

CSS 规范手册

一 CSS 书写顺序

1.1 *

```
*{  
    /*显示属性*/  
    display  
    position  
    float  
    clear  
    cursor  
    ...  
  
    /*盒模型*/  
    margin  
    padding  
    width  
    height  
  
    /*排版*/  
    vertical-align  
    white-space  
    text-decoration  
    text-align  
    ...  
  
    /*文字*/  
    color  
    font  
    content  
  
    /*边框背景。为什么要把 border 和 background 放在最后的原因是修改的频率会较  
    之前的频繁，放在最后查看起来方便。*/  
    border  
    background  
}
```

1.2 属性顺序

下表顺序为从上到下，从左到右：

display || visibility

list-style : list-style-type || list-style-position || list-style-image
position

top || right || bottom || left

z-index

clear

float

width

max-width || min-width

height

max-height || min-height

overflow || clip

margin : margin-top || margin-right || margin-bottom || margin-left

padding : padding-top || padding-right || padding-bottom || padding-left

outline : outline-color || outline-style || outline-width

border

background : background-color || background-image || background-repeat ||
background-attachment || background-position

color

font : font-style || font-variant || font-weight || font-size || line-height || font-family

font : caption | icon | menu | message-box | small-caption | status-bar

text-overflow

text-align

text-indent

line-height

white-space

vertical-align

cursor

二 CSS 命名规则

2.1 文件命名规范

全局样式：global.css;

框架布局：layout.css;

字体样式：font.css;

链接样式：link.css;

打印样式: print.css;
主要的 master.css
专栏 columns.css
主题 themes.css
主要的 master.css
模块 module.css
基本共用 base.css
表单 forms.css
补丁 mend.css

2.2 页面结构

容器: container
页头: header
内容: content
页面主体: main
页尾: footer
栏目: column
页面外围控制整体布局宽度: wrapper
左右中: left right center

2.3 导航

导航: nav
主导航: mainbav
子导航: subnav
顶导航: topnav
边导航: sidebar
左导航: leftsidebar
右导航: rightsidebar
菜单: menu
子菜单: submenu
摘要: summary

2.4 功能

标志: logo
广告: banner
登陆: login
登录条: loginbar
注册: regsiter
搜索: search

功能区: shop
标题: title
加入: joinus
状态: status
按钮: btn
滚动: scroll
标签页: tab
文章列表: list
提示信息: msg
当前的: current
小技巧: tips
图标: icon
注释: note
指南: guild
服务: service
热点: hot
新闻: news
下载: download
投票: vote
合作伙伴: partner
友情链接: link
版权: copyright

Container div
#container
容器
Layout
#layout
布局
Header or banner div
#head、#header
页头部分
Footer div
#foot、#footer
页脚部分
Navigation list
#nav
主导航
Sub-navigation list
#subNav
二级导航
Menu
#menu
菜单

Sub Menu

#submenu

子菜单

Left or right side columns

#sidebar_a、#sidebar_b

左边栏或右边栏

Main div

#main

页面主体

Tag

#tag

标签

Message

#msg #message

提示信息

Tips

#tips

小技巧

Vote

#vote

投票

Friend Link

#friendlink

友情连接

Title

#title

标题

Summary

#summary

摘要

Search input

#searchInput

搜索输入框

Search output

#search_output

搜索输出和搜索结果相似

Search

#search

搜索

Search bar

#searchBar

搜索条

Search results

#search_results

搜索结果

Copyright information

#copyright

版权信息

brand

#branding

商标

branding-logo

#logo

LOGO

Site information

#siteinfo

网站信息

Copyright information etc.

#siteinfoLegal

法律声明

Designer or other credits

#siteinfoCredits

信誉

Join us

#joinus

加入我们

Partnership opportunities

#partner

合作伙伴

Services

#service

服务

Regsiter

#regsiter

注册

Arrow

arr/arrow

箭头

Little

#little

标题

Website map

#sitemap

网站地图

List

#list

列表

Home page

#homepage

首页

Sub page

subpage

二级页面子页面

Toolbar

#tool、#toolbar

工具条

Next pulls

#drop

下拉

Next pulls menu

#dorpmenu

下拉菜单

Status

#status

状态

Container div

#container

容器

Header or banner div

#header

页头部分

Main or global navigation div

#mainNav

主导航

Menu

#menu

菜单

Sub Menu

#submenu

子菜单

Left or right side columns

#sidebarA、#sidebarB

左边栏或右边栏

Main div

#main

页面主体

Content div

#content

内容部分

The main content area

#contentMain

主要内容区域

Footer div

#footer

页脚部分

Tag

#tag

标签

Message

#msg #message

提示信息

Tips

#tips

小技巧

Vote

#vote

投票

Friend Link

#friendlink

友情连接

Title

#title

标题

Summary

#summary

摘要

Sub-navigation list

#subNav

二级导航

Search input

#searchInput

搜索输入框

Search output

#searchOutput

搜索输出和搜索结果相似

Search

#search

搜索

Search results

#searchResults

搜索结果

Copyright information

#copyright

版权信息

brand

#branding

商标

branding-logo

#brandingLogo

LOGO

Site information

#siteinfo

网站信息

Copyright information etc.

#siteinfoLegal

法律声明

Designer or other credits

#siteinfoCredits

信誉

Join us

#joinus

加入我们

Partnership opportunities

#partner

合作伙伴

Services

#service

服务

Regsiter

#regsiter

注册

Status

#status

状态

Products

.products

产品

Products prices

.productsPrices

产品价格

Products description

.productsDescription

产品描述

Products review

.productsReview

产品评论

Editor's review

.editorReview

编辑评论

New release

.newsRelease

最新产品

Publisher

.publisher

生产商

Screen shot

.screenshot

缩略图

FAQ

.faqs

常见问题

Keyword

.keyword

关键词

Blog

.blog

博客

Forum

.forum

论坛

2.5 class 命名

(1)颜色：使用颜色的名称或者 16 进制代码

.red { color: red; }

.f60 { color: #f60; }

.ff8600 { color: #ff8600; }

(2)字体大小：直接使用“font+字体大小”作为名称

.font12px { font-size: 12px; }

.font9pt { font-size: 9pt; }

(3)对齐样式：使用对齐目标的英文名称

.left { float:left; }

.bottom { float:bottom; }

(4)标题栏样式,使用“类别+功能”的方式命名

.barnews { }

.barproduct { }

三 CSS 写作建议

1. 所有 xhtml 代码小写。
2. 属性的值一定要用双引号(“”)括起来，且一定要有值。
3. 每个标签都要有开始和结束，且要有正确的层次。
4. 空元素要有结束的 tag 或于开始的 tag 后加上 “/”。
5. 表现与结构完全分离，代码中不涉及任何的表现元素，如 style、font、bgColor、border 等。
6. <h1>到<h5>的定义，应遵循从大到小的原则，体现文档的结构，并有利于搜索引擎的查询。
7. 给每一个表格和表单加上一个唯一的、结构标记 id 。
8. 给重要的区块加上注释。
9. 给图片加上 alt 标签。
10. 所有的标签必须进行合理的嵌套。
11. 根元素前必须有元素，宣告使用那一种 DTD。
12. 根元素必须有 xmlns 属性来指定使用 <http://www.w3.org/1999/xhtml> 的 namespace。

3.1 常规书写规范及方法

1. 选择 DOCTYPE

XHTML 1.0 提供了三种 DTD 声明可供选择：

- 过渡的(Transitional):要求非常宽松的 DTD，它允许你继续使用 HTML4.01 的标识 (但是要符合 xhtml 的写法)。完整代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

- 严格的(Strict):要求严格的 DTD，你不能使用任何表现层的标识和属性，例如
。完整代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

- 框架的(Frameset):专门针对框架页面设计使用的 DTD，如果你的页面中包含有框架，需要采用这种 DTD。完整代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

理想情况当然是严格的 DTD，但对于大多数刚接触 Web 标准的设计师来说，过渡的 DTD(XHTML 1.0 Transitional)是目前理想选择(包括本站，使用的也是过渡型 DTD)。因为这种 DTD 还允许我们使用表现层的标识、元素和属性，也比较容易通过 W3C 的代码校验。

2. 指定语言及字符集

为文档指定语言：

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="en">
```

为了被浏览器正确解释和通过 W3C 代码校验，所有的 XHTML 文档都必须声明它们所使用的编码语言；如：

□ 常用的语言定义：

```
<meta http-equiv= " Content-Type " content= " text/html; charset=utf-8 " />
```

□ 标准的 XML 文档语言定义：

```
<?xml version= " 1.0 " encoding= " utf-8 " ?>
```

□ 针对老版本的浏览器的语言定义：

```
<meta http-equiv= " Content-Language " content= "utf-8 " />
```

为提高字符集，建议采用“utf-8”。

3. 调用样式表

外部样式表调用：

页面内嵌法：就是将样式表直接写在页面代码的 head 区。 如：

```
<style type="text/css"><!-- body { background : white ; color : black ; } --> </style>
```

外部调用法：将样式表写在一个独立的.css 文件中，然后在页面 head 区用类似以下代码调用。

```
<link rel="stylesheet" rev="stylesheet" href=style.css" type="text/css" media="all" />
```

在符合 web 标准的设计中，推荐使用外部调用法，可以不修改页面只修改.css 文件而改变页面的样式。如果所有页面都调用同一个样式表文件，那么改一个样式表文件，可以改变所有文件的样式。

4. 选用恰当的元素

根据文档的结构来选择 HTML 元素，而不是根据 HTML 元素的样式来选择。例如，使用 P 元素来包含文字段落，而不是为了换行。如果在创建文档时找不到适当的元素，则可以考虑使用通用的 div 或者是 span。

避免过渡使用 div 和 span。少量、适当的使用 div 和 span 元素可以使文档的结构更加清晰合理并且易于使用样式。

尽可能少地使用标签和结构嵌套，这样不但可以使文档结构清晰，同时也可以保持文件的小巧，在提高用户下载速度的同时，也易于浏览器对文档的解释及呈现。

5. 派生选择器

可以使用派生选择器给一个元素里的子元素定义样式，在简化命名的同时也使结构更加的清晰化，如：

```
.mainMenu ul li {background:url(images/bg.gif);}
```

6. 辅助图片用背影图处理

这里的”辅助图片”是指那些不是作为页面要表达的内容的一部分，而仅仅用于修饰、间隔、提醒的图片。将其做背影图处理，可以在不改动页面的情况下通过 CSS 样式来进行改动，如： #logo {background:url(images/logo.jpg) #FEFEFE no-repeat right bottom;}

7. 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 样式表来实现结构与样式的分离。

8. 文档的结构化书写

页面 CSS 文档都应采用结构化的书写方式，逻辑清晰易于阅读。如：

```

<div id="mainMenu">
<ul>
<li><a href=SPAN>" #" >首页</a></li>
<li><a href=SPAN>" #" >介绍</a></li>
<li><a href=SPAN>" #" >服务</a></li>
</ul>
</div>
/*=====主导航=====*/
#mainMenu {
width:100%;
height:30px;
background:url(images/mainMenu_bg.jpg) repeat-x;
}
#mainMenu ul li {
float:left;
line-height:30px;
margin-right:1px;
cursor:pointer;
}
/*=====主导航结束=====*/

```

9. 鼠标手势

在 XHTML 标准中, hand 只被 IE 识别, 当需要将鼠标手势转换为“手形”时, 则将“hand”换为“pointer”, 即“cursor:pointer;”

3.2 注释书写规范

1、行间注释

直接写于属性值后面, 如:

```

.search{
border:1px solid #fff;/*定义搜索输入框边框*/
background:url(..images/icon.gif) no-repeat #333;/*定义搜索框的背景*/
}

```

2、整段注释

分别在开始及结束地方加入注释, 如:

```

/*=====搜索条=====*/
.search {
border:1px solid #fff;
background:url(..images/icon.gif) no-repeat #333;
}
/*=====搜索条结束=====*/

```

3. 协助注释

非作者维护时所加入的表示修改时间、修改人等标识信息。

在区域注释或单行注释的基础上加上修改人和修改时间等信息。 例如(区域注释):

```
<!--==S 注释内容[修改人和修改时间]-->
```

```
<div>
```

```
...
```

```
</div>
```

```
<!--==E 注释内容[修改人和修改时间]-->
```

```
/*=S 注释内容[修改人和修改时间]*/
```

```
.class{
```

```
...
```

```
}
```

```
.class{
```

```
...
```

```
}
```

```
/*=E 注释内容[修改人和修改时间]*/
```

例如(单行注释):

```
<div>
```

```
<!-- 注释内容[修改人和修改时间]-->
```

```
...
```

```
</div>
```

```
.class{
```

```
/*注释内容[修改人和修改时间]*/
```

```
...
```

```
}
```

3.3 样式属性代码缩

1. 不同类有相同属性及属性值的缩写

对于两个不同的类,但是其中有部分相同甚至是全部相同的属性及属性值时,应对其加以合并缩写,特别是当有多个不同的类而有相同的属性及属性值时,合并缩写可以减少代码量并易于控制。例如:

```
#mainMenu {
```

```
    background:url(..images/bg.gif);
```

```
    border:1px solid #333;
```

```
    width:100%;
```

```
    height:30px;
```

```
    overflow:hidden;
```

```
}
```

```
#subMenu {
```

```
    background:url(..images/bg.gif);
```

```
    border:1px solid #333;
```

```
    width:100%;
```

```
    height:20px;
```

```
        overflow:hidden;
    }

```

两个不同类的属性值有重复之处，刚可以缩写为：

```
#mainMenu,#subMenu {
    background:url(..images/bg.gif);
    border:1px solid #333;
    width:100%;
    overflow:hidden;
}
#mainMenu {height:30px;}
#subMenu {height:20px;}
```

2、同一属性的缩写

同一属性根据它的属性值也可以进行简写，例如：

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

```

3、内外侧边框的缩写

在 CSS 中关于内外侧边框的距离是按照上、右、下、左的顺序来排列的，当这四个属性值不同时也可直接缩写。例如：

```
.btn {
    margin-top:10px;
    margin-right:8px;
    margin-bottom:12px;
    margin-left:5px;
    padding-top:10px;
    padding-right:8px;
    padding-bottom:12px;
    padding-left:8px;
}

```

则可缩写为：

```
.btn {
    margin:10px 8px 12px 5px;
    padding:10px 8px 12px 5px;
}

```

而如果当上边与下边、左边与右边的边框属性值相同时，则属性值可以直接缩写为两个。
例如：

```
.btn {  
    margin-top:10px;  
    margin-right:5px;  
    margin-bottom:10px;  
    margin-left:5px;  
}
```

缩写为:

```
.btn {margin:10px 5px;}
```

而当上下左右四个边框的属性值都相同时，则可以直接缩写成一个。例如：

```
.btn {  
    margin-top:10px;  
    margin-right:10px;  
    margin-bottom:10px;  
    margin-left:10px;  
}
```

缩写为:

```
.btn{margin:10px;}
```

4、颜色值的缩写:

当 RGB 三个颜色值数值相同时，可缩写颜色值代码。例如：

```
.menu { color:#ff3333;}
```

可缩写为:

```
.menu {color:#f33;}
```

3.4 hack 书写规范

因为不同浏览器对 W3C 标准的支持不一样，各个浏览器对于页面的解释呈现也不尽相同，比如 IE 在很多情况下就与 FF 存在 3px 的差距，对于这些差异性，就需要利用 CSS 的 hack 来进行调整，当然在没有必要的情况下，最好不要写 hack 来进行调整，避免因为 hack 而导致页面出现问题。

1. 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;

根据上述表达，同一类/ID 下的 CSS hack 可写为:

```
.searchInput {  
    background-color:#333;/*三者皆可*/  
    *background-color:#666 !important; /*仅 IE7*/  
    *background-color:#999; /*仅 IE6 及 IE6 以下*/  
}
```

一般三者的书写顺序为：FF、IE7、IE6。

□ 写法二:

IE6 可识别 “_”, 而 IE7 及 FF 皆不能识别, 所以当只针对 IE6 与 IE7 及 FF 之间的区别时, 可这样书写:

```
.searchInput {  
    background-color:#333;/*通用*/  
    _background-color:#666;/*仅 IE6 可识别*/  
}
```

□ 写法三:

*+html 与 *html 是 IE 特有的标签, Firefox 暂不支持。

```
.searchInput {background-color:#333;}  
*html .searchInput {background-color:#666;}/*仅 IE6*/  
*+html .searchInput {background-color:#555;}/*仅 IE7*/
```

□ 屏蔽 IE 浏览器:

select 是选择符, 根据情况更换。第二句是 MAC 上 safari 浏览器独有的。

```
*:lang(zh) select {font:12px !important;} /*FF 的专用*/  
select:empty {font:12px !important;} /*safari 可见*/
```

□ IE6 可识别:

这里主要是通过 CSS 注释分开一个属性与值, 注释在冒号前。

```
select { display /*IE6 不识别*/:none;}
```

□ IE 的 if 条件 hack 写法:

所有的 IE 可识别:

```
<![if IE]> Only IE <![end if]->
```

只有 IE5.0 可以识别:

```
<![if IE 5.0]> Only IE 5.0 <![end if]->
```

IE5.0 包换 IE5.5 都可以识别:

```
<![if gt IE 5.0]> Only IE 5.0+ <![end if]->
```

仅 IE6 可识别:

```
<![if lt IE 6]> Only IE 6- <![end if]->
```

IE6 以及 IE6 以下的 IE5.x 都可识别:

```
<![if gte IE 6]> Only IE 6/+ <![end if]->
```

仅 IE7 可识别:

```
<![if lte IE 7]> Only IE 7/- <![end if]->
```

2. 清除浮动

在 Firefox 中, 当子级都为浮动时, 那么父级的高度就无法完全的包住整个子级, 那么这时用这个清除浮动的 HACK 来对父级做一次定义, 那么就可以解决这个问题。

```
select:after {  
    content:"";  
    display:block;  
    height:0;  
    clear:both;  
    visibility:hidden;  
}
```

3.5 注意事项

1. id 和 class 命名采用该板块的英文单词或组合命名，并第一个单词小写，第二个单词首字母大写，如:newRelease（最新产品/new+Release）。

常用类的命名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。对于二级类/ID 命名，则采用组合书写的模式，后一个单词的首字母应大写：诸如“搜索框”则应命名为“searchInput”、“搜索图标”命名这“searchIcon”、“搜索按钮”命名为“searchBtn”……。

2. CSS 样式表各区块用注释说明。

3. 尽量使用英文命名原则。

4. 不用加中杠和下划线。

5. 尽量不缩写，除非一看就明白的单词。

6. 检查 HTML 元素是否有拼写错误、是否忘记结束标记。

即使是老手也经常会弄错 div 的嵌套关系。可以用 dreamweaver 的验证功能检查一下有无错误。

7. 检查 CSS 是否正确。

检查一下有无拼写错误、是否忘记结尾的 } 等。可以利用 CleanCSS 来检查 CSS 的拼写错误。CleanCSS 本是为 CSS 减肥的工具，但也能检查出拼写错误。

8. 确定错误发生的位置。

如果错误影响了整体布局，则可以逐个删除 div 块，直到删除某个 div 块后显示恢复正常，即可确定错误发生的位置。

9. 利用 border 属性确定出错元素的布局特性。

使用 float 属性布局一不小心就会出错。这时为元素添加 border 属性确定元素边界，错误原因即水落石出。

10. float 元素的父元素不能指定 clear 属性。

MacIE 下如果对 float 的元素的父元素使用 clear 属性，周围的 float 元素布局就会混乱。这是 MacIE 的著名的 bug，倘若不知道就会走弯路。

11. float 元素务必指定 width 属性。

很多浏览器在显示未指定 width 的 float 元素时会有 bug。所以不管 float 元素的内容如何，一定要为其指定 width 属性。

另外指定元素时尽量使用 em 而不是 px 做单位。

12. float 元素不能指定 margin 和 padding 等属性。

IE 在显示指定了 margin 和 padding 的 float 元素时有 bug。因此不要对 float 元素指定 margin 和 padding 属性（可以在 float 元素内部嵌套一个 div 来设置 margin 和 padding）。也可以使用 hack 方法为 IE 指定特别的值。

13. float 元素的宽度之和要小于 100%。

如果 float 元素的宽度之和正好是 100%，某些古老的浏览器将不能正常显示。因此请保证宽度之和小于 99%。

14. 是否重设了默认的样式。

某些属性如 margin、padding 等，不同浏览器会有不同的解释。因此最好在开发前首先将全体的 margin、padding 设置为 0、列表样式设置为 none 等。

15. float 的 div 一定要闭合。

例如：(其中 floatA、floatB 的属性已经设置为 float:left;)，程序代码：

```
<#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。

这段代码在 IE 中毫无问题，问题出在 FF。原因是 NOTfloatC 并非 float 标签，必须将 float 标签闭合。例如：

```
<#div class="floatB"></#div>
<#div class="NOTfloatC"></#div>
```

在上面代码之间加上：

```
<#div class="clear"></#div>
```

这个 div 一定要注意声明位置，一定要放在最恰当的地方，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 这种样式定义为如下即可：

```
.clear{ clear:both;}
```

此外，为了让高度能自动适应，要在 wrapper 里面加上 overflow:hidden;。

当包含 float 的 box 的时候，高度自动适应在 IE 下无效，这时候应该触发 IE 的 layout 私有属性用 zoom:1; 可以做到，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;
}
```

16. margin 加倍的问题。

设置为 float 的 div 在 ie 下设置的 margin 会加倍。这是一个 ie6 都存在的 bug。解决方案是在这个 div 里面加上 display:inline;，例如：

```
<#div id="imfloat"></#div>
```

相应的 CSS 为：

```
#IamFloat{
    float:left;
    margin:5px;/*IE 下理解为 10px*/
    display:inline;/*IE 下再理解为 5px*/
}
```

17. 关于容器的包涵关系。

很多时候，尤其是容器内有平行布局。例如，两、三个 float 的 div 时，宽度很容易出现问题。在 IE 中，外层的宽度会被内层更宽的 div 挤破。一定要用 Photoshop 或者 Firework 量取像素级的精度。

18. 关于高度的问题。

如果是动态地添加内容，高度最好不要定义。浏览器可以自动伸缩，然而如果是静态的内容，高度最好定好。

3.6 常见兼容问题

1. DOCTYPE 影响 CSS 处理。

2. FF: div 设置 margin-left, margin-right 为 auto 时已经居中，IE 不行。

3. FF:body 设置 text-align 时,div 需要设置 margin: auto(主要是 margin-left,margin-right) 方可居中。

4. FF: 设置 padding 后,div 会增加 height 和 width,但 IE 不会,故需要用 !important 多设一个 height 和 width。

5. FF: 支持 !important, IE 则忽略,可用 !important 为 FF 特别设置样式。

6. div 的垂直居中问题: vertical-align:middle; 将行距增加到和整个 DIV 一样高 line-height:200px; 然后插入文字,就垂直居中了。缺点是要控制内容不要换行。

7. cursor: pointer 可以同时 IE FF 中显示鼠标手指状, hand 仅 IE 可以。

8. FF: 链接加边框和背景色,需设置 display: block, 同时设置 float: left 保证不换行。参照 menubar, 给 a 和 menubar 设置高度是为了避免底边显示错位,若不设 height,可以在 menubar 中插入一个空格。

9. 在 mozilla firefox 和 IE 中的 BOX 模型解释不一致导致相差 2px 解决方法:

```
div{margin:30px!important;margin:28px;}
```

注意,这两个 margin 的顺序一定不能写反,据阿捷的说法!important 这个属性 IE 不能识别,但别的浏览器可以识别。所以在 IE 下其实解释成这样:

```
div{margin:30px;margin:28px}
```

重复定义的话按照最后一个来执行,所以不可以只写 margin:XXpx!important;

10. IE5 和 IE6 的 BOX 解释不一致。

IE5 下程序代码

```
div{width:300px;margin:0 10px 0 10px;}
```

div 的宽度会被解释为 300px-10px(右填充)-10px(左填充)最终 div 的宽度为 280px,而在 IE6 和其他浏览器上宽度则是以 300px+10px(右填充)+10px(左填充)=320px 来计算的。这时我们可以做如下修改:

```
div{width:300px!important;width/**/:340px;margin:0 10px 0 10px}
```

关于这个/**/, IE5 和 firefox 都支持,但 IE6 不支持。

11. ul 标签在 Mozilla 中默认是有 padding 值的,而在 IE 中只有 margin 有值所以先定义:

```
ul{margin:0;padding:0;}
```

注意书写格式的问题:一定要注意缩进。对齐。要有良好的编程习惯。