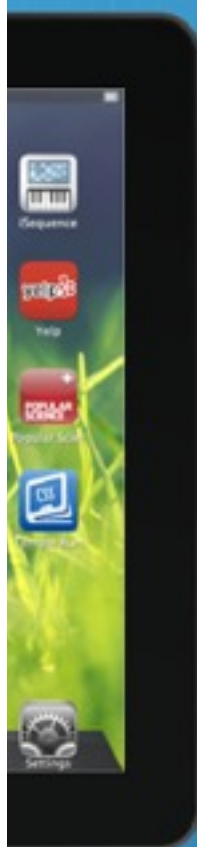




NEC
nec.netease.com

Nice & Easy Css IN Ibook





NEC优势

宗旨:致力于为前端开发人员提供高效率高质量的前端页面开发解决方案,提高多人协作效率,也为非专业人员提供快速制作网页的解决方案

特色功能

读规范,让你的代码更具智慧和美丽的方法

用框架,选择一份合适的HTML和CSS基础代码

找代码,在这里找到她,或给你启发,或她就是你的



nec
nec.netease.com



当前版本: 1.0 [检查更新](#)

联系我们

邮箱:nehzwd@163.com

地址:浙江省杭州市滨江区网商路599号网易大厦

网址:<http://nec.netease.com>

案例

这些网站使用了NEC,她们或遵照了NEC的规范或使用了NEC的框架和代码库内容。

LOFTER

iStyle | 爱搭配
网易旗下真人搭配社区

网易优惠券

网易摄影

网易闪印 sy.163.com
印像派旗下冲印软件

网易阅读

游戏江湖
hi.163.com

fm.163.com
网易闪电邮

目录

规范

- CSS规范
一系列规则和方法,帮助你架构并管理好样式
- HTML规范
一系列建议和方法,帮助你搭建简洁严谨的结构

框架

- css reset
重置样式,适用于需要兼容所有浏览器的网站,基于html4&css2
- css function
功能样式,适用于需要兼容所有浏览器的网站,基于html4&css2
- css animation
常见动画效果的集合,主要用于效果演示和参考,基于css3

- html template

包含完整头部信息的html模板, 基于html4

- html template2

包含完整头部信息的html模板, 基于html5

代码库

- 布局

一系列建议和方法,帮助你搭建简洁严谨的结构

- 模块

提供各种常见的网页内容模块,比如导航、菜单、幻灯、图文列表等

- 元件

提供各种常见的网页内容元件,比如按钮、标题、输入框等

- BUG

展示了浏览器的BUG及其解决方案,比如IE双倍边距、3像素问题等

- 技巧

提供各种较难或特殊设计的解决技巧,比如垂直居中、自适应等

目录

- 动画

使用CSS3实现的各种常见动画效果,比如淡入淡出、弹出等

插件

- Eclipse 代码插件

xml文件, 通过软件自带的模板功能导入, 自动代码提示

- Vim 代码插件

snippets文件, 通过结合其他插件实现, 自动代码提示



规范

你是否常常碰到以下问题：你总是看不懂他写的代码，或者读起来很吃力；你需要改他的代码却无从下手，或总是要去问他这里是什么改了会不会影响其他代码；你和他一起开发一个产品，你总是怕代码和他有冲突或互相影响；你的代码在多次维护任务之后变得越来越臃肿，越来越难以维护。

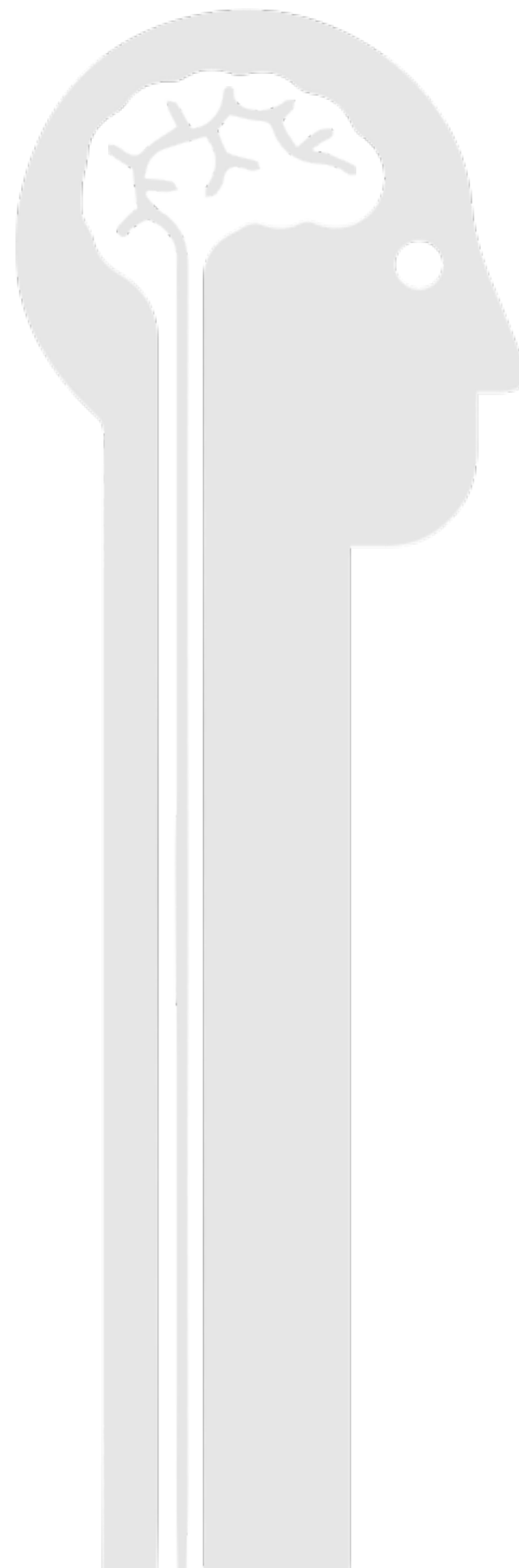
解决以上问题只需一种方法----读我们的规范！

●CSS规范

一系列规则和方法,帮助你架构并管理好样式

●HTML规范

一系列建议和方法,帮助你搭建简洁严谨的结构



规范 / CSS规范 / 分类方法

■ CSS文件的分类和引用顺序

通常，一个项目我们只引用一个CSS，但是对于较大的项目，我们需要把CSS文件进行分类。

我们按照CSS的性质和用途，将CSS文件分成“公共型样式”、“特殊型样式”、“皮肤型样式”，并以此顺序引用（按需求决定是否添加版本号）。

1. 公共型样式：包括了以下几个部分：“标签的重置和设置默认值”、“统一调用背景图和清除浮动或其他需统一处理的长样式”、“网站通用布局”、“通用模块和其扩展”、“元件和其扩展”、“功能类样式”、“皮肤类样式”。
2. 特殊型样式：当某个栏目或页面的样式与网站整体差异较大或者维护率较高时，可以独立引用一个样式：“特殊的布局、模块和元件及扩展”、“特殊的功能、颜色和背景”，也可以是某个大型控件或模块的独立样式。
3. 皮肤型样式：如果产品需要换肤功能，那么我们需要将颜色、背景等抽离出来放在这里。

```
1 <link href="assets/css/global.css" re
2 <link href="assets/css/index.css" rel
3 <link href="assets/css/skin.css" rel=
```

■ CSS内部的分类及其顺序

1. 重置 (reset) 和默认 (base) (tags)：消除默认样式和浏览器差异，并设置部分标签的初始样式，以减少后面的重复劳动！你可以根据你的网站需求设置！
2. 统一处理：建议在这个位置统一调用背景图（这里指多个布局或模块或元件共用的图）和清除浮动（这里指通用性较高的布局、模块、元件内的清除）等统一设置处理的样式！

3. 布局 (grid) (.g-)：将页面分割为几个大块，通常有头部、主体、主栏、侧栏、尾部等！
4. 模块 (module) (.m-)：通常是一个语义化的可以重复使用的较大的整体！比如导航、登录、注册、各种列表、评论、搜索等！
5. 元件 (unit) (.u-)：通常是一个不可再分的较为小巧的个体，通常被重复用于各种模块中！比如按钮、输入框、loading、图标等！
6. 功能 (function) (.f-)：为方便一些常用样式的使用，我们将这些使用率较高的样式剥离出来，按需使用，通常这些选择器具有固定样式表现，比如清除浮动等！不可滥用！
7. 皮肤 (skin) (.s-)：如果你需要把皮肤型的样式抽离出来，通常为文字色、背景色（图）、边框色等，非换肤型网站通常只提取文字色！非换肤型网站不可滥用此类！

功能类和皮肤类样式为表现化的样式，请不要滥用！以上顺序可以按需求适当调整。

分类的命名方法详见命名规则。

规范 / CSS规范 / 命名规则

■ 使用类选择器，放弃ID选择器

ID在一个页面中的唯一性导致了如果以ID为选择器来写CSS，就无法重用。

■ 分类的命名方法：使用单个字母+"-"为前缀

布局 (grid) (.g-)；模块 (module) (.m-)；元件 (unit) (.u-)；功能 (function) (.f-)；皮肤 (skin) (.s-)。

详情参见：分类方法中的“CSS内部的分类及其顺序”。

注意：

在你样式中的选择器总是要以上面五类开头，然后在里面使用后代选择器。

如果这五类不能满足你的需求，你可以另外定义一个或多个大类，但必须符合单个字母+"-"为前缀的命名规则，即 .x- 的格式。

特殊：.j-将被专用于JS获取节点，请勿使用.j-定义样式。

■ 后代选择器命名

- 约定不以单个字母+"-"为前缀且长度大于等于2的类选择器为后代选择器，如：.item为m-list模块里的每一个项，.text为m-list模块里的文本

部分：.m-list .item{}.m-list .text{}

- 一个语义化的标签也可以是后代选择器，比如：.m-list li{}
- **不允许单个字母的类选择器**出现，原因详见下面的“模块和元件的后代选择器的扩展类”。

通过使用后代选择器的方法，你不需要考虑他的命名是否已被使用，因为他只在当前模块或元件中生效，**同样的样式名可以在不同的模块或元件中重复使用**，互不干扰；在多人协作或者分模块协作的时候效果尤为明显！

后代选择器不需要完整表现结构树层级，**尽量能短则短**。

注意：

后代选择器**不要在页面布局中使用**，因为污染的可能性较大。

```
1  /* 这里的.itm和.cnt只在.m-list中有效 */
2  .m-list{margin:0;padding:0;}
3  .m-list .itm{margin:1px;padding:1px;}
4  .m-list .cnt{margin-left:100px;}
5  /* 这里的.cnt和.num只在.m-page中有效 */
6  .m-page{height:20px;}
7  .m-page .cnt{text-align:center;}
8  .m-page .num{border:1px solid #ddd;}
```


规范 / CSS规范 / 命名规则

■ 命名应简约而不失语义

```
1  /* 反对：表现化的或没有语义的命名 */
2  .m-abc .green2{}
3  .g-left2{}
4  /* 推荐：使用有语义的简短的命名 */
5  .m-list .wrap2{}
6  .g-side2{}
```

■ 相同语义的不同类命名

方法：直接加数字或字母区分即可（如：.m-lst、.m-lst2、.m-lst3等，都是列表模块，但是是完全不一样的模块）。

其他举例：.f-fw0、.f-fw1、.s-fc0、.s-fc1、.m-logo2、.m-logo3、u-btn、u-btn2等等。

■ 模块和元件的扩展类的命名方法

当A、B、C、...它们类型相同且外形相似区别不大，那么就以它们中出现率最高的做成基类，其他做成基类的扩展。

方法：+"-" + 数字或字母（如：.m-lst的扩展类为.m-lst-1、.m-lst-2等）。

补充：基类自身可以独立使用（如：class="m-lst"即可），扩展类必须基于基类使用（如：class="m-lst m-lst-2"）。

如果你的扩展类是表示不同状态，那么你可以这样命名：u-btn-dis, u-btn-hov, m-box-sel, m-box-hov等等，然后像这样使用：class="u-btn u-btn-dis"。

■ 模块和元件的后代选择器的扩展类

有时候模块内会有些类似的东西，如果你没有把它们做成元件和扩展，那么也可以使用后代选择器和扩展。

后代选择器：.m-login .btn{}

后代选择器扩展：.m-login .btn-1{}.m-login .btn-dis{}

规范 / CSS规范 / 命名规则

注意：

此方法用于类选择器，直接使用标签做为选择器的则不需要使用此命名方法。

为防止后代选择器的扩展类和大类命名规范冲突，后代选择器不允许使用单个字母。

比如：`.m-list .a{}`是不允许的，因为当这个`.a`需要扩展的时候就会变成`.a-bb`，这样就和大型的命名规范冲突。

■ 组合选择器有时可以代替扩展方法

有时候虽然两个同类型的模块很相似，但是我希望他们之间不要有依赖关系，也就是说你不希望使用扩展的方法，那么你可以通过合并选择器来设置共性的样式。

使用本方法的前提是：相同类型、功能和外观都相似，写在同一片代码区域方便维护。

```
1  /* 两个元件共性的样式 */
2  .u-tip1,.u-tip2{}
3  .u-tip1 .itm,.u-tip2 .itm{}
4  /* 在分别是两个元件各自的样式 */
5  /* tip1 */
6  .u-tip1{}
7  .u-tip1 .itm{}
8  /* tip2 */
9  .u-tip2{}
10 .u-tip2 .itm{}
```

■ 防止污染和被污染

当模块或元件之间互相嵌套，且使用了相同的标签选择器或其他后代选择器，那么里面的选择器就会被外面相同的选择器所影响。

所以，如果你的模块或元件可能嵌套或被嵌套于其他模块或元件，那么要慎用标签选择器，必要时采用类选择器，并注意命名方式，可以采用`.m-layer .layerxxx`、`.m-list2 .list2xxx`的形式来降低后代选择器的污染性。

规范 / CSS规范 / 代码格式

■ 选择器、属性和值使用小写

在xhtml标准中规定了所有标签、属性和值都小写，CSS也是如此。

■ 单行写完一个选择器定义

便于选择器的寻找和阅读，也便于插入新选择器和编辑，便于模块等的识别。去除多余空格，使代码紧凑减少换行。

如果有嵌套定义，可以采取内部单行的形式。

```
1  /* 单行定义一个选择器 */
2  .m-list li,.m-list h3{width:100px;padding:0 10px;}
3  /* 这是一个有嵌套定义的选择器 */
4  @media all and (max-width:600px){
5      .m-class1 .itm{height:17px;line-height:1.2;}
6      .m-class2 .itm{width:100px;overflow:hidden;}
7  }
8  @-webkit-keyframes showitm{
9      0%{height:0;opacity:0;}
10     100%{height:100px;opacity:1;}
11 }
```

■ 最后一个值也以分号结尾

通常在大括号结束前的值可以省略分号，但是这样做会对修改、添加和维护工作带来不必要的失误和麻烦。

■ 省略值为0时的单位

为节省不必要的字节同时也使阅读方便，我们将0px、0em、0%等值缩写为0。

```
1  .m-box{margin:0 10px;background-position:right top;}
```

■ 使用单引号

省略url引用中的引号，其他需要引号的地方使用单引号。

```
1  .m-box{background:url(bg.png);}
2  .m-box:after{content:'.';}
```

■ 使用16进制表示颜色值

除非你需要透明度而使用rgba，否则都使用#f0f0f0这样的表示方法，并尽量缩写。

规范 / CSS规范 / 代码格式

```
1 .m-box{color:#f00;background:rgba(0,0
```

position	margin	text-decoration
float	padding	vertical-align
clear	border	white-space
list-style	overflow	color
top	min-width	background

■ 依属性的重要性按顺序书写

只遵循横向顺序即可，先显示定位布局类属性，后盒模型等自身属性，最后是文本类及修饰类属性。

显示属性	自身属性	文本属性和其他修饰
display	width	font
visibility	height	text-align

```
1 .m-box{position:relative;width:600px;
```

■ 私有在前，标准在后

先写带有浏览器私有标志的，后写W3C标准的。

```
1 .m-box{-webkit-box-shadow:0 0 0 #000;
```

■ 注释格式：/* 注释文字 */

原则上每个系列的样式都需要有一个注释，言简意赅的表明名称、用途、注意事项等。

对选择器的注释统一写在被注释对象的上一行，对属性及值的注释写于分号后。

注释内容两端需空格，已确保即使在编码错误的情况下也可以正确解析样式。

规范 / CSS规范 / 代码格式

在必要的情况下，可以使用块状注释，块状注释保持统一的缩进对齐。

```
1  /* 块状注释文字
2     * 块状注释文字
3     * 块状注释文字
4     */
5  .m-list{width:500px;}
6  .m-list li{height:20px;overflow:hidde
7  .m-list li a{color:#333;}
8  /* 单行注释文字 */
9  .m-list li em{color:#666;}
```

■ 原则上不允许使用Hack

很多不兼容问题可以通过改变方法和思路来解决，并非一定需要Hack，根据经验你完全可以绕过某些兼容问题。

一种合理的结构和合理的样式，是极少会碰到兼容问题的。

由于浏览器自身缺陷，我们无法避开的时候，可以允许使用适当的Hack。

■ 统一Hack方法

统一使用“*”和“_”分别对IE7和6进行Hack。如下代码所示：

```
1  /* IE7会显示灰色#888，IE6会显示白色#fff，
2  .m-list{color:#000;*color:#888;_color
```

■ 建议并适当缩写值

“建议并适当”是因为缩写总是会包含一系列的值，而有时候我们并不希望设置某一值，反而造成了麻烦，那么这时候你可以不缩写，而是分开写。

当然，在一切可以缩写的情况下，请务必缩写，

它最大的好处就是节省了字节，便于维护，并使阅读更加一目了然。

缩写方法请查阅css手册。

■ 选择器顺序

请综合考虑以下顺序依据：

- 从大到小（以选择器的范围为准）
- 从低到高（以等级上的高低为准）
- 从先到后（以结构上的先后为准）
- 从父到子（以结构上的嵌套为准）

以下仅为简单示范：

规范 / CSS规范 / 代码格式

```
1  /* 从大到小 */
2  .m-list p{margin:0;padding:0;}
3  .m-list p.part{margin:1px;padding:1px}
4  /* 从低到高 */
5  .m-logo a{color:#f00;}
6  .m-logo a:hover{color:#fff;}
7  /* 从先到后 */
8  .g-hd{height:60px;}
9  .g-bd{height:60px;}
10 .g-ft{height:60px;}
11 /* 从父到子 */
12 .m-list{width:300px;}
```

■ 选择器等级

a = 行内样式style。

b = ID选择器的数量。

c = 类、伪类和属性选择器的数量。

d = 类型选择器和伪元素选择器的数量。

选择器	等级 (A,B,C,D)
style=""	1,0,0,0
#wrapper #content{}	0,2,0,0
#content .dateposted{}	0,1,1,0
div#content{}	0,1,0,1
#content p{}	0,1,0,1
#content{}	0,1,0,0
p.comment .dateposted{}	0,0,2,1
div.comment p{}	0,0,1,2
.comment p{}	0,0,1,1
p.comment{}	0,0,1,1
.comment{}	0,0,1,0
div p{}	0,0,0,2
p{}	0,0,0,1

规范 / CSS规范 / 优化方案

■ 值缩写

缩写值可以减少CSS文件大小，并增加可读性和可维护性。

但并非所有的值都必须缩写，因为当一个属性的值缩写时，总是会将所有项都设置一遍，而有时候我们不希望设置值里的某些项。

```
1  /* 比如我们用下面这个样式来让某个定宽的容器水平居中，我们要的只是left:
2  * 而top和bottom不是这个样式要关心的（如果设置了反倒会影响其他样式在
3  * 所以这时我们就不需要缩写
4  */
5  .f-mgha{margin-left:auto;margin-right:auto;}
6  /* 比如下面这个模块的样式设置，我们确实需要设置padding的所有项，于是
7  .m-link{padding:6px 12px;}
```

常用的缩写方法请参见代码格式。

■ 避免耗性能的属性

以下所列举的属性可能造成渲染性能问题。不过有时候需求大于一切.....

```
1  /* expression */
2  .class{width:expression(this.width>100?'100px':'auto');}
3  /* filter */
4  .class{filter:alpha(opacity=50);}
```

■ 选择器合并

即CSS选择器组合，可以一次定义多个选择器，为你节省很多字节和宝贵时间。

通常我们会将定义相同的或者有大部分属性值相同（确实是因为相关而相同）的一系列选择器组合到一起（采用逗号的方法）来统一定义。

规范 / CSS规范 / 优化方案

```
1  /* 以下对布局类选择器统一做了清除浮动的操作 */
2  .g-hd:after,.g-bd:after,.g-ft:after{display:block;visibility:
3  .g-hd,.g-bd,.g-ft{zoom:1;}
4  /* 通常background总是会占用很多字节，所以一般情况下，我们都会这样统一
5  .m-logo,.m-help,.m-list li,.u-tab li,.u-tab li a{background:u
6  .m-logo{background-position:0 0;}
7  /* 以下是某个元件的写法，因为确实很多元素是联动的或相关的，所以采用了结
8  .u-tab li,.u-tab li a{display:inline;float:left;height:30px;
9  .u-tab li{margin:0 3px;}
10 .u-tab li a{padding:0 6px;}
```

■ 背景图优化合并

图片本身的优化：

- 图像质量要求和图像文件大小决定你用什么格式的图片，用较小的图片文件呈现较好的图像质量。
- 当图片色彩过于丰富且无透明要求时，建议采用jpg格式并保存为较高质量。
- 当图片色彩过于丰富又有透明或半透明要求或阴影效果时，建议采用png24格式，并对IE6进行png8退化（或在不得已情况下使用滤镜）。
- 当图片色彩不太丰富时无论有无透明要求，请采用png8格式，大多数情况下建议采用这种格式。

- 当图片有动画时，只能使用gif格式。
- 你可以使用工具对图片进行再次压缩，但前提是会影响色彩和透明。

多张图片的合并：

- 单个图标之间必须保留空隙，空隙大小由容器大小及显示方式决定。这样做的好处是既考虑了“容错性”又提高了图片的可维护性。
- 图标的排列方式，也由容器大小及显示方式决定。排列方式分为以下几种：横向排列（容器宽度有限）、纵向排列（容器高度有限）、斜线排列（容器宽高不限）、靠左排列（容器背景居左）、靠右排列（容器背景居右）、水平居中排列（容器背景水平居中）、垂直居中排列（容器背景垂直居中）。
- 合并后图片大小不宜超过50K，建议大小在20K-50K之间。
- 为保证多次修改后的图片质量，请保留一份PSD原始图，修改和添加都在PSD中进行，最后导出png。

分类合并：

并不是把所有的图标都合并在一张图片里就是最好的，除了要控制图片大小之外还要注意以下方法。

- 按照图片排列方式，把排列方式一样的图片进行合并，便于样式控制。
- 按照模块或元件，把同属于一个模块或元件的图片进行合并，方便模块或元件的维护。

规范 / CSS规范 / 优化方案

- 按照图片大小，把大小一致或差不多的图片进行合并，可充分利用图片空间。
- 按照图片色彩，把色彩一致或差不多的图片进行合并，保证合并后图片的色彩不过于丰富，可防止色彩失真。
- 综合以上方法进行合并。

■ Hack的避免

- 当避免的代价较大时，可以使用Hack而不避免，比如你需要增加很多HTML或多写很多CSS时会得不偿失。
- 丰富的实战经验可以帮助你了解那些常见问题并用多种不同的思路来避免它，所以经验和思维方法在这里显得很重要。
- 根据你自己的能力来解决Hack的问题，我们不建议你用一个自己都没有把握的方法来避免Hack，因为也许你这个方法本身存在你没有发现的问题。

■ 便于阅读修改

如果你做到了“CSS规范”的所有要求，自然你也就写出了便于阅读和修改的漂亮的CSS。

当然，代码格式和命名规则是相对重要一些的。

■ 清晰的CSS模块

如果你做到了命名规则的要求，你的CSS模块也就清晰可见了。

用“注释”来说明每一个模块对于较大的CSS文件来说显得尤为重要。

■ 文件压缩

合理的书写CSS能很大程度上减少文件大小，完成后，在不损坏文件内容的情况下，想尽一切办法压缩你的CSS，你可以借助压缩工具把注释和多余的空格、换行去掉。

压缩工具详见：“HTML/CSS工具”部分。

■ 其他格式优化

优化方法请参见代码格式。

规范 / CSS规范 / 最佳实践

■ 最佳选择器写法（模块）

/* 这是某个模块 */

.m-nav{} /* 模块容器 */

.m-nav li,.m-nav a{} /*先共性 优化组合*/

.m-nav li{}/* 后个性 语义化标签选择器 */

.m-nav a{}/* 后个性中的共性 按结构顺序 */

.m-nav a.a1{}/* 后个性中的个性 */

.m-nav a.a2{}/* 后个性中的个性 */

.m-nav .crt a{}/* 交互状态变化 */

.m-nav .crt a.a1{}

.m-nav .crt a.a2{}

.m-nav .btn{}/* 典型后代选择器 */

.m-nav .btn-1{}/* 典型后代选择器扩展 */

.m-nav .btn-dis{}/* 典型后代选择器扩展 */

.m-nav .m-sch{}/*控制内部其他模块位置*/

.m-nav .u-sel{}/* 控制内部其他元件位置 */

.m-nav-1{}/* 模块扩展 */

.m-nav-1 li{}

.m-nav-big{}/* 模块扩展 */

■ 统一语义理解和命名

结构语义命名 (.g-、.u-)			内容语义命名 (.m-、.u-)			表现语义命名 (.f-、.s-)		
结构语义	命名	简写	内容语义	命名	简写	表现语义	命名	简写
文档	doc	doc	热点	hot	hot	字体大小	fontsize	fs
头部	head	hd	排行	top	top	字体颜色	fontcolor	fc
主体	body	bd	登录	login	log	字体粗细	fontweight	fw
尾部	foot	ft	标志	logo	logo	背景	background	bg
主栏	main	mn	广告	advertise	ad	背景颜色	backgroundcolor	bgc
主栏子容器	mainc	mnc	搜索	search	sch	背景图片	backgroundimage	bgi
侧栏	side	sd	幻灯	slide	sld	背景定位	backgroundposition	bgp
侧栏子容器	sidec	sdc	提示	tips	tips	边框颜色	bordercolor	bdc
导航	nav	nav	帮助	help	help	文本居中	textaligncenter	tac
子导航	subnav	snav	新闻	news	news	文本居右	textalignright	tar
面包屑	crumb	crm	下载	download	dld	文本居左	textalignleft	tal
切换标签	tab	tab	注册	regist	reg	垂直居中	verticalalignmiddle	vam
模块标题区	title	tt	状态	status	stts	浮动清除	clearboth	cb
模块内容区	content	ct	按钮	button	btn	向左浮动	floatleft	fl
列表	list	lst	投票	vote	vote	向右浮动	floatright	fr
定义列表	dlist	dlst	版权	copyright	cppt	内联块级	inlineblock	ib
表格	table	tb	下载	download	dld	溢出隐藏	overflowhidden	oh
表单	form	fm	错误	error	err	完全消失	displaynone	dn

规范 / CSS规范 / 典型错误

■ 不符合NEC规范的选择器用法

- `.class{}`

不要以一个没有类别的样式作为主选择器，这样的选择器只能作为后代选择器使用，比如`.m-xxx .class{}`。

- `.m-xxx div{}`

不要以没有语义的标签作为选择器，这会造成大面积污染，除非你可以断定现在或将来你的这个选择器不会污染其他同类。

- `.g-xxx .class{}`

不要在页面布局中使用后代选择器，因为这个后代选择器可能会污染里面的元素。

- `.g-xxx .m-yyy{}.g-xxx .u-yyy{}`

不要用布局去控制模块或元件，模块和元件应与布局分离独立。

- `.m-xxx .f-xxx{}.m-xxx .s-xxx{}`

不要通过模块或其他类来重定义或修改或添加已经定义好的功能类选择器和皮肤类选择器。

- `.m-xxx .class .class .class .class{}`

不要将选择器写的过于冗长，这会额外增加文件大小并且限制了太小范围的选择器，使树形结构过于严格应用范围过于局限，建议3-4个长度之内写完。

选择器并不需要完整反映结构嵌套顺序，相反，能简则简。

- `.m-xxx .m-yyy .zzz{}`

不要越级控制，如果`.zzz`是`.m-yyy`的后代选择器，那么不允许`.m-yyy`之外的选择器控制或修改`.zzz`。

此时可以使用`.m-yyy`的扩展来修改`.zzz`，比如`.m-yyy-1 .zzz{}`。

■ 扩展类使用错误

扩展类必须和其基类同时使用于同一个节点。

- 错误：`class="g-xxx g-yyy-1" class="m-xxx-1" class="u-xxx u-yyy-1" class="xxx-yyy"`。

- 正确：`class="g-xxx g-xxx-1" class="m-xxx m-xxx-1" class="u-yyy u-yyy-1" class="xxx xxx-yyy"`。

规范 / HTML规范 / 整体结构

■ html基础设施

- 文件应以“<!DOCTYPE>”首行顶格开始，推荐使用“<!DOCTYPE html>”。
- 必须申明文档的编码charset，且与文件本身编码保持一致，推荐使用UTF-8编码<meta charset="utf-8"/>。
- 根据页面内容和需求填写适当的keywords和description。
- 页面title是极为重要的不可缺少的一项。

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8"/>
5 <title>this is a html structure example</title>
6 <meta name="keywords" content="CSS HTML XHTML Standard Hack I
7 <meta name="description" content="this is a css and html lib
8 <link rel="icon" href="favicon.ico" type="image/x-icon"/>
9 <link rel="shortcut icon" href="favicon.ico" type="image/x-i
10 <link href="../assets/css/global.css" rel="stylesheet" type=
11 </head>
12 <body>
13
14 </body>
15 </html>
```

■ 结构顺序和视觉顺序基本保持一致

- 按照从上至下、从左到右的视觉顺序书写HTML结构。
- 有时候为了便于搜索引擎抓取，我们也会将重要内容在HTML结构顺序上提前。
- 用div代替table布局，可以使HTML更具灵活性，也方便利用CSS控制。
- table不建议用于布局，但表现具有明显表格形式的数据，table还是首选。

■ 结构、表现、行为三者分离，避免内联

- 使用link将css文件引入，并置于head中。
- 使用script将js文件引入，并置于body底部。

■ 保持良好的简洁的树形结构

- 每一个块级元素都另起一行，每一行都使用Tab缩进对齐（head和body的子元素不需要缩进）。删除冗余的行尾的空格。
- 使用4个空格代替1个Tab（大多数编辑器中可设置）。
- 对于内容较为简单的表格，建议将tr写成单行。

规范 / HTML规范 / 整体结构

- 你也可以在大的模块之间用空行隔开，使模块更清晰。

```
1 <body>
2 <!-- 侧栏内容区 -->
3 <div class="m-side">
4     <div class="side">
5         <div class="sidein">
6             <!-- 热门标签 -->
7             <div class="sideblk">
8                 <div class="m-hd3"><h3 class="tit">热门标签</h3>
9                 ...
10            </div>
11
12            <!-- 最热TOP5 -->
13            <div class="sideblk">
14                <div class="m-hd3"><h3 class="tit">最热TOP5</h3>
15                ...
16            </div>
17        </div>
18    </div>
```

■ 另外，请做到以下几点

- 结构上如果可以并列书写，就不要嵌套。

如果可以写成<div></div><div></div>那么就不要写成
<div><div></div></div>

- 如果结构已经可以满足视觉和语义的要求，那么就不要再有额外的冗余的结构。

比如<div><h2></h2></div>已经能满足要求，那么就不要再写成
<div><div><h2></h2></div></div>

- 一个标签上引用的className不要过多，越少越好。

比如不要出现这种情况：<div class="class1 class2 class3 class4"></div>

- 对于一个语义化的内部标签，应尽量避免使用className。

比如在这样一个列表中，li标签中的itm应去除：<ul class="m-help"><li class="itm"><li class="itm">

规范 / HTML规范 / 代码格式

■ 说明文案的注释方法

采用类似标签闭合的写法，与HTML统一格式；注释文案两头空格，与CSS注释统一格式。

- 开始注释：<!-- 注释文案 -->（文案两头空格）。
- 结束注释：<!-- /注释文案 -->（文案前加"/"符号，类似标签的闭合）。
- 允许只有开始注释！

```
1 <!-- 头部 -->
2 <div class="g-hd">
3   <!-- LOGO -->
4   <h1 class="m-logo"><a href="#">L
5   <!-- /LOGO -->
6   <!-- 导航 -->
7   <ul class="m-nav">
8     <li><a href="#">NAV1</a></li>
9     <li><a href="#">NAV2</a></li>
10   <!-- 更多导航项 -->
11 </ul>
12 <!-- /导航 -->
```

■ 代码本身的注释方法

单行代码的注释也保持同行，两端空格；多行代码的注释起始和结尾都另起一行并左缩进对齐。

```
1 <!-- <h1 class="m-logo"><a href="#">L
2 <!--
3 <ul class="m-nav">
4   <li><a href="#">NAV1</a></li>
5   <li><a href="#">NAV2</a></li>
6 </ul>
7 -->
```

■ HTML注释在IE6中的BUG

- 如果两个浮动元素之间存在注释，那么可能导致布局错位或文字的BUG。
- 所以，这种情况下，我们通常将注释去掉，或者索性采用模板语言（ftl、vm）的注释。

■ 严格的嵌套

- 尽可能以最严格的xhtml strict标准来嵌套，比如内联元素不能包含块级元素等等。
- 正确闭合标签且必须闭合。

■ 严格的属性

- 属性和值全部小写，每个属性都必须有一个值，每个值必须加双引号。
- 没有值的属性必须使用自己的名称做为值（checked、disabled、readonly、selected等等）。
- 可以省略style标签和script标签的type属性。

■ 常用的标签

见nec官网的常见标签表。

规范 / HTML规范 / 内容语义

■ 内容类型决定使用的语义标签

在网页中某种类型的内容必定需要某种特定的HTML标签来承载，也就是我们常常提到的根据你的内容语义化HTML结构。

■ 加强“资源型”内容的可访问性

在资源型的内容上加入描述文案，比如给img添加alt属性，在audio内加入文案等等。

■ 加强“不可见”内容的可访问性

背景图上的文字应该同时写在html中，并使用css使其不可见，有利于搜索引擎抓取你的内容，也可以在css失效的情况下看到内容。

■ 使用实体

以实体代替与HTML类似或相同的字符，避免浏览解析错误。

常用HTML字符实体：

字符	名称	实体名	实体数
"	双引号	"	"
&	&符	&	&

<	左尖括号(小于号)	<	<
>	右尖括号(大于号)	>	>
	空格	 	
	中文全角空格		　

常用特殊字符实体：

字符	名称	实体名	实体数
¥	元	¥	¥
	断竖线	¦	¦
©	版权	©	©
®	注册商标R	®	®
TM	商标TM	™	™
.	间隔符	·	·
«	左双尖括号	«	«
»	右双尖括号	»	»
°	度	°	°
×	乘	×	×
÷	除	÷	÷
‰	千分比	‰	‰

规范 / HTML规范 / 邮件内容

■ 邮件环境

邮件内容所在上下文或者说所在外部容器（以下简称环境）都是由邮箱服务商决定的，这就要求邮件内容需要在任何一种情况下都要正确显示。

这些环境可能是以下某几种情况：

- 可能是个iframe，你的内容是被放在body里面的；可能只是个div，你的内容就被放在这个div里面。
- 可能邮箱自身设置了些css，他可能对你产生未知的影响。
- 可能根本没有申明doctype，即使申明了，也不是你想要的doctype。

■ 避免嵌套在不正确的容器里

惑：因为容器可能是body或div，所以，我们邮件内容不应该是一个完整的html。

解：所以邮件内容应该是以div为根节点的html片段。

■ 避免css冲突或被覆盖

惑：因为环境中可能已经设置了css，比如一些reset、一些.class。

解：所以我们只能使用行内style来确保我们的效果，并且在内容根节点上设置基础style，并且尽量使用div、span等无语义标签。

```
1 <!-- 根节点 -->
2 <div style="width:600px;text-align:left;">
3   内容区域
4 </div>
5 <!-- 根节点-邮件内容居中 -->
6 <div style="text-align:center;">
7   <div style="width:600px;margin:0;">
8     内容区域
9   </div>
10 </div>
11 <!-- 如果使用语义化标签，那么需要多写一些 -->
12 <h2 style="width:100px;height:100px;">
```

■ 避免盒模型错误

惑：因为doctype的不确定性，我们在写style的时候，应该考虑无论doctype是什么情况，都

可以正常显示，doctype影响最大的就是盒模型的解析。

解：所以我们要将盒模型拆分开来写，比如我们将原本要定义在某个div上的height和padding分别写到这个div和他的父元素或子元素上。

```
1 <div style="height:100px;padding:20px;">
2 <!-- 上面的写法应该改成以下写法 -->
3 <div style="padding:20px 0;"><div style="height:100px;">
```

■ 其他注意事项

- 因为只能使用行内style，所以清除浮动需要使用额外标签。
- 避免使用绝对定位，可能会被过滤。

规范 / HTML规范 / 邮件内容

- 避免使用js, 可能会被过滤。
- 避免使用table布局, 不易于修改维护。
- 背景图片或内容图片上的文字信息, 必须在代码中可见。
- 如果没有特殊要求, 所有a链接都要从新窗口打开, 即target="_blank", 且a标签内容不能为空。
- 所有链接必须设置使用颜色、是否下划线, 即style="text-decoration:;color:;"。
- 重点检查ie!!!

■ 发现的问题及解决方案

问题: 部分智能手机的邮件客户端可能会有只显示部分的bug (宽度被截)。

解决: 在外面套一个同宽的table即可。

```
1 <table width="600" cellpadding="0" cellspacing="0" border="0"><tr><td>
2     <div style="width:600px;text-align:left;color:#000;font:normal 12px/15px simsun;ba
3         .....
4     </div>
5 </td></tr></table>
```

```
1 <div style="width:600px;text-align:le
2     <div style="height:268px;backgrou
3         <div style="height:228px;">
4             <div style="padding:21px
5                 <a href="http://yuedu
6                     text-indent:-2000px;
7             </div>
8             <h2 style="margin:0;paddi
9                 iPad不等于愤怒的小鸟! 不等于
10            </div>
11            <div style="padding:0 0 0 35p
12                target="_blank" style="color
```

框架

你是否为新建一个空空的html或者css文件而烦恼?来这里,选择一份合适的基础文件,然后在她的基础上开始你的代码!

- **css reset**

重置样式, 适用于需要兼容所有浏览器的网站, 基于html4&css2

- **css function**

功能样式, 适用于需要兼容所有浏览器的网站, 基于html4&css2

- **css animation**

常见动画效果的集合, 主要用于效果演示和参考, 基于css3

- **html template**

包含完整头部信息的html模板, 基于html4

- **html template2**

包含完整头部信息的html模板, 基于html5



框架 / CSS reset

■ 概述:

针对html标签, 统一将他们的默认样式全部清除, 并为你的网站配置合适的默认样式。

使用方法

通过右键另存为下载 **reset.css**(2012-09-28) 文件, 直接使用
或将代码复制到你 的css文件顶部。

框架代码 :

```
1  /* reset */
2  body,div,dl,dt,dd,ul,ol,li,h1,h2,h3,h4,h5,h6,pre,code,form,fieldset,legend,input,button,textarea,p,blockquote,th,td{margin:0}
3  table{border-collapse:collapse;border-spacing:0;}
4  caption,th{text-align:left;}
5  fieldset,img,a img,iframe,html,body{border:0;}
6  li{list-style:none;}
7  address,caption,cite,code,dfn,em,strong,th,var{font-style:normal;font-weight:normal;}
8  h1,h2,h3,h4,h5,h6,em,strong,b{font-size:100%;font-weight:bold;}
9  abbr,acronym{border:0;font-variant:normal;}
10 sup{vertical-align:text-top;}
11 sub{vertical-align:text-bottom;}
12 del,ins{text-decoration:none;}
13 [hidefocus]{outline:0;}
14 textarea,input,button,select{font-size:inherit;line-height:inherit;color:inherit;font-family:inherit;outline:0;}
15 textarea{overflow:auto;resize:none;}
16 body{word-wrap:break-word;font:12px/normal arial,simsun;color:#333;background:#fff;}
```

框架 / CSS function

■ 概述:

抽离了部分常用的功能型样式,使用前请先阅读规范中相关条目。

使用方法

通过右键另存为下载 **function.css**(2012-09-28) 文件,直接使用或将代码复制到你 的css文件顶部。



框架代码 :

```
1  /* function */
2  .f-cb:after,.f-cbli li:after{display:block;clear:both;}
3  .f-cb,.f-cbli li{zoom:1;}
4  .f-ib{display:inline-block;*display:inline;*zoom:1;}
5  .f-dn{display:none;}
6  .f-db{display:block;}
7  .f-fl{float:left;}
8  .f-fr{float:right;}
9  .f-pr{position:relative;}
10 .f-prz{position:relative;zoom:1;}
11 .f-oh{overflow:hidden;}
12 .f-ff0{font-family:arial;}
13 .f-ff1{font-family:simsun,\5b8b\4f53;}
14 .f-fs1{font-size:12px;}
15 .f-fs2{font-size:14px;}
16 .f-fwn{font-weight:normal;}
17 .f-fwb{font-weight:bold;}
18 .f-tal{text-align:left;}
19 .f-tac{text-align:center;}
20 .f-tar{text-align:right;}
21 .f-taj{text-align:justify;text-justify:inter-ideogram;}
22 .f-vam,.f-vama *{vertical-align:middle;}
23 .f-wsn{word-wrap:normal;white-space:nowrap;}
```

框架 / CSS Animation

■ 概述:

常见动画效果的集合，主要用于效果演示和参考（[点击查看代码库演示](#)），也可以直接调用。

使用方法

通过右键另存为下载 [animation.css](#)(2012-09-28)，直接使用或将代码复制到你的css文件中。

框架代码：

```
1  /* animation */
2  /* 淡入 */
3  .a-fadein{-webkit-animation-name:fadein;-moz-animation-name:fadein;-ms-animation-name:fadein;animation-name:fadein;}
4  /* define */
5  /* 淡入 */
6  @-webkit-keyframes fadein{
7      0%{opacity:0;}
8      100%{opacity:1;}
9  }
10 @-moz-keyframes fadein{
11     0%{opacity:0;}
12     100%{opacity:1;}
13 }
14 @-ms-keyframes fadein{
15     0%{opacity:0;}
16     100%{opacity:1;}
```


框架 / HTML Template

■ 概述:

包含了doctype及完整head信息的html基础模板,适用于您新建网页时使用,基于html4。

使用方法

通过右键另存为下载 **template.html**(2012-09-28) 文件, 直接使用或截取部分。



框架代码 :

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="utf-8"/>
5  <title>页面标题</title>
6  <meta name="keywords" content=""/>
7  <meta name="description" content=""/>
8  <meta http-equiv="x-ua-compatible" content="ie=edge"/>
9  <link type="text/css" rel="stylesheet" href="css/style.css"/>
10 <link href="favicon.ico" rel="icon" type="image/x-icon"/>
11 </head>
12 <body>
13 <div class="g-doc">
14     <!-- 头部 -->
15     <div class="g-hd">
16
17     </div>
18     <!-- /头部 -->
19     <!-- 主体 -->
20     <div class="g-bd">
21         <!-- 主栏 -->
22         <div class="g-mn">
23
```

框架 / HTML Template2

■ 概述:

包含了doctype及完整head信息的html基础模板,适用于您新建网页时使用,基于html5。

使用方法

通过右键另存为下载 [template2.html](#)(2012-09-28) 文件,直接使用或截取部分。

框架代码 :

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="utf-8"/>
5  <title>页面标题</title>
6  <meta name="keywords" content=""/>
7  <meta name="description" content=""/>
8  <link type="text/css" rel="stylesheet" href="css/style.css"/>
9  <link href="favicon.ico" rel="icon" type="image/x-icon"/>
10 </head>
11 <body>
12 <div class="g-doc">
13     <!-- 头部 -->
14     <header class="g-hd">
15
16     </header>
```

代码库

这里有各种常见布局、模块、技巧等代码,在这里找到她,或给你启发,或她就是你的

- 布局

提供各种常见的页面布局方法,比如定宽布局、自适应布局等

- 模块

提供各种常见的网页内容模块,比如导航、菜单、幻灯、图文列表等

- 元件

提供各种常见的网页内容元件,比如按钮、标题、输入框等

- BUG

展示了浏览器的BUG及其解决方案,比如IE双倍边距、3像素问题等

- 技巧

提供各种较难或特殊设计的解决技巧,比如垂直居中、自适应等

- 动画

使用CSS3实现的各种常见动画效果,比如淡入淡出、弹出等

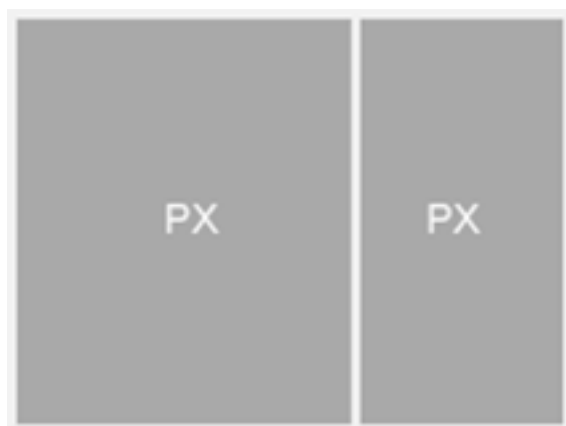


代码库

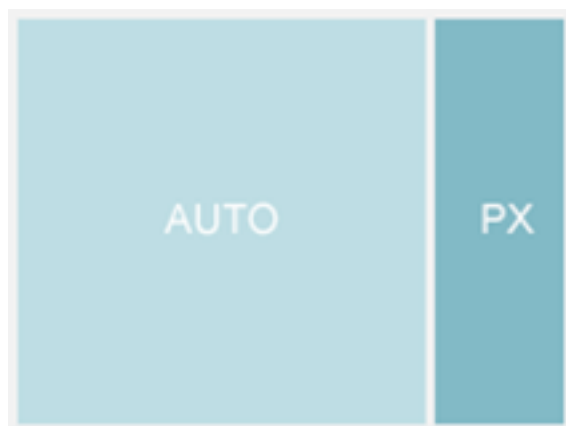
■ 布局

提供各种常见的页面布局方法,比如定宽布局、自适应布局等

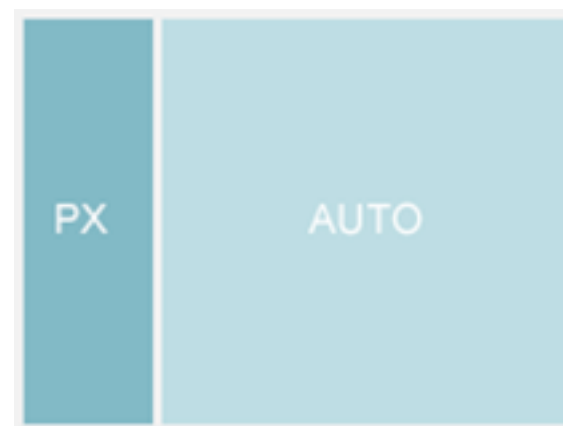
 查看更多



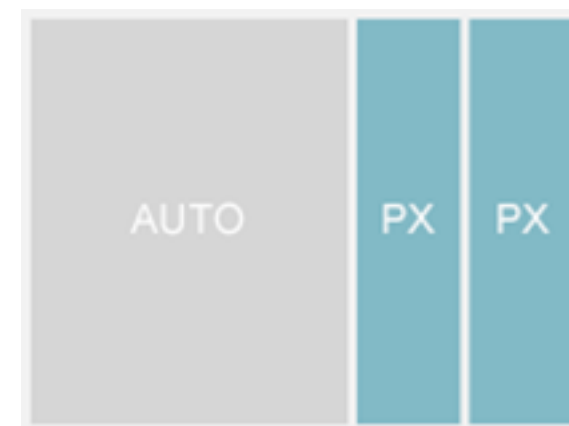
两列定宽布局



两列左侧自适应布局



两列右侧自适应布局



三列左侧自适应布局



三列右侧自适应布局

代码库

■ 模块

提供各种常见的网页内容模块,比如导航、菜单、幻灯、图文列表等

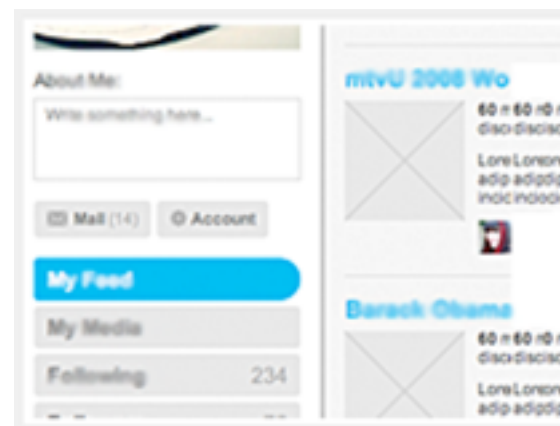
 查看更多



文本型圆角背景导航



不定宽简单边框型tab



左图右文图文列表



简易数据表格



简易图片幻灯片



简易翻页器



简易弹出浮层



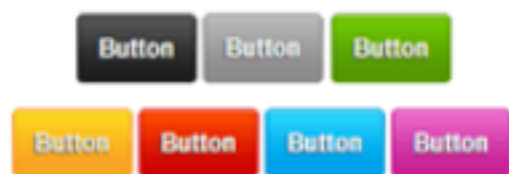
上图下文图文列表

代码库

■ 元件

提供各种常见的网页内容元件,比如按钮、标题、输入框等

 查看更多



定宽定高的自定义按钮



简易文本标题栏



简易单行文本输入框

代码库

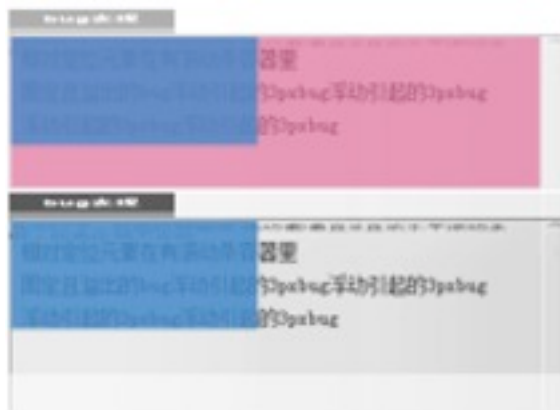
■ BUG

展示了浏览器的BUG及其解决方案,比如IE双倍边距、3像素问题等

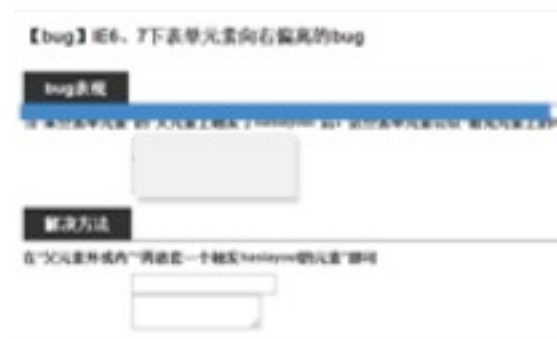
 查看更多



IE6中浮动块级元素的margin双倍



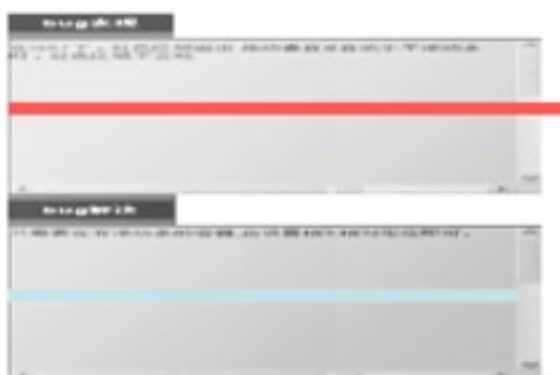
IE6中浮动引起的3px的bug



IE6中下表单元素向右偏离的bug



IE6中绝对定位位置错乱



IE6/7中相对定位的元素在有滚动条的容器里会固定不动且溢出



IE6中绝对定位在奇数宽高容器中的right和bottom的问题



IE6中绝对定位元素在浮动元素前后消失的问题



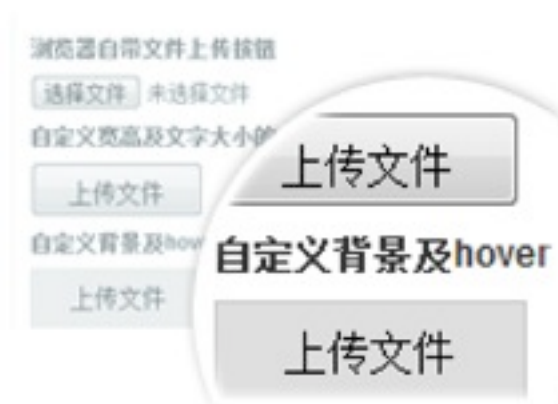
IE6/7中图片链接点击无效的bug

代码库

■ 技巧

提供各种较难或特殊设计的解决技巧,比如垂直居中、自适应等

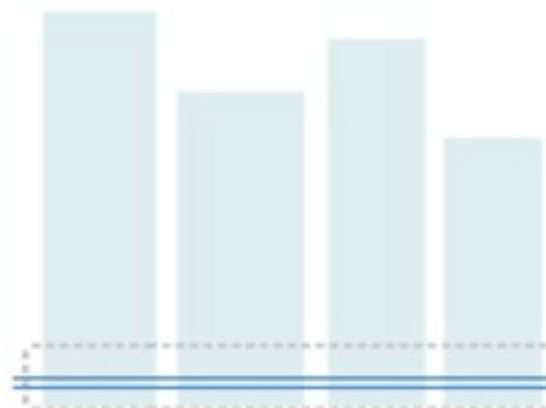
查看更多



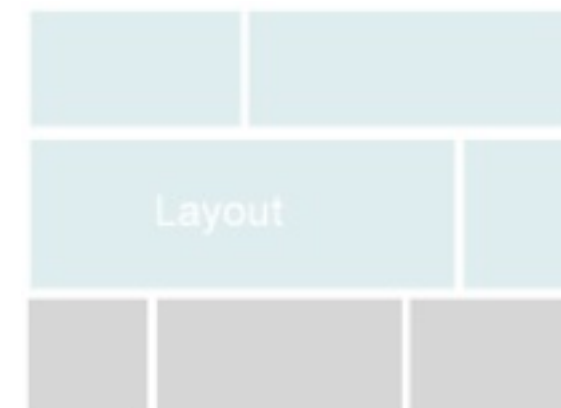
定制文件上传按钮



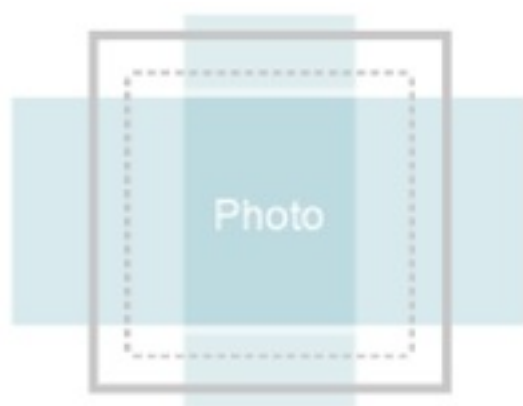
四周不留白图文列表



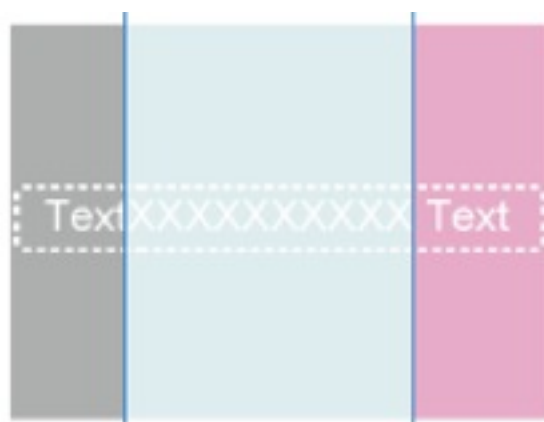
未知每项宽高的水平列表底部对齐



常见两栏三栏自适应布局



实现不同尺寸的图片水平垂直居中, 且溢出隐藏



实现单行前文本宽度自适应和后文本跟随



radio和checkbox与文本对齐

代码库

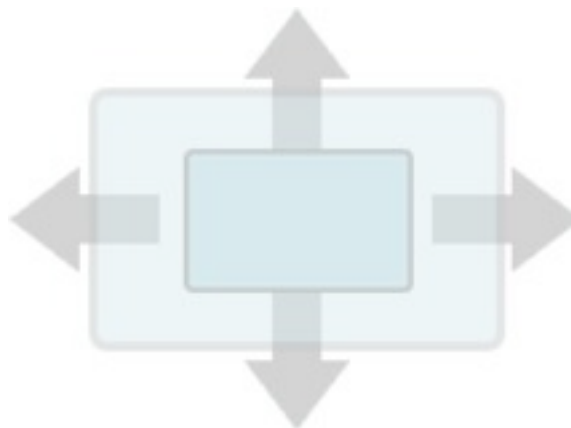
■ 动画

使用CSS3实现的各种常见动画效果,比如淡入淡出、弹出等

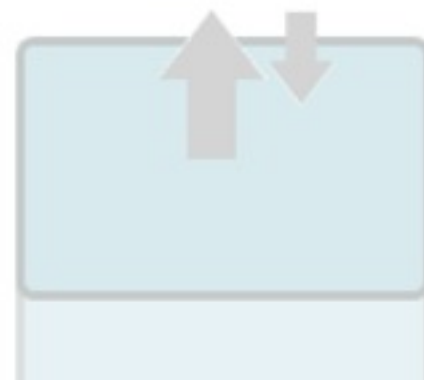
 查看更多



从下淡入效果



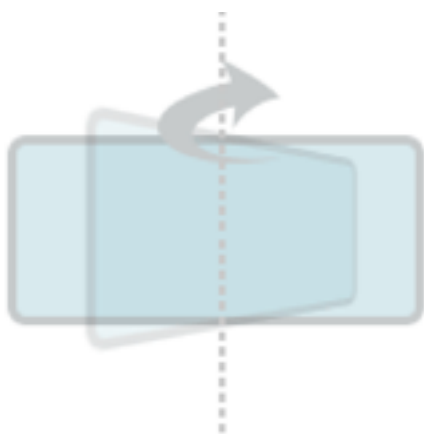
弹出效果



从下弹出效果



转入效果



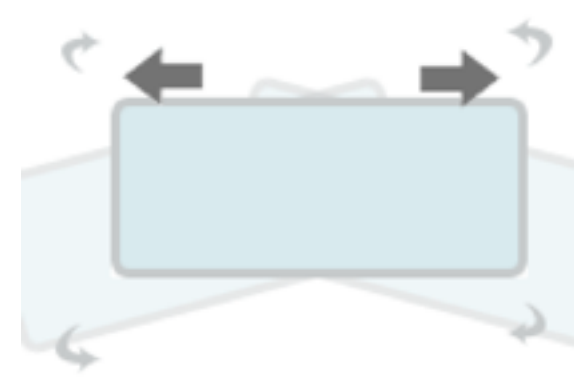
沿纵轴翻入效果



震颤效果



摇摆效果



摇晃效果

插件

你是否为新建一个空空的html文件或css文件而烦恼?来这里,选择一份合适的基础文件,然后在她的基础上开始你的代码!

- Eclipse插件
- Vim插件



插件 / Eclipse插件 / 安装

■ 介绍

xml文件，通过软件自带的模板功能导入，自动代码提示。

■ 下载地址

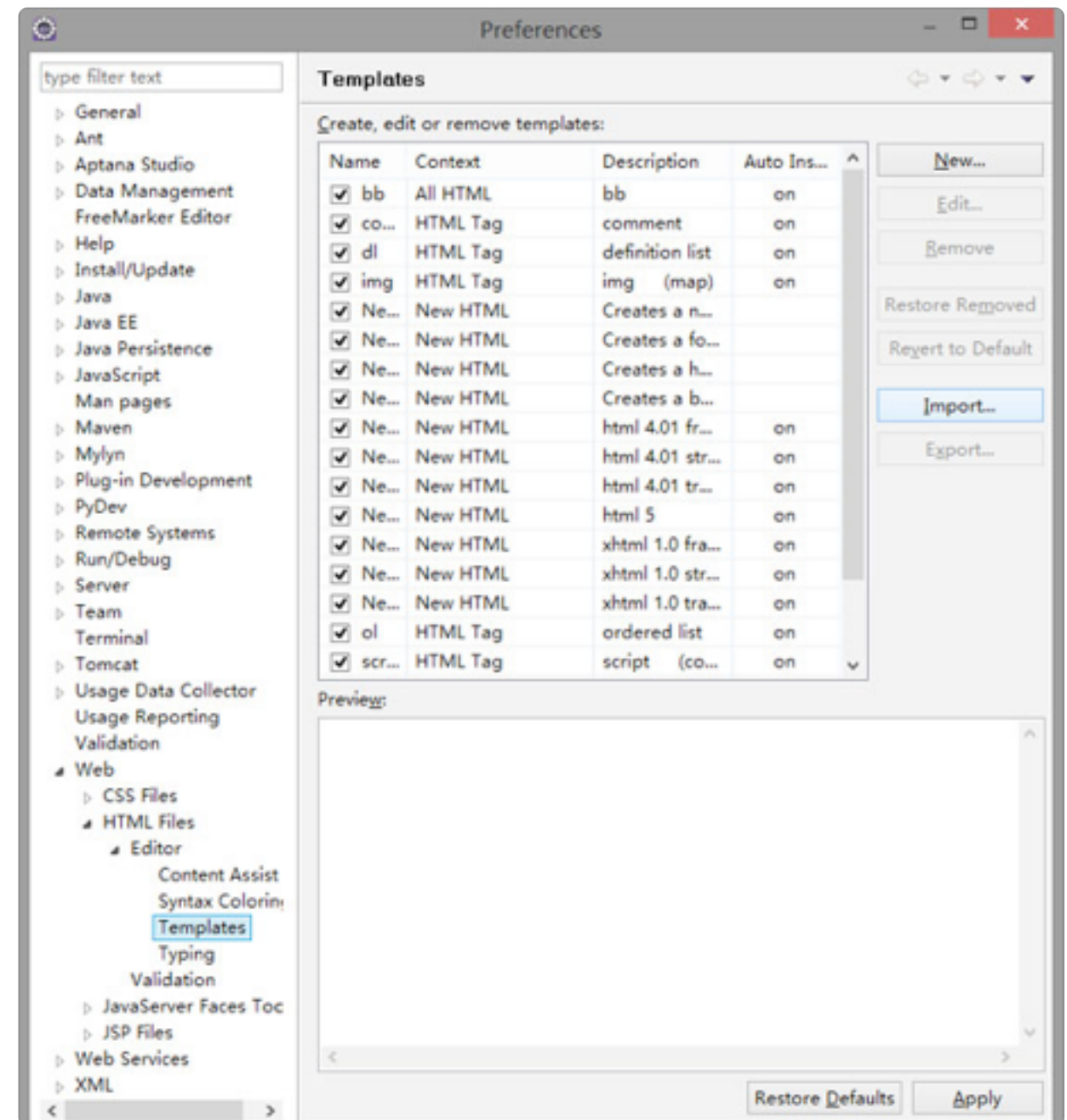
- [nec-plugin-eclipse-html.xml](#)(12-09-28)
- [nec-plugin-eclipse-css.xml](#)(12-09-28)

■ 安装步骤

1. 在菜单栏中打开 Window >> Preferences, 在左侧树形列表中, 选择 Web >> HTML Files (或CSS Files) >> Editor >> Templates。如右图。
2. 在右侧窗口中点击“import”按钮, 然后选择之前下载好的XML (html或css) 文件导入。

需要注意的问题

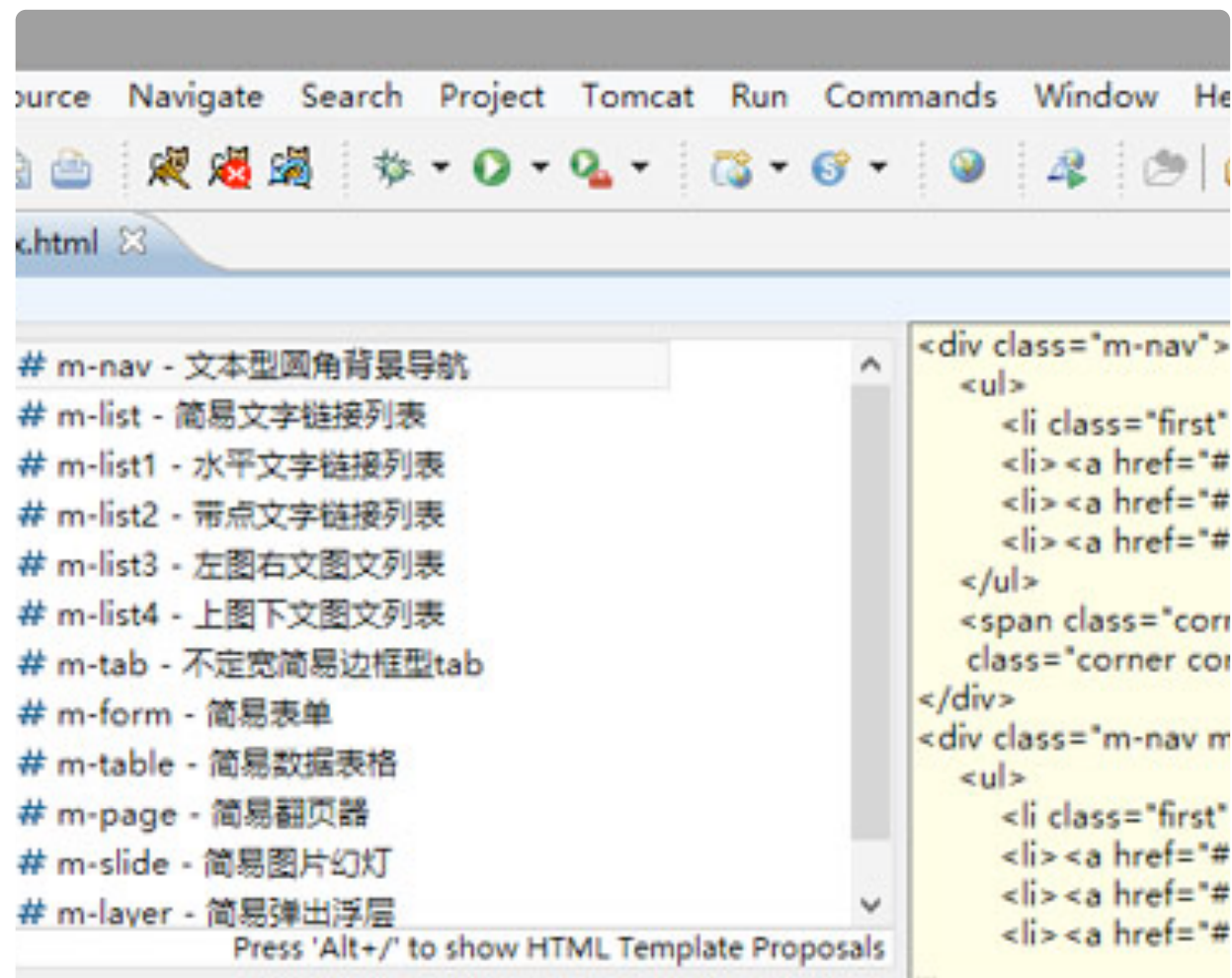
如果你的 Eclipse 的 Window >> Preferences 中没有 Web 项, 则需要安装 WTP (Web Tools Platform) 插件, 或者安装其他有web项的Eclipse版本。



插件 / Eclipse插件 / 使用

■ 代码自动插入

在编辑器（HTML Editor或CSS Editor）中，输入“m”（或g、u等）后按“alt+ /”（快捷键可以自定义，默认“alt+ /”）出现自动提示，即可选择您需要的代码，或者继续输入后面的字符“-xxx”，编辑器将自动匹配过滤及选中。



需注意的问题

编辑器自身问题：在CSS Editor中，可能无法自动过滤和选中，需手动选择。“-”可能造成在插入的片段前有已输入字符冗余，手动删除即可。

插件 / Vim插件 / 安装

■ 介绍

snippets文件，通过结合其他插件实现，自动代码提示。

■ 下载地址

- [nec-plugin-vim-html.snippets](#)(12-09-28)
- [nec-plugin-vim-css.snippets](#)(12-09-28)

■ 安装第三方插件：

1. snipMate:使 Vim 具有 TextMate 风格的 snippets 功能。
2. vim-addon-mw-utils: snipMate 依赖组件。
3. tlib_vim: snipMate 依赖组件。
4. neocomplcache : Vim 的代码自动提示插件（如无需此功能，可不安装本插件和下一个插件）。

5. neocomplcache-snippets-complete: neocomplcache 对 snipMate 的功能插件。

提示：

下载上述插件并解压，将各插件子目录下的文件放置到对应的 Vim 运行时目录（WIN：.../vim/vimfiles/）（Linux： ~/.vim/）的子目录中。

■ 设置自动提示

执行 |:NeoComplCacheEnable| 命令或者在你的 vimrc 文件中加入 let g:neocomplcache_enable_at_startup = 1。

■ 安装NEC插件

将已下载的snippets文件复制到 .../vim/vimfiles/snippets/ 下，重命名为 html.snippets（或css.snippets），若目录中已有同名文件，请用编辑器将插件内容复制添加到文件末尾。

■ 完成安装

重启Vim（每次插件更新后都需要重启Vim）。

插件 / Vim插件 / 使用

■ 设置文件类型

:setf html (或:setf css)。

[No Name] - GVIM

File Edit Tools Syntax Buffers Window Help

VIM - Vi IMproved

version 7.3.4

by Bram Moolenaar

Vim is open source and freely available

Become a registered User

type :help register<Enter>

type :q<Enter>

type :help<Enter> or <F1>

type :help version7<Enter>

:setf html

■ 代码自动插入

切换至 Vim 的插入 (insert) 模式, 输入“m-” (或g、u-等), 出现自动提示, 选择后“回车键”或点击完成关键字输入, 然后再按“Tab”键完成代码片段插入。

The screenshot shows a Vim editor window titled "[No Name] - GVIM". The interface includes a standard toolbar at the top. The main editing area contains a list of CSS classes and their corresponding Snippet code, displayed in two columns. The first column lists various CSS classes like "m-form", "m-ilst-lr", etc., followed by comments in Chinese. The second column shows the corresponding Snippet code starting with "[Snip]" and "<div class='...'>".

CSS Class / Comment	Snippet Code
m-form//简单表单	[Snip] <div class="m-f...
m-ilst-lr//左图右文图文列表	[Snip] <div class="m-il...
m-ilst-tb//上图下文图文列表	<Snip> <div class="m-il...
m-layer//简单弹出浮层	[Snip] <div class="m-la...
m-nav//文本型圆角背景导航	[Snip] <div class="m-na...
m-page//简单翻页器	[Snip] <div class="m-pa...
m-slide//简单图片幻灯	[Snip] <div class="m-sl...
m-tab//不定宽简单边框型tab	[Snip] <div class="m-ta...
m-table//简单数据表格	[Snip] <table class="m-
m-tlist-x//水平文字链接列表	[Snip] <div class="m-tl...
m-tlist-y//简易文字链接列表	[Snip] <ul class="m-tli...
m-tlist//带点文字链接列表	[Snip] <div class="m-tl...

At the bottom of the screen, there is a status bar that reads: "-- User defined completion (^U^N^P) Back at original".

需注意的问题

Vim中文乱码的问题：直接安装的Vim编辑器可能会出现中文乱码的情况，可以通过修改安装目录下的“_vimrc”文件进行解决，具体修改方法请使用搜索引擎查找。



本版本只含部分NEC内容，了解更多内容请参见

网址：<http://nec.netease.com>

联系我们

邮箱：nehzwd@163.com

地址：浙江省杭州市滨江区网商路599号网易大厦



nec
nec.netease.com

