**WEB界面编码规范**

**编制单位：西部大区一部**

**编 制 人：王瀚贤**

**审 核：**

**批 准：**

**用友汽车信息科技（上海）股份有限公司**

**Yonyou Auto Information Technology (Shanghai) Co., Ltd.‘**

**文档更新记录**

| **编号** | **章节** | **变更类型** | **修订内容简述** | **修订日期** | **修订前版本** | **修订后版本** | **修订人** |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| 1 | 所有章节 | C |  | 2017-06-19 | / | 1.0 | 王瀚贤 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

\*变更类型：C——创建，A——增加，M——修改，D——删除

# 目录结构规范

1. 目录建立原则：以最少的层次提供最清晰的访问结构。
2. 目录的命名以小写英文字母、下划线组成。
3. 根目录一般只放index.html已经其他必须的系统文件。
4. 根目录下的images用于存放各页面都要使用的公用图片。

# 设计命名规范

## 页面基本框架结构

container

header

4

5

1

3

2

navbar

menu

main

container—就是将页面中的所有元素包在一起的部分；

header—是页面的头部区域，一般来讲，它包含网站的logo和其他一些元素；

navbar—等同于横向的导航栏，是最典型的页面元素，也可以命名为nav；

menu—此区域包含一般的链接和菜单，也可以命名为subNav，links；

main—是网站的主要区域，也可以命名为content；

## 元素命名规则注意点

1. 尽量考虑为元素命名其本身的作用或用意，达到语义化不要使用表面形式命名，如：red/left/big等；
2. 组合命名规则：[元素类型]-[元素作用/内容]。如搜索按钮btn-search、登录表单form-login。；
3. 凡涉及交互行为的元素通常会有正常、悬停、点击和已经浏览等不同样式。参考：搜索按钮btnSearch、btnSearchHover、btnSearchVisited。

## 常用命名汇总

|  |  |
| --- | --- |
| 名称 | 命名规范 |
| 页头 | header top |
| 登录条 | loginbar |
| 标志 | logo |
| 侧栏 | sidebar |
| 导航 | nav |
| 子导航 | subNav |
| 菜单 | menu |
| 子菜单 | submenu |
| 工具条 | toolbar |
| 表单 | form |
| 栏目 | column |
| 箭头 | arrow |
| 搜索 | search |
| 滚动条 | scroll |
| 内容 | content |
| 标签页 | tab |
| 列表 | list |
| 提示 | tips |
| 栏目标题 | title |
| 链接 | links |
| 页脚 | footer end |
| 服务 | service |
| 下载 | download |
| 注册 | register reg |
| 状态 | state |
| 按钮 | btn |
| 上传 | upload |
| 登录 | login |
| 合作伙伴 | partner |
| 版权 | copyright |
| 网站地图 | sitemap |

## 图片命名规则

1. 图片名称分为头尾两部分，用下划线隔开。禁止用中文名；
2. 头部分表示此图片的大类性质；
3. 例如：放置在顶部的长方形图片可以取名banner；标志图片取名logo；位置不固定并且带有链接的小图片取名button；做栏目链接的图片取名menu；不带链接表示标题的取名title；装饰的照片取名pic；等；
4. 尾部分用来表示图片的具体含义，用英文字母表示，如banner\_ad.gif logo\_firefox.gif button\_next.gif menu\_aboutus.gif title\_news.gif pic\_people.jpg.
5. 有onmouse效果的图片，两张分别在原有文件名后加”\_on”和”\_off”命名。

# 布局规范

推荐使用CSS+DIV来布局页面，少用table，table只用于数据处理。优势如下：

1. 表现和内容相分离，将布局放在一个独立的样式文件中，html文件只存放文本信息；
2. 提高页面浏览速度，采用CSS+DIV结构的页面容量比table的页面容量小得多，且<table>标签要等</table>下载完后才可以显示，而使用CSS+DIV标签不用等待</div>下载好就可以显示，所有速度快；
3. 易于维护和改版，只要简单的修改几个CSS文件就可以重新设计整个网站的页面。例如：

layout.css

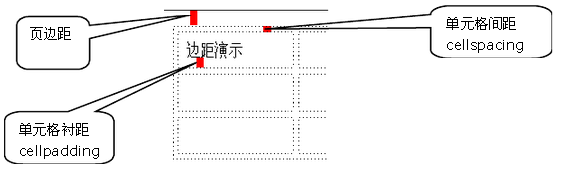
|  |
| --- |
| body { font-family:Verdana; font-size:14px; margin:0;}  #container {margin:0 auto; width:100%;}  #header { height:100px; background:#9c6; margin-bottom:5px;}  #menu { height:30px; background:#693; margin-bottom:5px;}  #mainContent { height:500px; margin-bottom:5px;}  #sidebar { float:left; width:200px; height:500px; background:#cf9;}  #sidebar2 { float:right; width:200px; height:500px; background:#cf9;}  #content { margin:0 205px !important; margin:0 202px; height:500px; background:#ffa;}  #footer { height:60px; background:#9c6;} |

index.html

|  |
| --- |
| <html>  <head>  <link href="layout.css" rel="stylesheet" type="text/css" />  </head>  <body>  <div id="container">  <div id="header">This is the Header</div>  <div id="menu">This is the Menu</div>  <div id="mainContent">  <div id="sidebar">This is the sidebar</div>  <div id="content">3列左右侧固定，中间列自适应居中+头部+导航+尾部</div>  </div>  <div id="footer">This is the footer</div>  </div>  </body>  </html> |

# 边距规范

页、表格都应该有边距，避免紧贴边沿的情况发生，最小边距值为“3px”，默认边距值应在CSS中设定。



# 字体规范

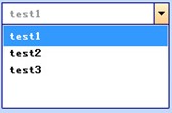
1. 字体尺寸使用pt为单位，对象和线条使用px为单位。基准字体为12pt；
2. 字体尽量采用Verdana, Arial, Helvetica, sans-serif, 宋体。一个页面最多包含两种字体；
3. 所有字体设定应该在CSS中完成。

# 颜色规范

1. 尽可能通过选择合适的主题颜色或系统颜色来选择颜色；
2. 不要混用颜色类型。也就是说，不要同时运用多个主题颜色。使用主题颜色外的颜色时最好从主题颜色派生，而不是硬编码颜色值；
3. 尽量限制色彩的数目；
4. 应当根据对象的重要性来选择颜色，重要对象用醒目的颜色表示；
5. 使用颜色的时候应当保持一致性，如错误提示用红色表示就要在其他时候保持一致。

# 输入框规范（重点）

1. 当打开一个页面时光标默认停留在第一个输入的文本框；
2. 当选择下拉框存在默认值时要默认选中。不存在默认值时，则默认为“请选择”，不要空着；

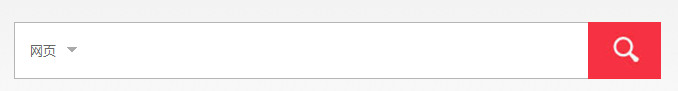
1. 必填项必须以红色\*号标识，红色\*号放在输入文本框之后；



1. 新增、编辑保存时，关闭弹出提示框提示出错后，要把焦点定位在出错的第一个输入框；
2. 只允许输入数字的输入框要控制输入的非法字符，而不是输入非法字符后给予提示；
3. 只允许输入日期、时间的输入框要给予格式化；
4. 输入内容的长度要限制，默认对应数据库相关字段的长度；
5. 对非法的字符的控制，限制输入或给予提示；
6. 只读文本框（灰色）和可输入文本框（白色）在颜色上必须区分。

# 搜索框规范

1. 搜索框文本框的长度应适中，至少应提供显示10个中文字符的宽度；
2. 搜索组件中使用的文本框必须为单行文本框；
3. 文本框的长度不得少于130个像素，高度不得低于18个像素；
4. 文本框要有标示性文字（即hint提示），颜色为灰色#cccccc，搜索按钮可突出显示。



# 提示框规范

1. 提示信息中的标点符号统一为全角符号；
2. 复杂的操作在成功后给予提示信息；
3. 提示信息不宜过长，宽度不能超过窗口的1/2，内容可考虑换行，不要滚动显示；
4. 窗口应居中显示；
5. 当按钮为图片按钮时，光标停留时需给予浮动提示；
6. 选择默认按钮，将最安全的或最常用的选项作为默认按钮。

# 键盘支持规范

1. 支持tab键移动光标焦点，使用tabindex属性改变移动顺序。

# 鼠标

1. 在整个交互式语境地中，需可以识别鼠标操作；
2. 支持滑轮（鼠标中间的滚动轮）上下翻动操作；
3. 对于相同种类的元素采用相同的操作激活；
4. 用“加载中”表示系统忙，用手型表示可以点击。

# 表单规范（重点）

1. 属性名称采用右对齐，属性名称后面不用加任何符号；
2. 文本框中对齐方式：文字、日期左对齐，数字右对齐；
3. 每一列文本框的长度对齐（文本框、选择框、下拉列表框的长度一样）。

## 单卡片

1. 从页面左上角开始；
2. 字段较多的情况，按三列处理；布局方式：从左到右，从上到下的方式；



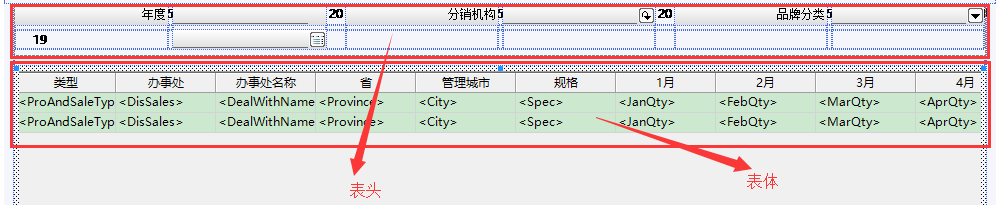
1. 字段少的情况：从上往下排列。



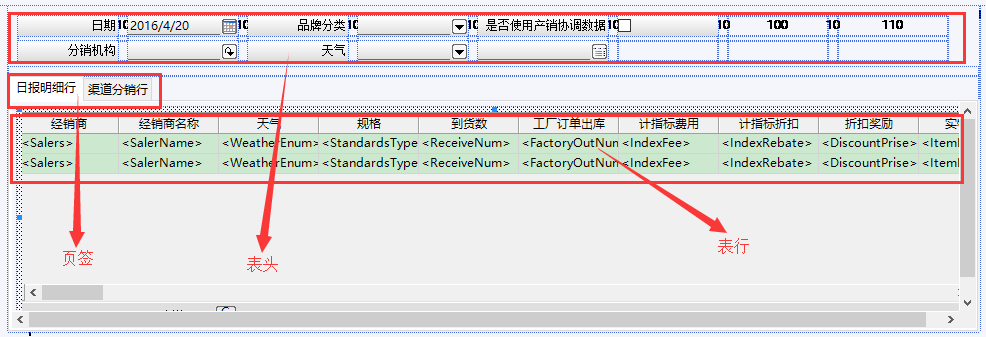
## 单卡单表（主子表）、单卡多表（主子子表）

1. 采用上面单头、档案基本信息，下面为表体；
2. 从页面左上角开始；
3. 单头、档案基本信息按三列处理；布局方式：从左到右，从上到下的方式；

* 单卡单表（主子关系）



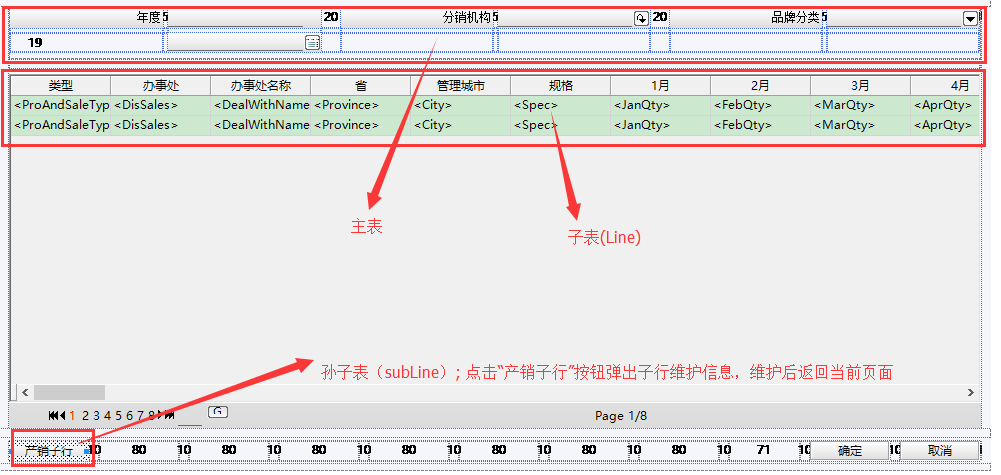
* 单卡多表（主子子关系）



## 单卡单表、单卡多表（主子孙）

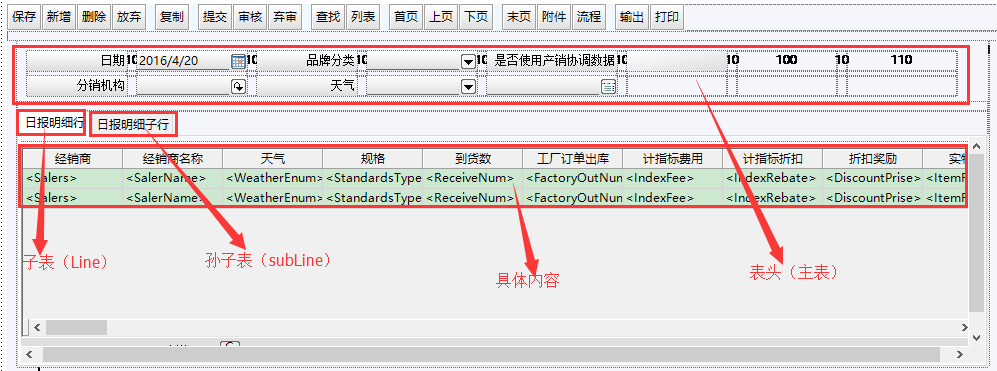
1. 采用上面单头、下面为表体；
2. 从页面左上角开始；
3. 单头、档案基本信息按三列处理；布局方式：从左到右，从上到下的方式；

* 单卡单表（主子孙关系）



适用场景：适用于子行有来源的场景，即子行是中间表的情况。

* 单卡多表（主子孙关系）



适用场景：只适用于主子孙关系，并且子行没有数据来源的场景，即子行不是中间表的情况。

## 表单显示

1. 新增、修改页面



关于精度：设置几位小数就显示几位，位数不够用0进行补齐；

1. 详情查看页面：

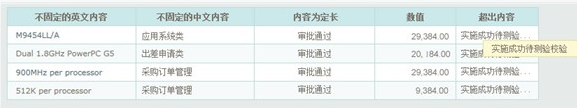


# 表格列表规范（重点）

1. 在写 <table> 互相嵌套时，严格按照的规范，对于单独的一个<table>来说，<table><tr>对齐，<td> 缩进一个TAB，<td> 中如果还有嵌套的表格，<table>也缩进TAB，如果<td>中没有任何嵌套的表格，</td> 结束标记应该与 <td> 处于同一行，不要换行。如：

|  |
| --- |
| <table> <tr>  <td><img src="images/icon.gif"></td> </tr> </table> |

1. 表头水平/垂直居中对齐；
2. 表格中内容如为定长（如：日期），则居中显示；如为不固定的中英文内容，则为居左显示；如为数值，则居右显示；
3. 保证表格的宽度不被挤压变形，对于不定长的内容，可以固定宽度。当超出此宽度后，内容不能自动换行，以省略号显示，光标停留后，详细内容再浮动显示；



1. 表格的嵌套控制在三层以内；
2. 列表的行被选中应以不同颜色突出。

# 分页页码规范

# 排版规范

1. 排版中我们经常会遇到需要进行首行缩进的处理，不要使用“&nbsp;”或者全角空格来达到效果，规范的做法是在样式表中定义 p { text-indent: 2em; } 然后给每一段加上 <p> 标记，注意，一般情况下，请不要省略 </p> 结束标记；
2. 原则上，禁止用 <img width=? height=?> 来人为干预图片显示的尺寸，而且建议<img> 标签中不要带上width 和height 两个属性，这是因为制作过程中，图片往往需要反复的修改，这样可以避免人为干预图片显示的尺寸，尽可能的发挥浏览器自身的功能；但是这样的一个副作用是当网页还未加载图片时，不会留出图片的站位大小，可能会造成网页在加载过程中抖动（如果图片是插在一个固定大小的表格里的，不会有这个现象），尤其是当图片的尺寸较大时，这种现象会很明显，所以当预料到这种会明显导致网页抖动的情况会发生时，请大家务必在最后给 <img>附上 width 和 height 属性；
3. 为了最大程度的发挥浏览器自动排版的功能，在一段完整的文字中请尽量不要使用<br> 来人工干预分段；
4. 不同语种的文字之间应该有一个半角空格，但避头的符号之前和避尾的符号之后除外汉字之间的标点要用全角标点，英文字母和数字周围的括号应该使用半角括号；
5. 所有的字号都应该用样式表来实现，禁止在页面中出现 <font size=?> 标记；
6. 请不要在网页中连续出现多于一个的“&nbsp;”也尽量少使用全角空格（英文字符集下，全角空格会变成乱码），空白应该尽量使用 text-indent, padding, margin, hspace, vspace 以及透明的gif 图片来实现；
7. 行距建议用百分比来定义，常用的两个行距的值是line-height:120%/150%；
8. 网站中的路径全部采用相对路径，一般链接到某一目录下的缺省文件的链接路径不必写全名，如我们不必这样：

|  |
| --- |
| <a href="aboutus/index.htm"> |

而应该这样：

|  |
| --- |
| <a href="aboutus/"> |

# 菜单栏规范

1. 菜单的标题需简明、有意义，尽量用一个词语；
2. 菜单标题字体间不要留空格，左对齐，只占一行；
3. 下拉菜单需根据菜单选项的含义进行分组；
4. 常用菜单应该另外设置区域放置快捷方式；
5. 菜单深度需控制在三层以内；
6. 菜单需按使用频度、逻辑顺序、使用顺序进行排序。

# 标题栏图标规范

1. 不同界面中的同一功能应该使用同样的图标和图片；
2. 图标、图片的色调、风格尽量保持一致；
3. 不同的目标采用不同的图标；
4. 图标尺寸，建议使用以下四种尺寸：48\*48像素，32\*32像素，24\*24像素，16\*16像素；
5. 图标上要有标注。

# 工具栏规范

1. 相同或相近功能的工具栏放在一起；
2. 工具栏中的每一个按钮要有及时提示信息；
3. 工具栏的图标能直观的代表要完成的操作；
4. 工具栏按钮应该设置工具提示 "Hint" 属性。

# 滚动条规范

1. 滚动条的长度要根据显示信息的长度或宽度能及时变换，以利于用户了解显示信息的位置和百分比；
2. 除非必要请避免水平滚动条，尽量使用垂直滚动条、加宽窗口、减小文本的宽度，或者使文本自动换行等。

# CSS样式书写规范

## 文件命名规范：

全局样式 global.css common.css

框架布局 layout.css

字体样式 font.css

链接样式 link.css

## 常用clsss/id命名规范

应尽量以常见英文单词为准，做到通俗易懂，并在适当的地方加以注释。对于二级class/id命名，采用驼峰命名方式，如：searchInput、searchIcon等。

## 结构与样式分离

在页面里只写入文档的结构，而将样式写于css文件中，通过外部调用css来实现结构与样式分离。

## 文档的结构化书写

页面和css文档都采用结构化的书写方式，逻辑清晰易于阅读。块级的结束标签如</div>要另起一行，其他结束便签可以与开始标签同一行。层级统一用4个空格缩进。css中的每个属性占一行。如：

|  |
| --- |
| <div id="mainMenu">  <ul>  <li><a href="#">首页</a></li>  <li><a href="#">简介</a></li>  <li><a href="#">关于</a></li>  </ul> </div> |

|  |
| --- |
| /\*===导航===\*/ #mainMenu {  width:100%;  height:33px;  background:url(images/mainMenu\_bg.jpg) repeat-x; }  #mainMenu ul li {  float:left;  line-height:30px;  cursor:pointer; } /\*===导航结束===\*/ |

## 注释书写规范

行间注释直接写于行的后面，如：

|  |
| --- |
| .search {  border:1px solid #fff; /\*搜索输入边框\*/  background:url(../images/icon.gif) no-repeat #333; /\*搜索框背景\*/ } |

整段注释分别在开始和结束的地方加入，占一行。如：

|  |
| --- |
| /\*===搜索条===\*/ .search { … } /\*===搜索条结束===\*/ |

对于页面结构层级较深的情况要整段注释，如：

|  |
| --- |
| <!—中间区域 --> <div id=”middle”>  <!—左侧 -->  <div>  <div>  …//n行  </div>  </div>  <!—左侧结束 --> </div> <!—中间区域结束 --> |

## 样式属性代码优化缩写

不同类有相同属性及属性值的缩写可以合并。如：

|  |
| --- |
| #mainMenu {  background:url(../images/bg.gif);  border:1px solid #333;  width:100%;  height:33px;  overflow:hidden; }  #subMenu {  background:url(../images/bg.gif);  border:1px solid #333;  width:100%;  height:20px;  overflow:hidden; } |

合并缩写为：

|  |
| --- |
| #mainMenu, #subMenu {  background:url(../images/bg.gif);  border:1px solid #333;  width:100%;  overflow:hidden; }  #mainMenu {  hight:33px; }  #subMenu {  height:20px; } |

同一属性可以合并多个值：

|  |
| --- |
| .search {  background-color:#333;  background-image:url(../images/icon.gif);  background-repeat:no-repeat;  background-position:50% 50%; } |

合并后：

|  |
| --- |
| .search {  background:#333 url(../images/icon.gif) no-repeat 50% 50%; } |

## CSS各浏览器解决不兼容问题

利用css的hack可以调整各种浏览器间对css解析的差异性。在没有必要的情况下不要写hack来调整；

IE6、IE7、Firefox之间的兼容写法：

写法一：

IE都能识别\*；标准浏览器如FF不能识别\*

IE6能识别\*，但不能识别!important

IE7能识别\*，也能识别!important

FF不能识别\*，但能识别!important

例如：

|  |
| --- |
| .search {  background-color:#333; /\*三者皆可\*/  \*background-color:#666; !important; /\*仅IE7\*/  \*background-color:#999; /\*仅IE6\*/ } |

一般书写顺序为 FF，IE7，IE6

写法二：

IE6可识别”\_”，而IE7和FF都不能识别。如：

|  |
| --- |
| .search {  \_background-color:#eee; } |

写法三：

\*+html与\*html是IE特有的标签，FF暂不支持。如：

|  |
| --- |
| \*html .search {  \_background-color:#eee; } \*+html .search {  \_background-color:#eee; } |

# 其他规范

所有的javascript脚本尽量采取外部调用；

Web页面编码一律统一用UTF-8编码；

所有连接使用相对路径；

所有的文件、目录、图片全部以小写字母命名。

开发期间，使用chrome浏览器作为开发及调试工具，需保证在chrome浏览器下兼容。