xu

neusoft |

css样式书写规范

版本追踪

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 文档版本 | 日期 | 作者 | 操作 | 描述 |
| V0.5 | 2017-03-20 | 徐健 | 创建 | 创建文档并搭建基本框架。 |
| V1.0 | 2017-05-11 | 徐健 | 增加 | 添加预处理框架章节 |

## 编码设置

采用utf-8编码，在CSS文件头部使用：

*@charset "utf-8";*

注意，必须要定义在 CSS 文件所有字符的前面（包括编码注释），@charset 才会生效。

反例：

*/\* 字符编码 \*/*

*@charset "utf-8";*

*html,*

*body {*

*height: 100%;*

*}*

*@charset "utf-8";*

## 命名空间规范

这里采用宽松的命名规范，需要注意一些约定俗成的规则。不要使用下划线（\_）进行连接，这样可以和js的变量区分开；选择器的名称都为小写。除了以上两点之外还有如下强制规定：

* 布局：以 g 为命名空间，例如：.g-wrap 、.g-header、.g-content。
* 状态：以 s 为命名空间，表示动态的、具有交互性质的状态，例如：.s-current、s-selected。
* 工具：以 u 为命名空间，表示不耦合业务逻辑的、可复用的的工具，例如：u-clearfix、u-ellipsis。
* 组件：以 m 为命名空间，表示可复用、移植的组件模块，例如：m-slider、m-dropMenu。
* 钩子：以 j 为命名空间，表示特定给 JavaScript 调用的类名，例如：j-request、j-open。

## 选择器

选择器的命名要遵循尽可能短的原则，尽量简洁且具有语义化，不应该出现 g-abc 这种语义模糊的命名。

* 当一个规则包含多个选择器时，每个选择器独占一行。
* 、+、~、> 选择器的两边各保留一个空格。

示例：

*.g-header > .g-header-des,*

*.g-content ~ .g-footer {*

*}*

## 规则声明块

* 当规则声明块中有多个样式声明时，每条样式独占一行。
* 在规则声明块的左大括号 { 前加一个空格。
* 在样式属性的冒号 : 后面加上一个空格，前面不加空格。
* 在每条样式后面都以分号 ; 结尾。
* 规则声明块的右大括号 } 独占一行。
* 每个规则声明间用空行分隔。
* 所有最外层引号使用单引号 ' 。
* 当一个属性有多个属性值时，以逗号 , 分隔属性值，每个逗号后添加一个空格，当单个属性值过长时，每个属性值独占一行（页面初始化样式除外）。

示例：

*.g-footer,*

*.g-header {*

*position: relative;*

*}*

*.g-content {*

*background:*

*linear-gradient(135deg, deeppink 25%, transparent 25%) -50px 0,*

*linear-gradient(225deg, deeppink 25%, transparent 25%) -50px 0,*

*linear-gradient(315deg, deeppink 25%, transparent 25%),*

*linear-gradient(45deg, deeppink 25%, transparent 25%);*

*}*

*.g-content::before {*

*content: '';*

*}*

## 数值与单位

* 当属性值或颜色参数为 0 - 1 之间的数时，不要省略小数点前的 0 。

示例：

*color: rgba(255, 255, 255, 0.5)*

*~~color: rgba(255, 255, 255, .5);~~*

* 当长度值为 0 时省略单位。

示例：

*~~margin: 0px auto~~*

*margin: 0 auto*

* 十六进制的颜色属性值使用小写和尽量不要使用简写。

示例：

*color: #ffcc00*

*~~color: #fc0~~*

## 样式属性顺序

单个样式规则下的属性在书写时，应按功能进行分组，并以 Positioning Model > Box Model > Typographic > Visual 的顺序书写，提高代码的可读性。

* 如果包含 content 属性，应放在最前面（before/after）；
* Positioning Model 布局方式、位置，相关属性包括：position / top / right / bottom / left / z-index / display / float / ...
* Box Model 盒模型，相关属性包括：width / height / padding / margin / border / overflow / ...
* Typographic 文本排版，相关属性包括：font / line-height / text-align / word-wrap / ...
* Visual 视觉外观，相关属性包括：color / background / list-style / transform / animation / transition / ...

原则说明：

如果使用float/flex/grid布局的话必须要在第一行表示，如果采用默认布局则Position处在第一位，因为他可以使一个元素脱离正常文本流，并且覆盖盒模型相关的样式。盒模型紧跟其后，因为他决定了一个组件的大小和位置。其他属性只在组件内部起作用或者不会对前面两种情况的结果产生影响，所以他们排在后面。

## 代码注释

* 单行注释星号（\*）与文字之间要保留一个空格

示例：

*/\* AQI变色样式 \*/*

* 多行注释星号要一列对齐，星号与内容之间必须保留一个空格。

示例：

*/\*\**

*\* 这是一条多行注释*

*\*/*

* 规则声明块内注释使用‘’//‘’注释，后面加上一个空格，注释独立一行。

示例：

*.g-footer {*

*border: 0;*

*// ....*

*}*

* 文件顶部必须包含文件注释，用 @name 标识文件说明。星号要一列对齐，星号与内容之间必须保留一个空格，标识符冒号与内容之间必须保留一个空格。

示例：

*/\*\**

*\* @name: 领导驾驶舱页面*

*\* @description: 用于页面的全局样式…*

*\* @author: bruce（bruce870827@gmail.com）*

*\* author-name2(mail-name2@domain.com)*

*\* @update: 2015-04-29*

*\*/*

备注：当该业务项目主要由固定的一个或多个人负责时，需要添加@author标识，一方面是尊重劳动成果，另一方面方便在需要时快速定位责任人。

## 预处理框架使用

使用 SASS 、 LESS 等预处理器时，建议嵌套层级不超过 3 层。使用框架的宗旨是可以多人快速便捷的开发，而不是炫耀自己掌握多少复杂的写法。在这里以less为例，根据其语法，简要制定书写规范：

* 变量

*@border-color : #b5bcc7;*

*.mythemes tableBorder{*

*border : 1px solid @border-color;*

*}*

* 混入
* 套嵌

*// 定义一个样式选择器*

*.roundedCorners(@radius:5px) {*

*-moz-border-radius: @radius;*

*-webkit-border-radius: @radius;*

*border-radius: @radius;*

*}*

*// 在另外的样式选择器中使用*

*#header {*

*.roundedCorners;*

*}*

*#footer {*

*.roundedCorners(10px);*

*}*

*#home{*

*color : blue;*

*width : 600px;*

*height : 500px;*

*border:outset;*

*#top{*

*border:outset;*

*width : 90%;*

*}*

*#center{*

*border:outset;*

*height : 300px;*

*width : 90%;*

*#left{*

*border:outset;*

*float : left;*

*width : 40%;*

*}*

*#right{*

*border:outset;*

*float : left;*

*width : 40%;*

*}*

*}*

*}*

* 深度选择器

备注：

*<style rel="stylesheet/scss" lang="scss" scoped>*

*.panThumb{*

*z-index: 100;*

*height: 70px!important;*

*width: 70px!important;*

*position: absolute!important;*

*top: -45px;*

*left: 0px;*

*border: 5px solid #ffffff;*

*background-color: #fff;*

*margin: auto;*

*box-shadow: none!important;*

*// 深度作用选择器*

*/deep/ .pan-info {*

*box-shadow: none!important;*

*}*

*}*

*</style>*

在less中我们最常用的基本上就是套嵌、混入和变量三个特性，如有业务需要必须使用其他特性，比如运算等特性一定要有完整的注释

## 其它

* 合理使用引号（“）

**Font-family内使用引号**，当字体名字中间有空格，中文名字体及 Unicode 字符编码表示的中文字体，为了保证兼容性，都建议在字体两端添加单引号或者双引号。

示例：

*body {*

*font-family: 'Microsoft YaHei', '黑体-简', '\5b8b\4f53';*

*}*

**Background-image的url地址要使用引号，**如果路径里面有空格，旧版 IE 是无法识别的，会导致路径失效，建议不管是否存在空格，都添加上单引号或者双引号。

示例：

*div {*

*background-image: url('...');*

*}*

* 避免使用！Important

除去某些极特殊的情况，尽量不要不要使用 !important。

进行多人协作的项目时候，当发现自己写的新的样式无法覆盖一个旧的样式，只有加上 !important 才能生效时，是因为你新定义的选择器的优先级不够旧样式选择器的优先级高。所以，一定要先优化自身新样式的选择器再做决定。

当任务紧急时，使用！important临时做调整后，一定要在上方添加“todo”注释，以便以后有时间进行优化

* 如果必须使用sass框架，用scss代替。

Scss是sass3中引入的新语法特性，由于sass在引用过程中的依赖较多、版本要求严格且不支持nodejs7+的版本，所以在用自动化构建工具打包时候可能会出现（win系统一定会出现）各种各样的问题，而scss作为sass的升级版本很好的解决了兼容的问题（尤其是在vue+webpack的项目中）。

* 在vue项目中，不允许在模板中插入样式片段。