IEM项目管理系统

前端编程规范

**作者：**秦丽琴

文档归类: 文思海辉机密

修订历史

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **版本号** | **版本发布日期** | **作者** | **审核人** | **批准人** | **受影响的部分和变更总结** |
| 0.1 | 207-11-10 | 秦丽琴 |  |  | 初稿 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目 录

[1. 通用 1](#_Toc296534814)

[1.1. 字符编码 1](#_Toc296534815)

[1.2. 文件的命名 1](#_Toc296534816)

[1.3. 缩进 1](#_Toc296534817)

[1.4. 目录结构 1](#_Toc296534818)

[2. HTML 3](#_Toc296534819)

[2.1. DOCTYPE 3](#_Toc296534820)

[2.2. 标签 3](#_Toc296534821)

[2.3. 标签的嵌套 3](#_Toc296534822)

[2.4. 行内CSS 4](#_Toc296534823)

[2.5. 自定义属性 4](#_Toc296534824)

[2.6. 图片 4](#_Toc296534825)

[2.7. 闭合浮动元素 4](#_Toc296534826)

[3. CSS 5](#_Toc296534827)

[3.1. CSS Code Style 5](#_Toc296534828)

[3.2. 选择器的使用 6](#_Toc296534829)

[3.3. 兼容问题 6](#_Toc296534830)

[3.4. !important 6](#_Toc296534831)

[4. JavaScript 7](#_Toc296534832)

[4.1. JavaScript Code Style 7](#_Toc296534833)

[4.2. 文档注释 7](#_Toc296534834)

[4.3. JavaScript文件引用在页面中的位置 8](#_Toc296534835)

[4.4. 避免在HTML中直接嵌入JavaScript 8](#_Toc296534836)

[4.5. 减少全局变量 8](#_Toc296534837)

[5. 附录 9](#_Toc296534838)

[5.1. 废弃的HTML标签及属性 9](#_Toc296534839)

[5.2. HTML嵌套规则 9](#_Toc296534840)

[5.3. CSS Sprites 12](#_Toc296534841)

[5.4. 浮动元素的闭合 12](#_Toc296534842)

[5.5. CSS选择器兼容性一览表 14](#_Toc296534843)

[5.6. CSS Hack 16](#_Toc296534844)

[5.7. IE的条件注释 16](#_Toc296534845)

[5.8. JavaScript的阻塞 17](#_Toc296534846)

[5.9. 无干扰的JavaScript 17](#_Toc296534847)

# 通用

## 字符编码

所有文件统一使用UTF-8编码。

## 文件的命名

文件名应该全部小写，多个单词以下划线“\_”分开。

## 缩进

缩进统一为4个空格。禁止使用Tab字符缩进。

## 目录结构

|  |  |  |
| --- | --- | --- |
| \_Root |  |  |
|  | cn | 存放中文HTML文件 |
|  | en | 存放英文HTML文件 |
|  | flash | 存放Flash文件 |
|  | images | 存放图片文件 |
|  | imagestudio | 存放PSD源文件 |
|  | flashstudio | 存放flash源文件 |
|  | inc | 存放include文件 |
|  | library | 存放DW库文件 |
|  | media | 存放多媒体文件 |
|  | project | 存放工程项目资料 |
|  | temp | 存放客户原始资料 |
|  | js | 存放JavaScript脚本 |
|  | css | 存放CSS文件 |

# HTML

## DOCTYPE

HTML页面页首必须填写DOCTYPE。DOCTYPE使用XHTML 1.0 Transitional。如下：

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

## 标签

按照XHTML规范，所有标签名及属性名必须全小写，属性值以双引号包裹，属性的等号两边禁止添加空格。所有标签必须闭合。例如： “<br>”必须写作“<br />”。非替换元素禁止直接结束，例如：<div></div>禁止写成<div />。

所有被废弃的标签禁止使用，例如：<b>，<i>，<font>等。具体参考附录5.1

## 标签的嵌套

标签的嵌套必须按照XHTML规范嵌套，例如：<li>必须在<ol>或<ul>标签中，不能脱离这两个标签。

区分块级元素及行内元素。行内元素禁止嵌套块级元素。

关于HTML标签的嵌套规则，请参考附录5.2

## 行内CSS

尽量减少行内CSS的使用，除一些类似规定表格栏位宽度等比较具有特殊性的位置，禁止在元素内使用行内CSS。行内CSS示例：

<div style=”background: black;”></div>

## 自定义属性

所有自定义属性建议以“data-”，多个单词以“-”分隔，例如“data-valid-type”，“data-field-name”等。该目的是为了兼容之后的HTML5，并且jQuery 1.6对此也有着更好的支持。

## 图片

对内容型的图片和背景型的图片进行区分。对于网站LOGO，按钮图片以及边框样式等均属于背景型图片，而商品的图片，注册用户的头像等则属于内容型的图片。背景型图片应尽量避免使用<img>标签显示在页面中，而是应该采用CSS，背景型图片应采用CSS Sprite技术，将零散的图片合成为一张图片，以减少请求。关于CSS Sprit技术，请参考5.3

图片的命名应该全小写，多个单词以下划线“\_”分隔。

## 闭合浮动元素

对于设置了float的元素，一定要对float进行闭合。具体参考附录5.4

# CSS

## CSS Code Style

### 样式命名

对于类的命名，采用全小写，多个单词以“-”分隔。例如：.nav，.panel-title等。命名采用英文单词，禁止使用汉语拼音，尽量避免使用数字。

### 属性

每条属性独占一行，属性名冒号后空一格。行结束以分号“;”结尾。

属性顺序建议按照一下顺序书写：

1. 布局属性（position，display，float，clear，display等）
2. 盒模型属性（margin，padding，width，height，border等）
3. 文本属性（font，text-decoration，text-align，line-height等）
4. 其他属性（background，color，opacity等）

### 代码示例

table.standard-table {

width: 100%;

border-spacing: 1px;

background: black;

}

## 选择器的使用

尽量缩小样式的应用范围。例如有个类样式“.list-table”用于table标签，如果该样式不会应用在其他标签上的话，那么该类样式应写作“table.list-table”，避免使用通配符。

避免使用ID选择器，样式应该尽量选择类选择器，ID有可能绑定着页面的业务逻辑，有可能发生改变，因此尽量避免使用ID选择器应用样式。

避免使用IE6不兼容的选择器，如子选择器“>”以及属性选择器“[type=’checkbox’]”等。关于CSS选择器的兼容性，请参考附录5.5

## 兼容问题

尽量避免使用CSS Hack，对于IE系列的浏览器兼容性问题，采用条件注释加载对应的CSS，覆盖默认样式的方式解决。有关CSS Hack技术及IE的条件注释技术具体请参考附录5.6和附录5.7。

## !important

!important会使样式属性优先级变为最高，在使用时需要一定的谨慎，建议一般情况下不要使用该关键字。

# JavaScript

## JavaScript Code Style

* 变量与函数名采用驼峰式命名
* 常量使用全大写，多个单词以下划线“\_”分隔
* 关键字后空一格
* 等号两边各空一格
* 一行语句结束时必须添加“;”
* if-else语句块必须添加“{}”，即使只有一行语句。
* 类中的方法应写入类的prototype中，私有方法以下划线“\_”开头。

示例：

var foo = function (params) {

if (params.length > 0) {

doSomthing(params);

}

};

function Student(name, gender) {

this.name = name;

this.gender = gender;

}

Student.prototype.study = function (course) {

doSomething();

};

## 文档注释

JavaScript的文档注释采用JSDoc文档注释，所有公共函数、类都应该添加文档注释。关于JSDoc，具体请参看<http://code.google.com/p/jsdoc-toolkit/w/list>。

JSDoc示例：

*/\*\**

*\** ***@function*** *为一个checkbox添加对一组checkbox的全选功能*

*\** ***@param*** *{String|Element} checkbox 目标element对象或id*

*\** ***@param*** *{String} checkBoxName 受控checkbox的name*

*\** ***@param*** *{String} [form] 受控checkbox所在表单的name或id，可选*

*\*/*

function addCheckAllFunction(checkbox checkboxName, form) {

}

## JavaScript文件引用在页面中的位置

尽可能的将JS的引入位置放到body的结束标签之前，以减少被JavaScript载入时阻塞页面的情况。有关JavaScript阻塞的问题，请参考附录5.8。

## 避免在HTML中直接嵌入JavaScript

尽量避免在html中直接嵌入JavaScript。例如：

<button id=”okButton” type=”button” onclick=”ok();”>Button</button>

建议采取无干扰的JavaScript方式，在JavaScript中注册该事件。有关无干扰的JavaScript技术，请参考附录5.9。

## 减少全局变量

尽量不要在全局作用域下声明变量或者函数。他们应该被组织在各自的业务命名空间下，或在一个匿名函数中。

对于必须使用的全局变量，变量名应以“$”开头。

# 页面风格

## 通用元素

1. 大按钮
2. 普通按钮
3. 小按钮

## 通用查询页

5.2.1页面结构

(元素上下间距30px,button左右间距20px)



.queryContent\_inner

.titleBar



将整体页面内容包在queryContent\_wrapper中，标题部分为titleBar，具体表单和表格部分为queryContent\_inner.

5.2.2 标题右侧按钮

按钮包含在.titleBtn\_wrapper中

5.2.2.1文字按钮

1510887974(1)

<div class="titleBar">

<span class="title-text">加班管理</span>

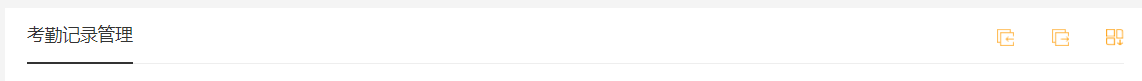
<div class="titleBtn\_wrapper">

<el-button @click="handleAdd">删除</el-button>

<el-button type="primary" @click="handleAdd">新增</el-button>

</div>

</div>



5.2.2.3导入导出图标按钮

每个子按钮加.imExport-btn-item



5.2.3高级查询部分

5.2.3.1表单结构

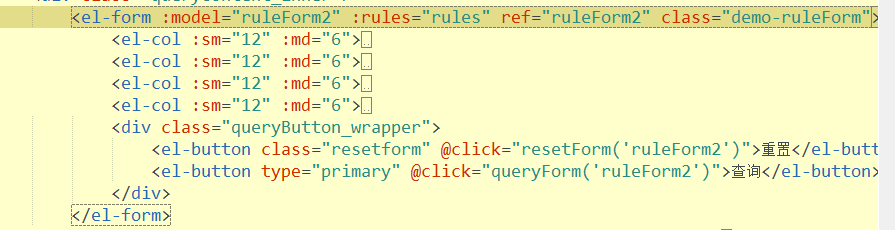


form、table、pagination放在queryContent\_inner内。

（el-form 不用设置label-width，单项label过长可单独设置label-width）

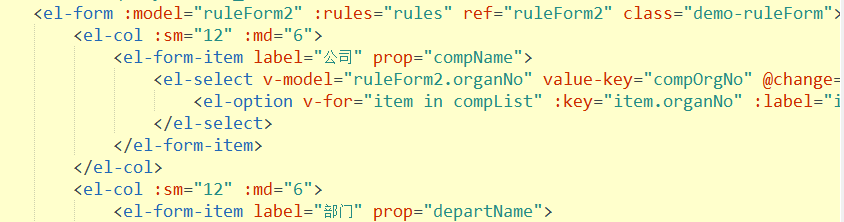


重置、查询按钮放在.queryButton\_wrapper内



5.2.3.1响应式布局

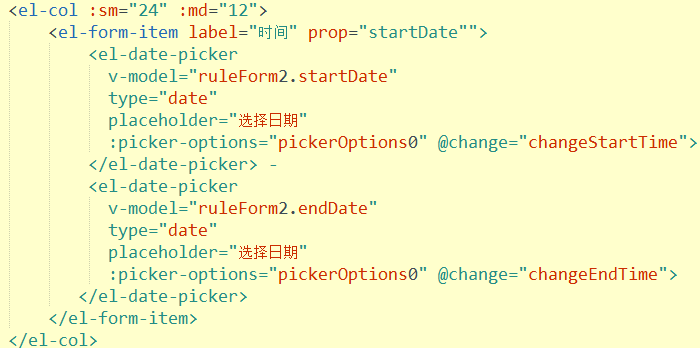
5.2.3.1.1 el-col中使用xs、sm、md、lg 和 xl等来代替span，如下，中屏为4列，小屏如下为2列



5.2.3.1.2 查询框有日期选择如下时

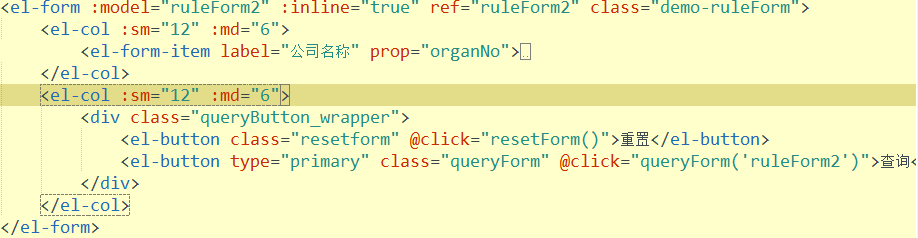


中屏为2列，小屏如下为1列



查询条件比较少，与按钮一行内显示时，queryButton\_wrapper包含在el-col里面



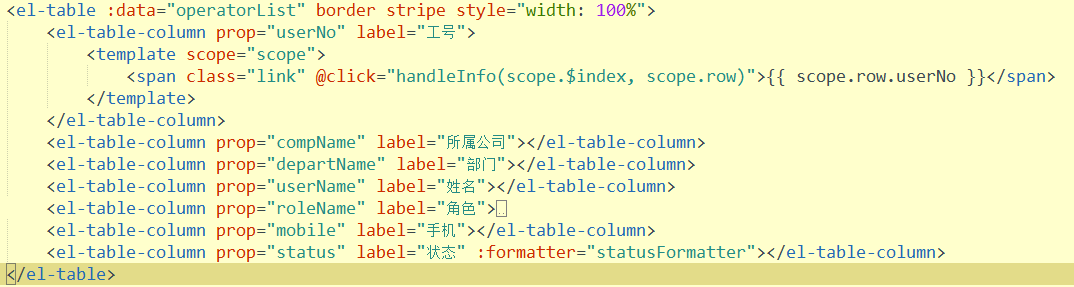


5.2.4表格部分



5.2.4.1 表格结构

有边框、斑马纹的el-table，添加border stripe 属性



5.2.4.2自定义列

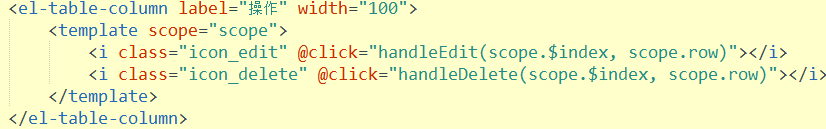
5.2.4.2.1自定义文本

通过 Scoped slot 可以获取到 row, column, $index 和 store（table 内部的状态管理）的数据如上图所示 添加link类

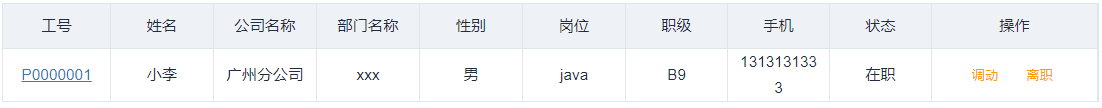
5.2.4.2.1自定义button

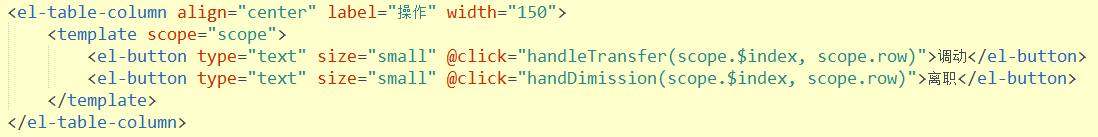
图标button





文字button

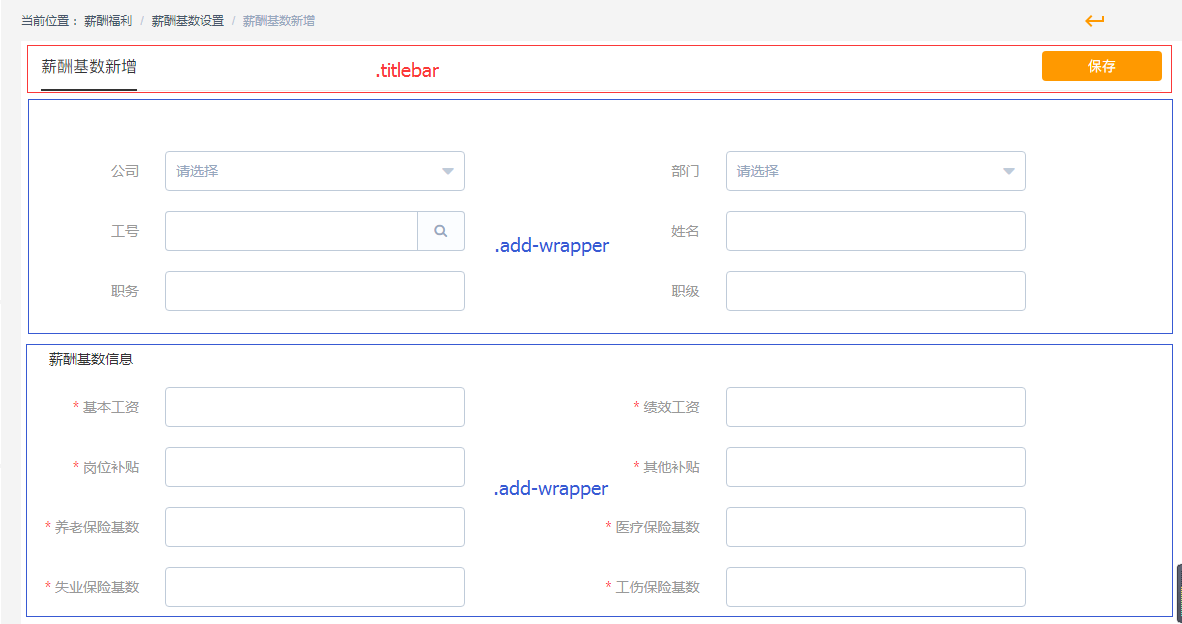


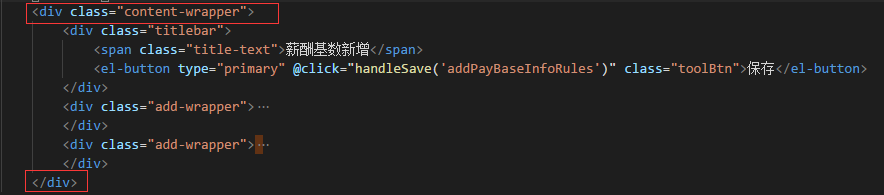


## 通用详情页

### 表单样式

#### 页面整体结构





将整体页面内容包在content-wrapper中，标题部分为titlebar，具体表单部分为add-wrapper.

#### 表单结构

1.将同一个小模块的form整体包在class=’’add-wrapper’’的div里



样式为

.add-wrapper {

overflow: hidden;

}

.add-wrapper form {

font-size: 0;

margin-top: 30px;

}

2.如一个form对应有小标题，如下



那么在对应的add-wrapper里，添加上item-title，如下：



样式代码：

.add-wrapper .item-title {

font-size: 14px;

height: 56px;

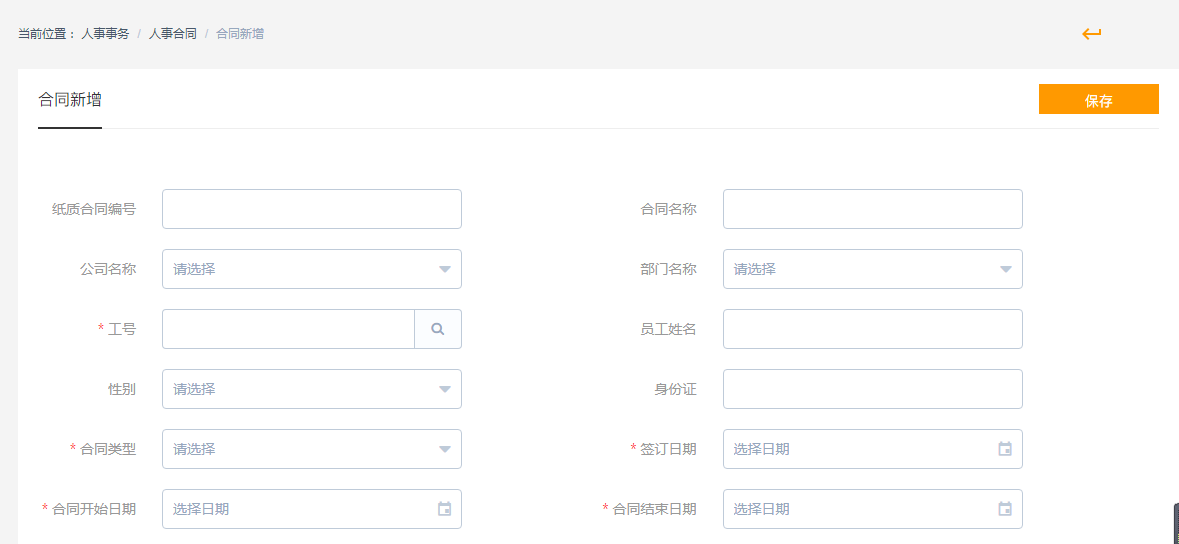
line-height: 56px;

padding-left: 8px;

}

#### 分栏布局

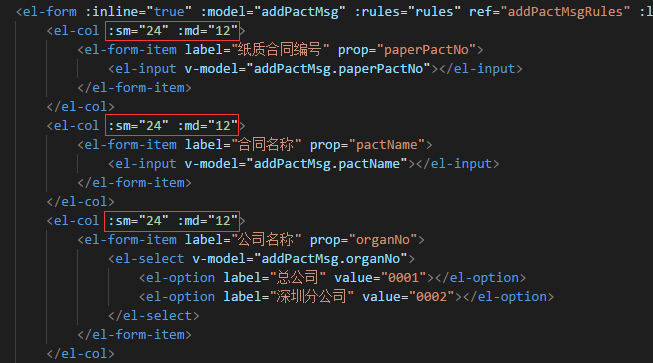
例：应为响应式布局，中屏如下为2列



小屏如下显示一列：

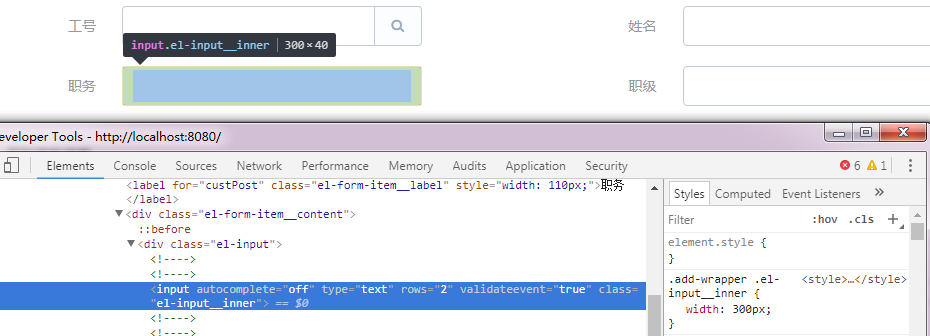


布局代码中应使用xs、sm、md、lg 和 xl等来代替span



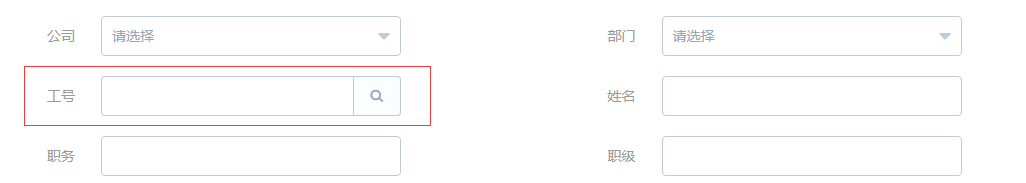
如有中屏需显示3列或更多，则需要结合实际需要来调整。

#### 普通输入框



宽高为：300px\*40px

#### 输入框带查询按钮



在button中加slot="append" icon="search"，代码示例：

<el-col :sm="24" :md="12">

<el-form-item label="工号" prop="userNo">

<el-input v-model="custInfo.userNo" @change="userNoChange">

<el-button slot="append" icon="search" @click="searchUserNo"></el-button>

</el-input>

</el-form-item>

</el-col>

#### 输入框带上传按钮



在该el-form-item体中，应有一个el-input输入框和el-upload框，数据model放在el-input中，文件事件处理放在el-upload中。el-upload请加上class="upload-demo"，el-upload里的el-button请加上class="uploadBtn"。

<el-col :sm="24" :md="12">

<el-form-item label="合同附件" prop="attachm">

       <el-input v-model="addPactMsg.attachm"></el-input>

       <el-upload class="upload-demo" :on-change="handleFileUpload" ref="upload" action="https://jsonplaceholder.typicode.com/posts/" :show-file-list="false" :auto-upload="false">

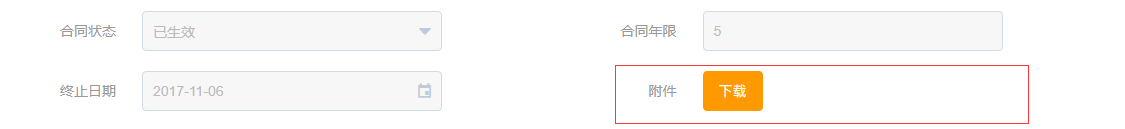
<el-button slot="trigger" size="small" type="primary" class="uploadBtn">选取文件</el-button>

</el-upload>

    </el-form-item>

</el-col>

#### 下载按钮



给el-button加上class="downloadBtn"

<el-col :sm="24" :md="12">

<el-form-item label="附件" prop="attachm">

       <el-button class="downloadBtn" @click="downloadFile">下载</el-button>

    </el-form-item>

</el-col>

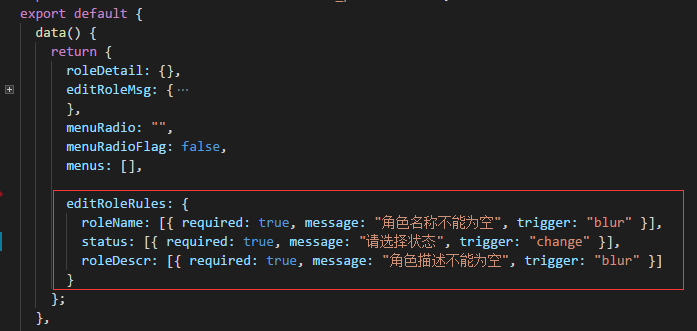
### 表单校验

例：需要校验如下图表单中角色名称、状态、描述等几项。

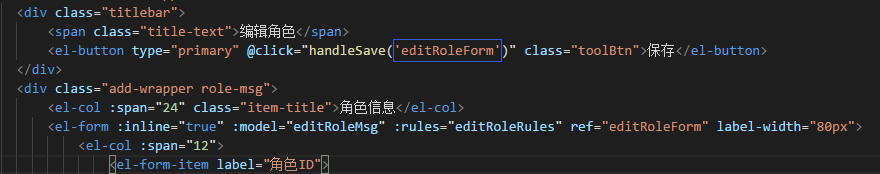


1.需在标签form中添加rules 属性传入约定的验证规则，示例:rules="editRoleRules"，并在各个需要做验证的对应form-item中添加prop，如红框所示。同时添加验证指向如ref="editRoleForm"。

1. 编辑同上对应的检验规则editRoleRules



1. 在触发校验的事件方法引用中加入步骤1中的ref指向"editRoleForm"



同时在事件方法的定义中，也做相应处理，代码如下：

handleSave(editRoleForm) {

this.$refs[editRoleForm].validate(valid => {

if (valid) {

// 校验成功的后续操作

} else {

// 校验失败的代码部分

console.log("error submit!!");

return false;

}

});

},

# 附录

## 废弃的HTML标签及属性

### 标签

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| applet | basefont | blackface | u | center | dir |
| font | i | isindex | layer | menu | noembed |
| s | shadow | strike |  |  |  |

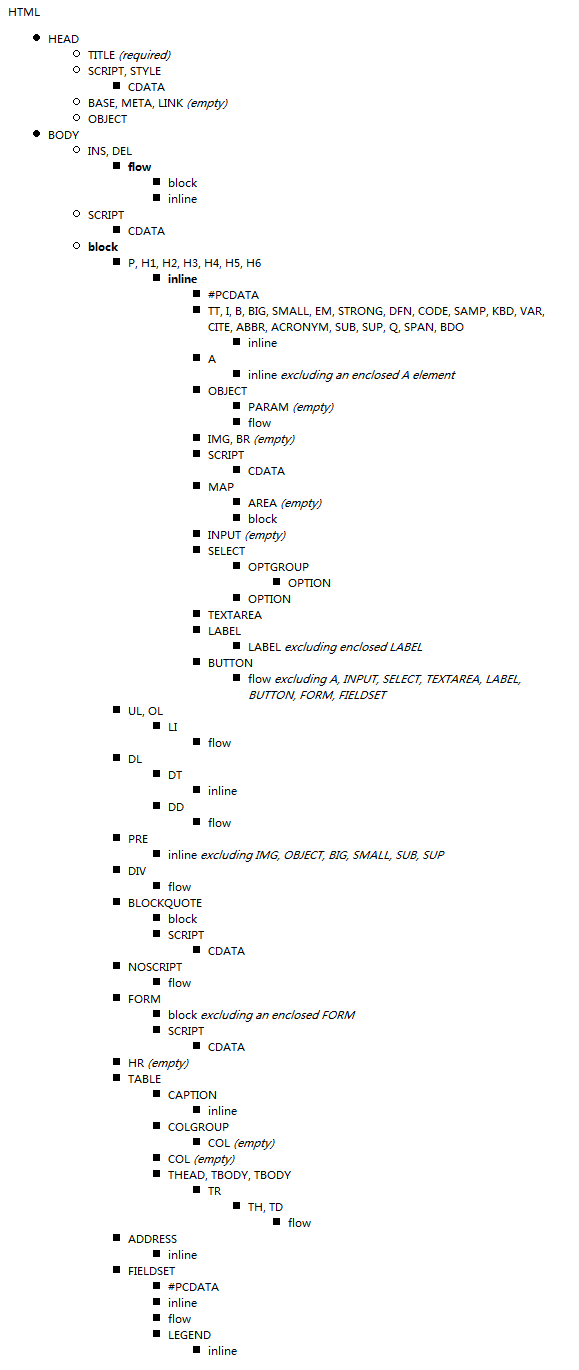
### 属性

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| alink | align | background | border | color | compact |
| face | height | language | noshade | nowrap | start |
| version | vlink | width |  |  |  |

## HTML嵌套规则

下图为HTML4 Strict的嵌套规则，XHTML与HTML4基本一致。下面是关于嵌套规则图的一些说明：

* 小写的单词表明一组或一系列 HTML 标签
* 每一项条目（标签）后都跟随一组标签列表，如果没有这个列表，那么表明该条目（标签）内部不允许包含任何标签。这意味着该条目内部只能包含纯文本内容（#PCDATA）。如果注明 (empty)，这意味着该条目内部不允许包含任何形式的内容。
* #PCDATA 的意思是“parsed character data”，即纯文本内容（不包括任何 HTML 标签，但是转义内容可以存在，比如 &auml; 和 &#228;）
* CDATA 的意思是“character data”，这意味着不包括转义内容的纯文本内容，详细内容可以参考CDATA Confusion
* excluding意即不得直接或者间接的包含所列的元素



## CSS Sprites

### 原理

CSS Sprites其实就是把网页中一些背景图片整合到一张图片文件中，再利用CSS的“background-image”，“background- repeat”，“background-position”的组合进行背景定位，background-position可以用数字能精确的定位出背景图片的位置。

### 优点

利用CSS Sprites能很好地减少了网页的http请求，从而大大的提高了页面的性能，这也是CSS Sprites最大的优点，也是其被广泛传播和应用的主要原因。

解决了网页设计师在图片命名上的困扰，只需对一张集合的图片上命名就可以了，不需要对每一个小元素进行命名，从而提高了网页的制作效率。

更换风格方便，只需要在一张或少张图片上修改图片的颜色或样式，整个网页的风格就可以改变。维护起来更加方便。

## 浮动元素的闭合

<div id=”floatA”></div>

<div id=”floatB”></div>

<div id=”NOTfloatC”></div>

这里的NOTfloatC并不希望继续平移，而是希望往下排 (其中floatA、floatB的属性已经设置为“float:left;”) 。原因是NOTfloatC没有设置float属性，必须将float标签闭合。在 <div class=”floatB”> <div class=”NOTfloatC”>之间加上 <div class=”clear”>这个div一定要注意位置，而且必须与两个具有float属性的div同级，之间不能存在嵌套关系，否则会产生异常。 并且将clear这种样式定义如下即可：

.clear{ clear:both;}

作为外部wrapper的div不要定死高度,为了让高度能自动适应，要在wrapper里面加上“overflow:hidden;”。当包含float的div的时候，高度自动适应在IE下无效，这时候应该触发IE的layout私有属性“zoom: 1;”，可以解决这个问题。这样就达到了兼容。

例如某一个wrapper如下定义：

.colwrapper { overflow: hidden; zoom:1; margin:5px auto;}

这里还有另一种方案添加闭合的方法，把需要闭合的div加上 class="clearfix" 即可,其定义如下：

.clearfix { display: inline-block; }

.clearfix:after {

content: ".";

display: block;

height :0px;

clear: both;

visibility: hidden;

}

/\* Hide from IE Mac \*/

.clearfix {display: block;}

/\* End hide from IE Mac \*/

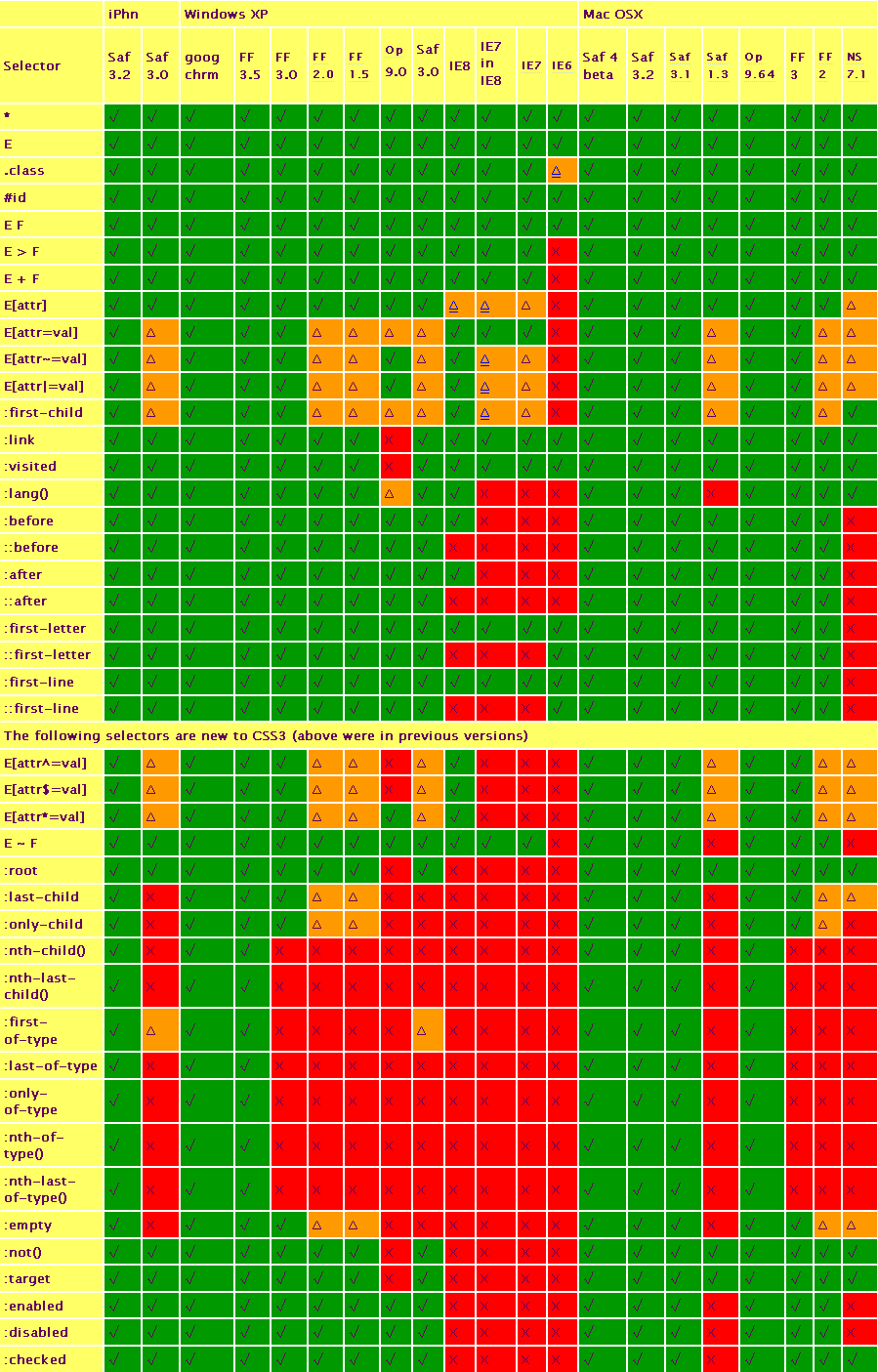
/\* end of clearfix \*/

或者这样设置：

.hackbox{ display: table; /\*将对象作为块元素级的表格显示\*/ }

## CSS选择器兼容性一览表

下图为各CSS选择器在不同系统中不同浏览器的兼容状况。其中绿色代表完全支持，橙色代表部分支持，红色代表完全不支持。



## CSS Hack

CSS Hack是在标准CSS没办法兼容各浏览器显示效果时才会用上的补救方法，利用了各个浏览器对CSS的支持和解析结果不一样的特点，例如： IE6能识别下划线"\_"和星号"\*"，IE7能识别星号"\*"，但不能识别下划线"\_"，而Firefox两个都不能识别。书写顺序一般是将识别能力强的浏览器的CSS写在后面。比如：

div{

background: green; /\* for firefox \*/

\*background: blue; /\* for IE7 \*/

\_background: red; /\* for IE6 \*/

}

这样在Firefox中看到的背景是绿色，在IE7中看到的是蓝色，而IE6中看到的是红色。

利用这种方式，可以解决多种由于IE独特的文档渲染方式造成的问题。

## IE的条件注释

### 介绍

条件注释是微软IE浏览器在HTML源代码中被解释的条件语句。最早出现于IE5，支持到IE8。条件注释用于在IE浏览器中显示或隐藏一段代码。

### 示例

<!--[if IE 6]>

<p>You are using Internet Explorer 6.</p>

<![endif]-->

<!--[if !IE]>-->

<p>You are not using Internet Explorer.</p>

<!--<![endif]-->

## JavaScript的阻塞

浏览器是按标签来解析的，每解析一段标签，就显示这段标签的内容。如果标签引入了外部文件（如图像），则会一边继续解析其他标签，一边继续下载该外部文件，直到载入完毕。

但浏览器在解析script标签时，由于JavaScript可能会改变浏览器内容，会停止所有页面元素的载入，直到这段script代码执行完毕。如果这个script标签引入了一个来自外部服务器的js文件，而这个服务器访问速度很慢，那么页面上其他元素便都得等到这个js文件载入并执行完毕才能继续载入。

## 无干扰的JavaScript

无干扰的 JavaScript 的首要目标就是保持 JavaScript 代码和 HTML 标记的分离。这种让内容分离于形为的方式，与将 CSS 放入样式表而保持内容与表现分离的方式如出一辙。为了实现这一目标，把所有的 JavaScript 代码放到外部文件中。

无干扰的 JavaScript 的第二个目标是它必须降低优雅性。脚本应该基于增加 HTML 的内容来构思和设计，但是，即使没有这些 JavaScript 代码，内容也应该能用。

无干扰的 JavaScript 的第三个目标是，它不能降低一个 HTML 页面的可访问性。