编码规范

1. C#与javascript命名规范:

变量命名的第一原则：作用范围内意义唯一

举个例子，如果在函数里有一条表示学校的记录变量叫record，后面又遇到一条记录表示学生，这时会被命名为studentRecord，这就违反了意义唯一的原则，第一条记录需要命名为schoolRecord。

变量命名的第二原则：不可以自己创造简写

例如WordPracticeLesson简写成wpl是不允许的，简写的一个重要标准是大家约定且都认识的。

变量命名的第三原则：表明自身是什么

如果是使用facade层的对象，那就是xxxxFacade，如果使用CacheSvc层的对象，那就是xxxCacheSvc，在xxx中仍然需要遵循第二原则。

函数命名第一原则： 动宾短语

即表明清楚要做的动作，和作用的对象，并且使用By, Of之类的词加以细节修饰，比如

GetStudentBySchoolID，就是一个清晰的函数命名。

另外，如果是获取单条记录请使用Get，获取一组记录使用List，

如果是非数据库对应记录的数据结构体，使用Fetch -------- 1.1

函数命名第二原则： 补充说明

如果实在无法在命名中体现出需要体现的东西时，这时就需要使用注释作为补充。

注释原则

除了上述特殊应用下的注释外，一般对于注释的要求是程序中出现特殊的判断(if)或者算法设计时，例如你需要使用一个字符串的按位操作来存储数据，就是需要注释清楚每一位的意义的。

空行与空格的处理 -------------------1.1

无论是C#，还是javascript代码书写中的空格处理应同C#代码中系统自动排版的格式保持一致；

函数与函数间保持空一行，一个函数中不同的逻辑块之间空一行，例如，函数中需要先生成订单数据，再生成订单子项的数据，则可能是类似下面的结构:

OrderInfo orderInfo = new OrderInfo();

orderInfo.Id = 3;

orderInfo.Name = "合同1";

.....

SaveOrder(Order)

(空一行）

List<OrderItem> orderItemList = new List<OrderItem>(10);

OrderItem itemInfo = new OrderItem();

......

SaveOrderItem(orderItemList);

2. jQuery使用规范

1.所有事件绑定使用 document.on，不得使用bind/ live绑定事件，严格区分 Event handler和 delegated-events handler

具体见：http://api.jquery.com/on/

2. 所有页面开始响应事件要求写完整，即完整的$(document).ready(function() { });

3. 页面的功能聚合使用闭包，如果一个页面完成两个大项功能，则使用两个闭包，闭包的设计遵循功能单一原则。所有闭包的命名均为小写字母开头，以Mgr/Mng结尾，例如sentenceAnalysisMgr

3. CSS 规范：

在CSS前缀限定中，除非所属层级实在过于简单，一般需要遵循的限定为三级制：

#DIV-ID .区域样式名 子级元素，例如

#newexample .editarea button { color: white; }

#mainarticle .paragraphlist p { line-height: 28px; }

总之需要用#Div-ID确定所属的大范围，使用窗口样式指定具体的功能区域，最后缀上要修饰的元素，最后一级可以是样式名，也可以是元素名，根据情况而定。

所有CSS样式名必须使用小写

html 规范:

1. 除了页面框架级的元素和模式窗体后，其它元素均不得使用ID，

显示对应信息的全部使用[data-field]，如姓名 <input type='text' data-field='name' />

表示操作的使用使用[data-act]，如<button data-act='save'>button</button>

2. 所有在html中出现的元素名，样式名，均不可以包括其所依附的元素，比如不可以出来 articleDiv，savebutton这样的div id或者样式名，在可以使用属性时，尽量不要使用ID，比如有一个按钮：

<button id='saveComment' data-act='savecomment'></button>，

这时完全可以使用data-act去实现javascript绑定的，不得使用id='saveComment'这样的属性，特殊属性下可以允许例外，比如lable中的for属性，但仅限于这些特殊应用下使用。

所有属性中的值也需要使用小写。

3. 表示步骤的使用 data-step，容器使用样式stepcontainer

表示页签的使用data-tab，容器使用样式tabcontainer，对于页签样式，统一的处理方式为：

<div class='stepcontainer'>

<div class='step' data-step='1'></div>

<div class='step' data-step='2'></div>

<div class='step' data-step='3'></div>

</div>

4. 其它属性需要在命名中区分是系统提供的答案数据，还是用户填写的数据，

如<div data-sys-senid='15' data-user-articleid=213'>

5. MVC

mvc: 禁止使用 @Url.Content，全部使用@Url.StaticContent扩展方法 来引入资源

名词术语表

UD\_ 此为自定义数据包，主要是为了实现一个练习页面多处应用而设计的通用数据返回格式。例如UD\_SentenceAnalysis。