**前端代码规范-CSS**

## CSS文件命名

多个单词组成时，采用中划线连接方式，比如说: my-css.css

## 语法

1. 使用四个空格的 soft tabs — 这是保证代码在各种环境下显示一致的唯一方式。
2. 使用组合选择器时，保持每个独立的选择器占用一行。
3. 为了代码的易读性，在每个声明的左括号前增加一个空格。
4. 声明块的右括号应该另起一行。
5. 每条声明 : 后应该插入一个空格。
6. 每条声明应该只占用一行来保证错误报告更加准确。
7. 所有声明应该以分号结尾。虽然最后一条声明后的分号是可选的，但是如果没有他，你的代码会更容易出错。
8. 逗号分隔的取值，都应该在逗号之后增加一个空格。比如说box-shadow
9. 不要在颜色值 rgb() rgba() hsl() hsla()和 rect()中增加空格，并且不要带有取值前面不必要的 0 (比如，使用 .5 替代 0.5)。
10. 所有的十六进制值都应该使用小写字母，例如 #fff。因为小写字母有更多样的外形，在浏览文档时，他们能够更轻松的被区分开来。
11. 尽可能使用短的十六进制数值，例如使用 #fff 替代#ffffff。
12. 为选择器中的属性取值添加引号，例如input[type="text"]。 他们只在某些情况下可有可无，所以都使用引号可以增加一致性。
13. 不要为 0 指明单位，比如使用 margin: 0; 而不是margin: 0px;。

## 声明顺序

相关的属性声明应该以下面的顺序分组处理：

Positioning 定位

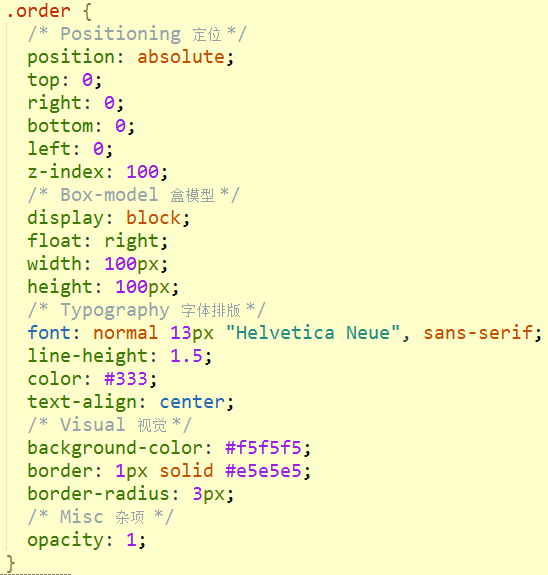
Box model 盒模型

Typographic 排版

Visual 外观

Positioning 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。

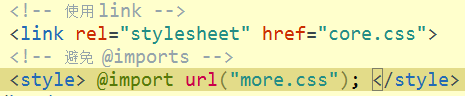
其他属性只在组件 内部 起作用或者不会对前面两种情况的结果产生影响，所以他们排在后面。





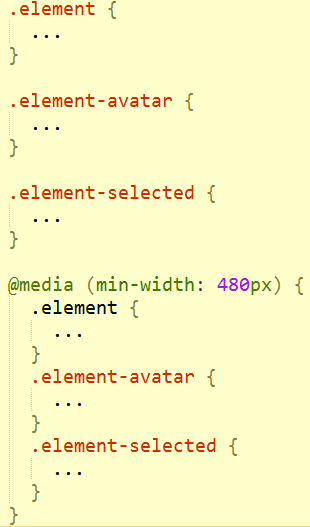
## 不要使用@import

因为使用@import引用的文件只有在引用它的那个css文件被下载、解析之后，浏览器才会知道还有另外一个css需要下载，这时才去下载，然后下载后开始解析、构建render tree等一系列操作。浏览器在页面所有css下载并解析完成后才会开始渲染页面，因此css @import引起的css解析延迟会加长页面留白期。 所以，要尽量避免使用css @import而尽量采用link标签的方式引入。



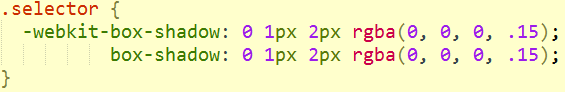
## 媒体查询位置

尽量将媒体查询的位置靠近他们相关的规则。不要将他们一起放到一个独立的样式文件中，或者丢在文档的最底部。这样做只会让大家以后更容易忘记他们。这里是一个典型的案例。



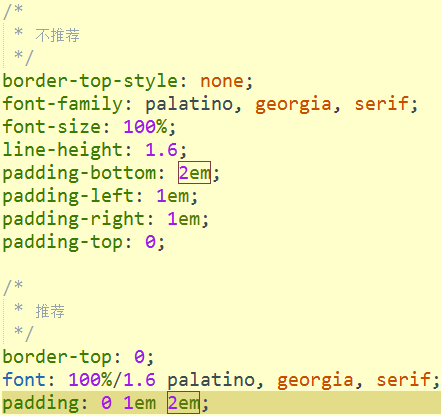
## 前缀属性

当使用厂商前缀属性时，通过缩进使取值垂直对齐以便多行编辑。



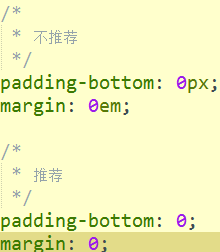
## 属性简写

CSS提供了各种缩写属性（如 font 字体）应该尽可能使用，即使在只设置一个值的情况下。善于使用缩写，可以有效的提高css样式的书写效率和代码阅读速度。使用缩写属性对于代码效率和可读性是有很有用的。



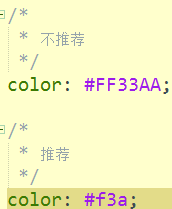
## 0 和 单位

省略“0”值后面的单位。不要在0值后面使用单位，除非有值。



## 十六进制表示法

在可能的情况下，使用3个字符的十六进制表示法。颜色值允许这样表示，3个字符的十六进制表示法更简短。始终使用小写的十六进制数字。

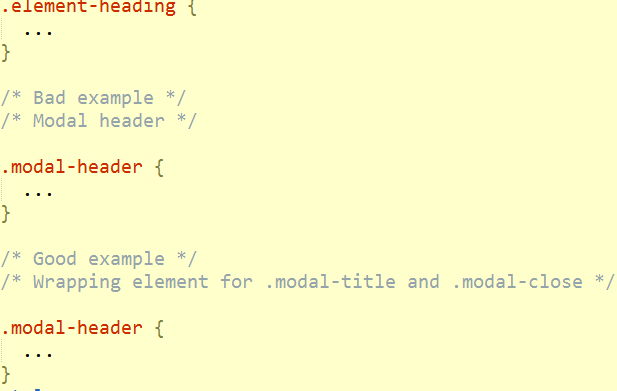


## LESS 和 SASS 中的嵌套

避免不必要的嵌套。可以进行嵌套，不意味着你应该这样做。只有在需要给父元素增加样式并且同时存在多个子元素时才需要考虑嵌套。

## 代码注释

代码是由人来编写和维护的。保证你的代码是描述性的，包含好的注释，并且容易被他人理解。好的代码注释传达上下文和目标。不要简单地重申组件或者 class 名称。



## Class 命名

保持 Class 命名为全小写，可以使用短划线（不要使用下划线和 camelCase 命名）。短划线应该作为相关类的自然间断。(例如，.btn 和 .btn-danger)。

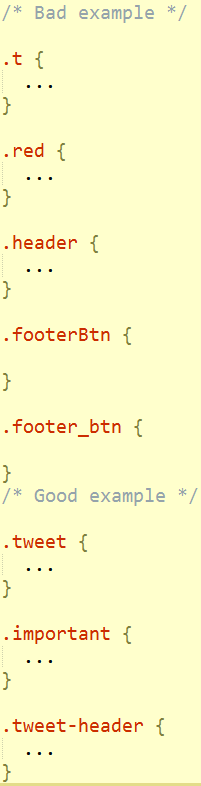
避免过度使用简写。.btn 可以很好地描述 button，但是 .s 不能代表任何元素。

Class 的命名应该尽量短，也要尽量明确。

使用有意义的名称；使用结构化或者作用目标相关，而不是抽象的名称。

命名时使用最近的父节点或者父 class 作为前缀。

使用 .js-\* classes 来表示行为(相对于样式)，但是不要在 CSS 中包含这些 classes。



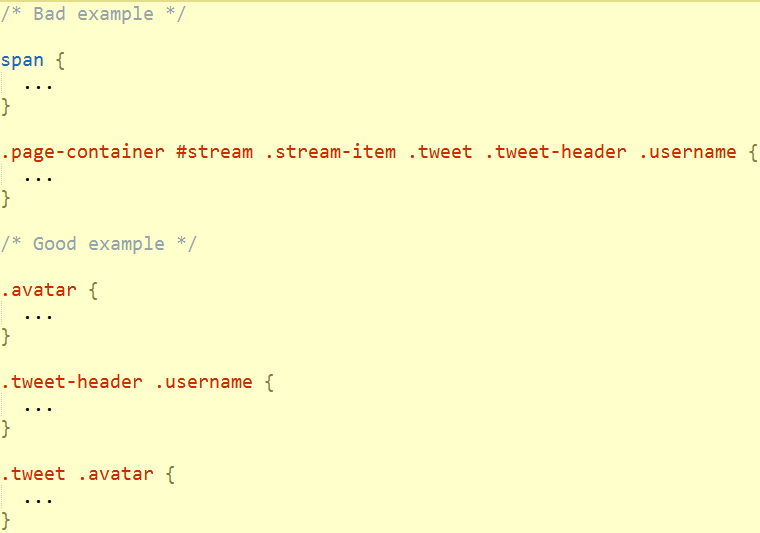
## 选择器

使用 classes 而不是通用元素标签来优化渲染性能。

避免在经常出现的组件中使用一些属性选择器 (例如，[class^="..."])。浏览器性能会受到这些情况的影响。

减少选择器的长度，每个组合选择器选择器的条目应该尽量控制在 3 个以内。

只在必要的情况下使用后代选择器 (例如，没有使用带前缀 classes 的情况).



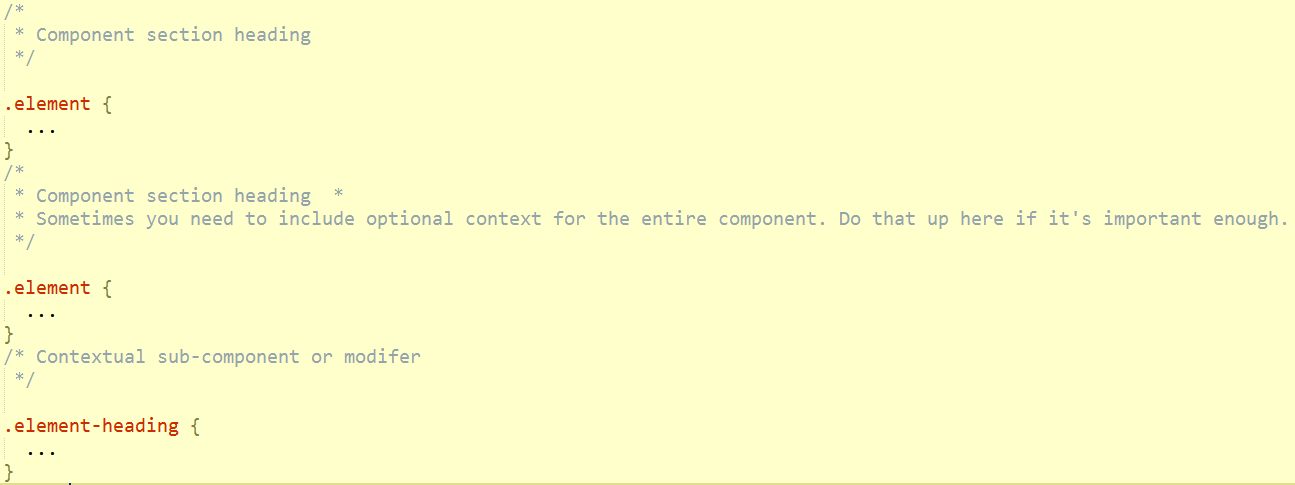
## 代码组织

以组件为单位组织代码。

制定一个一致的注释层级结构。

使用一致的空白来分割代码块，这样做在查看大的文档时更有优势。

当使用多个 CSS 文件时，通过组件而不是页面来区分他们。页面会被重新排列，而组件移动就可以了。



## 编辑器配置

根据以下的设置来配置你的编辑器，来避免常见的代码不一致和丑陋的 diffs。

使用四个空格的 soft-tabs。

在保存时删除尾部的空白字符。

设置文件编码为 UTF-8。

在文件结尾添加一个空白行。