**1 前言**

Stylus 是一个CSS的预处理框架，2010年产生，来自Node.js社区，主要用来给Node项目进行CSS预处理支持，所以 Stylus 是一种新型语言，可以创建健壮的、动态的、富有表现力的CSS。比较年轻，其本质上做的事情与 sass/less等类似，应该是有很多借鉴，所以近似脚本的方式去写CSS代码。 Stylus默认使用 .styl 的作为文件扩展名，支持多样性的CSS语法。 Stylus功能上更为强壮，和js联系更加紧密。

**2 代码风格**

**2.1 文件**

**[建议] CSS 文件使用无 BOM 的 UTF-8 编码。**

解释：

UTF-8 编码具有更广泛的适应性。BOM 在使用程序或工具处理文件时可能造成不必要的干扰。

### **2.2 缩进**

**[强制] 使用 4 个空格做为一个缩进层级，不允许使用 2 个空格 或 tab 字符。**

示例：

.code

    background-image url(/assets/images/home/code.png)

    background-repeat no-repeat

    height 34px

    width 100px

### **2.3 空格**

**[强制] 属性名 与之后的 : 之间不允许包含空格， : 与 属性值 之间必须包含空格。**

示例：

margin 0

#### **[强制] 列表型属性值 书写在单行时，, 后必须跟一个空格。**

示例：

font-family Arial, sans-serif

**2.4 选择器**

#### **[强制] 当一个 rule 包含多个 selector 时，每个选择器声明必须独占一行。**

示例：

/\* good \*/  
.post,

.page,

.comment

    line-height 1.5

/\* bad \*/

.post, .page, .comment

    line-height 1.5

**[强制] 避免>、+、~ 选择器的使用，而使用stylus代替，最多不可超过5层。**

示例：

.main

    background #fff

    .nav

        margin-bottom 10px

        .mask

            float right

**2.5 样式书写规范格式**

**[强制] 每个样式对应的格式书写应省略 “:” 和 “;” ,使代码尽可能简洁。**

示例：

/\* good \*/

.comment

    line-height 1.5

/\* bad \*/

.comment

    line-height: 1.5;

### **2.6 属性**

**[强制] 属性定义必须另起一行。**

示例：

.selector

    margin 0

    padding 0

## **3 通用**

**3.1 选择器**

#### **[强制] 如无必要，不得为 id、class 选择器添加类型选择器进行限定。**

解释：

在性能和维护性上，都有一定的影响。

示例：

/\* good \*/

#error,

.danger-message

    font-color #c00

/\* bad \*/

dialog#error,

p.danger-message

    font-color #c00

### **3.2 属性缩写**

#### **[建议] 在可以使用缩写的情况下，尽量使用属性缩写。**

示例：

/\* good \*/

.post

    font 12px/1.5 arial, sans-serif

/\* bad \*/

.post

    font-family arial, sans-serif

    font-size 12px

    line-height 1.5

**[建议] 使用 border / margin / padding 等缩写时，应注意隐含值对实际数值的影响，确实需要设置多个方向的值时才使用缩写。**

解释：

border / margin / padding 等缩写会同时设置多个属性的值，容易覆盖不需要覆盖的设定。如某些方向需要继承其他声明的值，则应该分开设置。

示例：

/\* good \*/

.page

    margin-right auto

    margin-left auto

.featured

    border-color #69c

/\* bad \*/

.page

    margin 5px auto /\* introducing redundancy \*/

.featured

    border 1px solid #69c /\* introducing redundancy \*/

### **3.3 属性书写顺序**

#### **[建议] 同一 rule set 下的属性在书写时，应按功能进行分组，并以 **Formatting Model（布局方式、位置） > Box Model（尺寸） > Typographic（文本相关） > Visual（视觉效果）** 的顺序书写，以提高代码的可读性。**

解释：

* Formatting Model 相关属性包括：position / top / right / bottom / left / float / display / overflow 等
* Box Model 相关属性包括：border / margin / padding / width / height 等
* Typographic 相关属性包括：font / line-height / text-align / word-wrap 等
* Visual 相关属性包括：background / color / transition / list-style 等

另外，如果包含 content 属性，应放在最前面。

示例：

.sidebar

    /\* formatting model: positioning schemes / offsets / z-indexes / display / ...  \*/

    position absolute

    top 50px

    left 0

    overflow-x hidden

    /\* box model: sizes / margins / paddings / borders / ...  \*/

    width 200px

    padding 5px

    border 1px solid #ddd

    /\* typographic: font / aligns / text styles / ... \*/

    font-size 14px

    line-height 20px

    /\* visual: colors / shadows / gradients / ... \*/

    background #f5f5f5

    color #333

    transition color 1s

### **3.4 清除浮动**

#### **[建议] 当元素需要撑起高度以包含内部的浮动元素时，通过对伪类设置 clear 或触发 BFC 的方式进行 clearfix。尽量不使用增加空标签的方式。**

解释：

触发 BFC 的方式很多，常见的有：

* float 非 none
* position 非 static
* overflow 非 visible

如希望使用更小副作用的清除浮动方法，参见 [A new micro clearfix hack](http://nicolasgallagher.com/micro-clearfix-hack/) 一文。

另需注意，对已经触发 BFC 的元素不需要再进行 clearfix。

### **3.5 !important**

#### **[建议] 尽量不使用 !important 声明。**

#### **[建议] 当需要强制指定样式且不允许任何场景覆盖时，通过标签内联和 !important 定义样式。**

解释：

必须注意的是，仅在设计上 确实不允许任何其它场景覆盖样式 时，才使用内联的 !important 样式。通常在第三方环境的应用中使用这种方案。下面的 z-index 章节是其中一个特殊场景的典型样例。

### **3.6 z-index**

#### **[建议] 将 z-index 进行分层，对文档流外绝对定位元素的视觉层级关系进行管理。**

解释：

同层的多个元素，如多个由用户输入触发的 Dialog，在该层级内使用相同的 z-index 或递增 z-index。

建议每层包含100个 z-index 来容纳足够的元素，如果每层元素较多，可以调整这个数值。

#### **[建议] 在可控环境下，期望显示在最上层的元素，z-index 指定为 999999。**

解释：

可控环境分成两种，一种是自身产品线环境；还有一种是可能会被其他产品线引用，但是不会被外部第三方的产品引用。

不建议取值为 2147483647。以便于自身产品线被其他产品线引用时，当遇到层级覆盖冲突的情况，留出向上调整的空间。

**[建议] 在第三方环境下，期望显示在最上层的元素，通过标签内联和 !important，将 z-index指定为 2147483647。**

解释：

第三方环境对于开发者来说完全不可控。在第三方环境下的元素，为了保证元素不被其页面其他样式定义覆盖，需要采用此做法。

**4 值与单位**

### **4.1 文本**

#### **[强制] 文本内容必须用双引号包围。**

解释：

文本类型的内容可能在选择器、属性值等内容中。

示例：

/\* good \*/

html:before

    font-family "Microsoft YaHei", sans-serif

    content "“"

/\* bad \*/

html:before

    font-family 'Microsoft YaHei', sans-serif

    content '“'

### **4.2 数值**

#### **[强制] 当数值为 0 - 1 之间的小数时，省略整数部分的 0。**

示例：

/\* good \*/

panel

    opacity .8

/\* bad \*/

panel

    opacity 0.8

### **4.3 url()**

#### **[强制] url() 函数中的路径不加引号。**

示例：

body

    background url(bg.png)

#### **[建议] url() 函数中的绝对路径可省去协议名。**

示例：

body

    background url(//baidu.com/img/bg.png) no-repeat 0 0

### **4.4 长度**

#### **[强制] 长度为 0 时须省略单位。 (也只有长度单位可省)**

示例：

/\* good \*/

body

    padding 0 5px

/\* bad \*/

body

    padding 0px 5px

### **4.5 颜色**

#### **[强制] RGB颜色值必须使用十六进制记号形式 #rrggbb。不允许使用 rgb()。**

解释：

带有alpha的颜色信息可以使用 rgba()。使用 rgba() 时每个逗号后必须保留一个空格。

示例：

/\* good \*/

.success

    box-shadow 0 0 2px rgba(0, 128, 0, .3)

    border-color #008000

/\* bad \*/

.success

    box-shadow 0 0 2px rgba(0,128,0,.3)

    border-color rgb(0, 128, 0)

#### **[强制] 颜色值可以缩写时，必须使用缩写形式。**

示例：

/\* good \*/

.success

    background-color #aca

/\* bad \*/

.success

    background-color #aaccaa

#### **[强制] 颜色值不允许使用命名色值。**

示例：

/\* good \*/

.success

    color #90ee90

/\* bad \*/

.success

    color lightgreen

## **7 响应式**

#### **[强制] Media Query 不得单独编排，必须与相关的规则一起定义。**

示例：

/\* Good \*/

/\* smart styles \*/

@media $smart

    /\* smart styles \*/

/\* tablet styles \*/

@media $tablet

    /\* tablet styles \*/

/\* desktop styles \*/

@media $desktop

    /\* desktop styles \*/

/\* Bad \*/

@media screen and (max-width: 480px)

#### **[强制] Media Query 如果有多个逗号分隔的条件时，应将每个条件放在单独一行中。**

示例：

@media

$smart,

$desktop

      /\* stylus code \*/

**[建议] 尽可能给出在高分辨率设备 (Retina) 下效果更佳的样式。**

**8 base样式**

**[强制] 基础样式编排尽量按bootstrap那一套规则编写。而元素在引入bootastp也得到了样式的默认设置**

<http://www.bootcss.com/>

**9 常用抽象类名与标签整理**

.cursor {cursor: pointer;}

.unselect{user-select: none;}

.w100{width: 100%;}

.h100{height: 100%;}

.displayIB{display: inline-block;}

.clearfix{clear: both;}

.pull-left{float: left;}

.pull-right{float: right;}

.pr{position: relative;}

.pb{position: absolute;}

.font-defualt{“Helvetic we’ve”, helvetica, Aria;}

.table-border-0{

                          border-top-width: 0;

                          border-right-width: 0;

                          border-bottom-width:0;

                          border-left-width:0;

                         }

.over-hidden{white-space:text-overflow;}

.text-dn{text-decoration: none;}

bootstrap:

<mark></mark>

<del></del>

<ins></ins>

<em></em>

text-left、text-right、text-center、text lowercase、text-uppercase、text-capitalize

<abbr></abbr>／.initialllism

<blockquote></blockquote>

<blockquete-reverse></blockquete-reverse>

list-inline

**10 颜色**

**[强制] 使用颜色统一配置，利用stylus特性定义统一变量名。**

示例：

／＊ 定义 ＊／

$redButton = #f45353

／＊ 使用 ＊／

button

     $redButton

颜色可以按照如下几种分类

|  |  |
| --- | --- |
| border | 主／副 |
| font | 主／副 |
| bg | 主／副 |
| input | default／focus |
| button | A／B.. |
| special | 突出／低调 |

**11 字体**

**11.1 字体族**

**[强制] 在index.styl引入定义变量指明字体库引用，从而切换字体风格。**

$icon-font-path = "../../bower\_components/bootstrap-stylus/fonts/"

**11.1 字体大小粗细**

**[强制] 使用同一配置，定义大小级别纬度覆盖率广泛的字体。**

示例：

／＊ 定义 ＊／

$font-large = 36px

／＊ 使用 ＊／

button

     $font-large

|  |  |
| --- | --- |
| large | 36px |
| nomal | 24px |
| small | 12px |
| bolder | 1000 |
| bold | 500 |
| light | 100 |

**12 图片**

**[强制] 使用同一文件包设定变量加包名下图片具体路径。**

示例：

／＊ 定义 ＊／

bg-image(src)

    background-image:url($imgRoot+src)

/\* good \*/

.box

    bg-image(bg.png)

/\* bad \*/

.box

    background url(assets/main/bg.png)

**13 状态**

**[强制] hover、active、link不同状态统一效果。**

|  |  |
| --- | --- |
| toggle-hover/toggle-active/toggle-link | opacity: num |
| scale-hover/... | transform:scale(num) |
| bg-hover/.. | background: url(string) |
| animate-hover/.. | animate: name |
| display-hover/.. | overflow:visible/auto |
| ... | ... |