。

void无/

above上/below/ lhs左/rhs右

hsides上下（ horizontal）/vsides左右（vertical）

box四边/border 效果一样

rules none/groups/rows/cols/all 规定内侧边框的哪个部分是可见的。

summary text 规定表格的摘要。

width %/pixels 规定表格的宽度。

如果您使用thead、tfoot 以及 tbody 元素，您就必须使用全部的元素。它们的出现次序是：thead、tfoot、tbody，

<thead>、<tbody> 以及 <tfoot> 很少被使用，这是因为糟糕的浏览器支持。

5.1、tr 行

5.2、th 表头单元格/一般为粗体

5.3、td 标准单元格--一般显示为正常字体且左对齐

abbr text 规定单元格中内容的缩写版本。

align 规定单元格内容的水平对齐方式。

left

right

center

justify 对行进行伸展

char

colspan number 规定单元格可横跨的列数。

headers header\_cells'\_id 规定与单元格相关的表头。

rowspan number 规定单元格可横跨的行数。

valign 规定单元格内容的垂直排列方式。

top

middle

bottom

baseline

caption 元素定义表格标题。

必须紧随 table 标签之后。您只能对每个表格定义一个标题。通常这个标题会被居中于表格之上。

6、tt 等宽字体

7、video HTML5 视频

autoplay autoplay 如果出现该属性，则视频在就绪后马上播放。

controls controls 如果出现该属性，则向用户显示控件，比如播放按钮。

height pixels 设置视频播放器的高度。

loop loop 如果出现该属性，则当媒介文件完成播放后再次开始播放。

muted muted 规定视频的音频输出应该被静音。

poster URL 规定视频下载时显示的图像，或者在用户点击播放按钮前显示的图像。

preload preload

如果出现该属性，则视频在页面加载时进行加载，并预备播放。

如果使用 "autoplay"，则忽略该属性。

src url 要播放的视频的 URL。

width pixels 设置视频播放器的宽度。

HTML属性：

draggable 规定元素是否可拖动。

tabindex 规定元素的 tab 键次序。

style 规定元素的行内 CSS 样式。

1、input type="reset"（按钮自动显示重置） 重置同一form中的表单，为默认值

2、type="submit"自动提交'

3、type="file"：选择文件

4、type="image" src=""：不写src，则是一个提交按钮

5、 <select size="1">disabled="true" multiple="true"

<option value="0">0</option>

<option selected="selected">1</option>

</select>

option属性

disabled disabled 规定此选项应在首次加载时被禁用。

label text 定义当使用 <optgroup> 时所使用的标注/直接使用则替换中间的值。

selected selected 规定选项（在首次显示在列表中时）表现为选中状态。

value text 定义送往服务器的选项值。

<optgroup label="Swedish Cars">把相关的选项组合在一起

6、点击文字自动选择控件

<label for="male">12</label><input type="checkbox" style="vertical-align: middle;" id="male">//

<label >12<input type="checkbox" style="vertical-align: middle;" id="male"></label>

7、disabled与readonly

　　　 禁用和只读属性。readonly只针对input(text / password)和textarea有效，

而disabled对于所有的表单元素都有效，包括select, radio, checkbox, button等。

但是表单元素在使用了disabled后，当我们将表单以POST或GET的方式提交的话，这个元素的值不会被传递出去，

而readonly会将该值传递出去。

8、<textarea> </textarea>

控制大小：加width,height限制(style="width:100px;height:200px;");或row,col限制(<textarea rows="5" cols="7">);

限制拖动： style="resize:none;"

滚动条效果： style="overflow:scroll;" //没有超出也显示滚动条

对滚动条隐藏:

水平滚动条: style="overflow-x:hidden;"

垂直滚动条: style="overflow-y:hidden;"

9、fieldSet分组

<fieldset>

<legend>信息</legend>

<label>身高：<input type="text"></label>

</fieldset>

10、abbr：被包含的元素，鼠标移上去，显示title值

The <abbr title="\*\*">sb</abbr>

11、a：超链接 href type

<a target="\_blank" href="http://www.chinawutong.com/">物通</a> 百度等的网站都有限制

12、address：斜体

13、area/map图像映射（image-map）指带有可点击区域的一幅图像。

area 元素永远嵌套在 map 元素内部。area 元素可定义图像映射中的区域。

<img>中的 usemap 属性可引用 <map> 中的 id 或 name 属性（取决于浏览器），所以我们应同时向 <map> 添加 id 和 name 属性。

圆形（circ 或 circle）

多边形（poly 或 polygon）

矩形（rect 或 rectangle）

default 意味着该区域覆盖整个图像。在实际中，浏览器默认使用矩形区域，并期望能找到 4 个 coords 值。如果没有指定形状，而且在标签中也没有包括 4 个坐标，那么浏览器会忽略整个区域。

<img src="/i/eg\_planets.jpg" border="0" usemap="#planetmap" alt="Planets" />

<map name="planetmap" id="planetmap">

圆：圆心（横纵），半径

<area shape="circle" coords="180,139,14" href ="/example/html/venus.html" target ="\_blank" alt="Venus" />

长方形：左上角坐标，右下角坐标

<area shape="rect" coords="0,0,110,260" href ="/example/html/sun.html" target ="\_self" alt="Sun" />

</map>

14、b:粗体

### 11-6 15天考试、input

15天考试

1、设计模式

2、装箱和拆箱

3、list和string[]

4、冒泡

5、分页：

6、List、ArrayList、数组

create procedure procName(@pageNum int, ) as begin end go

from() as Table没有写 as

input属性

checked checked 规定此 input 元素首次加载时应当被选中。

alt text 定义图像输入的替代文本。

autocomplete 5 on/off 规定是否使用输入字段的自动完成功能。

height 5 pixels/% 定义 input 字段的高度。（适用于 type="image"）

max 5 number/date 规定输入字段的最大值。请与 "min" 属性配合使用，来创建合法值的范围。

maxlength number 规定输入字段中的字符的最大长度。

pattern 5 regexp\_pattern 规定输入字段的值的模式或格式。例如 pattern="[0-9]" 表示输入值必须是 0 与 9 之间的数字

placeholder 5 text 规定帮助用户填写输入字段的提示。

required 5 required 指示输入字段的值是必需的。

value value 规定 input 元素的值。

## 11-1

1、cboPayType.Text ；comboBox获取选中的值； SelectedText为""

2、 SQLite建表【CREATE TABLE IF NOT EXISTS t\_Order（）】、

引用【using System.Data.SQLite;】、

连接：new SQLiteConnection("data source=d:\Logistic.sqlite" )

SQL读取数据到表中：SqlDataAdapter adapter = new SqlDataAdapter(cmd); adapter.Fill(dt);

没有自增：选取max(id) null 1:+1

事务：SQLiteTransaction tran = cn.BeginTransaction(); cmd.Transaction = tran;

cmd.Parameters.AddWithValue("@id", item[0]);

cmd.ExecuteNonQuery();

tran.Commit();

3、省市县

ListBox：失去焦点隐藏【Leave】、切换显示、回车选中【KeyDown：e.KeyCode == Keys.Enter】

4、货物

列数据必填：放在Panel上，遍历，有值遍历；

5、屏幕随鼠标移动

鼠标在窗口上，且点击鼠标移动（一直点着），获取鼠标移动的相对位置（往上、右的变动距离），窗口位置加上相对值

【MouseDown，设置一个bool标志，获取鼠标在窗口上的位置；MouseMove，设置窗口移动；Controls\_MouseUp，false标志】

6、背景色的格式：BackgroundImageLayout--

ImageLayout

None = 0, // 图像沿控件的矩形工作区顶部左对齐。

Tile = 1,// 图像沿控件的矩形工作区平铺。

Center = 2,// 图像在控件的矩形工作区中居中显示。

Stretch = 3,// 图像沿控件的矩形工作区拉伸。

Zoom = 4, // 图像在控件的矩形工作区中放大。

最大化窗口：this.WindowState = FormWindowState.Maximized;

7、关闭窗口：System.Environment.Exit(0);

2017-10-26

1、 private void button2\_Click(object sender, EventArgs e)

{

OpenFileDialog filename = new OpenFileDialog(); //定义打开文件

filename.InitialDirectory = Application.StartupPath; //初始路径,这里设置的是程序的起始位置，可自由设置

filename.Filter = "All files(\*.\*)|\*.\*|txt files(\*.txt)|\*.txt";//设置打开类型,设置个\*.\*和\*.txt就行了

filename.FilterIndex = 2; //文件类型的显示顺序（上一行.txt设为第二位）

filename.RestoreDirectory = true; //对话框记忆之前打开的目录

if(filename.ShowDialog() == DialogResult.OK)

{

textBox1.Text = filename.FileName.ToString();//获得完整路径在textBox1中显示

StreamReader sr = new StreamReader(filename.FileName,Encoding.Default);//将选中的文件在textBox2中显示

textBox2.Text = sr.ReadToEnd();

sr.Close();

}

}

2、Math.Floor 返回double

2017-10-26

1、File类、FileInfo、FileStream、Path：追加、复制、创建、是否存在、得到创建日期、移动、打开、读取、替换、设置时间、写入

2、 int test = (int)(wait \* 1000);

与(int)wait \* 1000; 好像先执行(int)wait

3、缓存：Cache cache=HttpRunTime.Cache;//HttpContext.Current.Cache;

把数据直接保存到一个txt文件中，以后读取。

4、关于数据库返回数据字符串只有2033

5、 DataTable

foreach (DataColumn item in dtWT.Columns)

{

string str=item.ColumnName;

}

dt.Rows.Add(dtWT.Rows[0]); 其他信息: 该行已经属于另一个表。

6、取所有数据的前10000条，有的类别中的数据有点少。在数据库中

先取第一个表中的数据，取完再取第二个表中的数据...

根据类别创建10个表

选择的时候，根据选择，筛选出特定的表 3

读取第一个表中的数据，如果表中的数据读取完，且pageCurrent=-1则读取下一个表中的数据。

三个指针：第一个是表的数目，第二个是读取的数目。

7、此行提示该行已经属于另一个表

//dt2.Rows.Add(dr);

//换成这种写法是正确的或者可以直接使用dt2.Rows.Add(dt1.Rows[i].ItemArray);

dt2.Rows.Add(dr.ItemArray);

原因在于DataTable newdt = new DataTable();此时newTable已经生成，但是此时并未给此表添加列,因此newTable表里的列数为0,在你dt.Rows.Add(rows);时就会异常。

解决的方法是:在DataTable newTable = new DataTable();之后使用newTable.Columns.add("列名");给newTable表对象加上列.然后再添加行

8、BackgroundWorker是.NET里面用来执行多线程任务的控件，它允许编程者在一个单独的线程上执行一些操作。耗时的操作（如下载和数据库事务）。

2017-10-25

1、System.Windows.Forms.Control.CheckForIllegalCrossThreadCalls = false;

在winform程序中运用到多线程调用ui时会出现“从不是创建控件 （控件名称）的线程访问它。” 的错误警告。使用这个则可以调用，没有警告

温馨提示：Control.CheckForIllegalCrossThreadCalls = false用这种方法来关闭安全检测并不是根本的解决方法。

CheckForIllegalCrossThreadCalls 属性是在using System.Windows.Forms命名空间下的。可以对单个控件进行该属性的设置，如button1.CheckForIllegalCrossThreadCalls=false。

2、SmtpClient client = new SmtpClient(server, 25);

关于别的机器不能连接的问题：让对方根据server地址登陆一下；换成ip地址；SmtpClient中需要的端口号是25（Smtp协议的端口号，而不是邮箱中的端口号）。

3、SqlConnection：server=;database;uid;pwd或者 server

Connection Timeout连接超时，秒；

Integrated Security- 或 -Trusted\_Connection：当false，连接中指定用户 ID 和密码。 当true，当前的 Windows 帐户凭据用于身份验证。可识别的值为true， false， yes， no，和sspi（强烈建议），这等同于true。

如果用户 ID 和密码是指定的集成安全性设置为 true，用户 ID 和密码将被忽略，并将使用集成安全性。

SqlCredential是更安全的方式指定凭据使用的连接SQL Server身份验证 (Integrated Security=false)。

password- 或 -PWD：密码

Initial Catalog- 或 -database：数据库名称

User ID- 或 -UID：用户密码

Data Source- 或 -server- 或 -Address- 或 -Addr- 或 -network address：服务器地址

DataSource 属性、Database 属性

conn.Credential = cred;号称是最安全的连接方式，但4.5之后才能用；

2017-10-24

1、对象还存在，但是已经释放。不能调用

2、 private void OrderForm\_FormClosed(object sender, FormClosedEventArgs e)

{

form.Close();

//form = null;

}

FormClosing事件

在窗体关闭时，FormClosing事件发生。此事件会得到处理。从而释放与窗体相关的所有资源。

如果取消此事件，则窗体仍然保持打开状态。

当窗体显示为模式对话框时，单击“关闭”会隐藏窗体并将DialogResult属性设为Cancel。

通过在些事件中设置DialogResult属性可以在用户单击右上角关闭按钮时重写DialogResult的值。

FormClosed事件

在用户或Application类的Close方法或Exit方法关闭窗体后，会发生FormClosed事件。

3、使子窗体不关闭 主窗体不可用

FormB fb=new FormB();

db.ShowDialogue();//模式窗体

4、判断是否有网络

无非几种解决方案

1、调用api查询当前状态

2、wmi查看当前连接

3、尝试打开某网页

4、尝试ping或telnet

最后还是觉得，根据具体要求选择比较好，如果是连接到某站点，直接尝试是否能打开站点页面。

或者是数据库连接或通讯连接等，都可用telnet尝试。

仅局域网可尝试ping网关或者发广播包等

5、SQLite获取表结构

PRAGMA table\_info('t\_Order')获取表结构

6、groupBox1.Location = new Point(groupBox1.Location.X, groupBox1.Location.Y + extendHeight);

7、++this.llpGoods.RowCount ;可以改变panel的列，但--不可以

8、http状态码：1服务器接收到请求 2请求成功 3重定向 4客户端错误 5服务器错误

9、HTTP：无连接（应答完立刻断开）、媒体独立（可以发送任何的数据）、无状态（不记录上一次请求）

请求消息：请求行、请求头、空行、请求数据

请求行：请求方法 空格 URL 空格 协议版本 回车符 换行符

请求头：键/值对 回车符 换行符

回车符 换行符

请求数据

GET /hello.txt HTTP/1.1

User-Agent: curl/7.16.3 libcurl/7.16.3 OpenSSL/0.9.7l zlib/1.2.3

Host: www.example.com

Accept-Language: en, mi

响应消息：状态行、消息报头、空行、响应正文

状态行：协议版本 状态码

消息报头：名值对 回车符 换行符

正文

HTTP/1.1 200 OK

Date: Mon, 27 Jul 2009 12:28:53 GMT

Server: Apache

Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT

ETag: "34aa387-d-1568eb00"

Accept-Ranges: bytes

Content-Length: 51

Vary: Accept-Encoding

Content-Type: text/plain

# 数据库

11-17

修改数据库：修改Action中53的adminaction为1

update action set adminaction=1 where id=53 and adminaction=0

select col.name,t.name,col.length,pro.value from sys.syscolumns col

left join sys.systypes t

on col.xtype=t.xusertype

left join sys.extended\_properties pro

on pro.major\_id=object\_id('Action') and pro.minor\_id=col.colid

where col.id=object\_id('Action')

## 1、自有表

37 FROM dbo.syscolumns col

38 LEFT JOIN dbo.systypes t ON col.xtype = t.xusertype

39 inner JOIN dbo.sysobjects obj ON col.id = obj.id

40 AND obj.xtype = 'U'

41 AND obj.status >= 0

42 LEFT JOIN dbo.syscomments comm ON col.cdefault = comm.id

43 LEFT JOIN sys.extended\_properties ep ON col.id = ep.major\_id

44 AND col.colid = ep.minor\_id

45 AND ep.name = 'MS\_Description'

46 LEFT JOIN sys.extended\_properties epTwo ON obj.id = epTwo.major\_id

47 AND epTwo.minor\_id = 0

48 AND epTwo.name = 'MS\_Description'

49 WHERE obj.name = 'Ath\_LoanApply\_tbl'--表名

50 ORDER BY col.colorder ;

dbo.syscolumns 当前数据库的所有字段都保留在里面。

name sysname, --字段名称

id int, --该字段所属的表的ID

xtype tinyInt, --该字段类型，关联sysTypes表

length smallint, --该字段物理存储长度

colid 列id

dbo.systypes SQL SERVER 数据库系统表systypes存储了字段类型

name sysname 数据类型名称。

xtype tinyint 物理存储类型。

xusertype 使用这个 smallint 扩展用户类型。

length smallint 数据类型的物理长度。

dbo.sysobjects:系统对象表。 保存当前数据库的对象，如约束、默认值、日志、规则、存储过程等

name sysname 对象名。

Id int 对象标识号。

xtype char(2) 对象类型。可以是下列对象类型中的一种：

C = CHECK 约束

D = 默认值或 DEFAULT 约束

F = FOREIGN KEY 约束

L = 日志

FN = 标量函数

IF = 内嵌表函数

P = 存储过程

PK = PRIMARY KEY 约束（类型是 K）

RF = 复制筛选存储过程

S = 系统表

TF = 表函数

TR = 触发器

U = 用户表

UQ = UNIQUE 约束（类型是 K）

V = 视图

X = 扩展存储过程

dbo.syscomments 包含数据库中每个视图、规则、默认值、触发器、CHECK 约束、DEFAULT 约束和存储过程的项。text 列包含原始的 SQL 定义语句。

不知道为什么不能查到数据

sys.extended\_properties

class tinyint 标识其上存在属性的项类。 可以是下列值之一：

0 = 数据库

1 = 对象或列

2 = 参数

3 = 架构

4 = 数据库主体

5 = 程序集

6 = 类型

7 = 索引

10 = XML 架构集合

15 = 消息类型

16 = 服务约定

17 = 服务

18 = 远程服务绑定

19 = 路由

20 = 数据空间（文件组或分区方案）

21 = 分区函数

22 = 数据库文件

27 = 计划指南

class\_desc nvarchar(60) 其上存在扩展属性的类的说明。 可以是下列值之一：

major\_id int 表id 其上存在扩展属性的项的 ID，根据项类进行解释。 对于大多数项，该 ID 适用于类所表示的项。 下列是非标准主 ID 的解释：

如果 class 为 0，则 major\_id 始终为 0。

如果 class 为 1、2 或 7，则 major\_id 为 object\_id。

minor\_id int 列id 其上存在扩展属性的项的辅助 ID，根据项类进行解释。 对于大多数项，ID 为 0；否则，ID 为下列值之一：

如果 class = 1，则 minor\_id 在项为列的情况下等于 column\_id，在项为对象的情况下等于 0。

如果 class = 2，则 minor\_id 为 parameter\_id。

如果 class = 7，则 minor\_id 为 index\_id。

name sysname 属性名，其 class、major\_id 和 minor\_id 是唯一的。

value sql\_variant 注释

OBJECT\_ID ( '[ database\_name . [ schema\_name ] . | schema\_name . ]

object\_name' [ ,'object\_type' ] ) 返回架构范围内对象的数据库对象标识号

架构是指包含表、视图、过程等的容器。

sys是包含系统对象的架构。

dbo是用户默认架构，sysadmin 固定服务器角色的所有成员都有默认架构 dbo.

所以sys.sysobjects==dbo.sysobjects