选择器命名规范

要起直观有意义的名字，这对直观理解文档含义有很高作用。

有意义命名 即使没有注释也能立即清楚代码结构大概含义，语义化的重要性

**无意义的名字无论对于自己还是别人都没有好处，尤其后期对项目的维护以及团队合作时都非常不利！**

**BEM 命名原则**

 block：模块，名字单词间用 - 连接

 element：元素，模块的子元素，以下划线 \_\_ 与 block 连接

 modifier：修饰，模块的变体，定义特殊模块，以中划线 -- 与 block 连接

举例:

.block\_element{

}

.block-modifier{

}

**不允许有无意义的注释，否则回过头来看自己写的代码，自己也不知道是什么意思。**

但是下面这样是可以的



1.尽量用准确的单词描述功能

2.一般默认采用中划线-拆分过长描述。命名可以用，字母、数字、中划线-、下划线\_构成

3.可以采用驼峰命名，（不是很必要，css不区分大小写）。

4.要求命名用-中划线分割字段，字段不允许超过三个

选择器使用规范

语义化标签：目的让标签更直观的表现标签带来的含义。（本身就是div，只是作用和意义不一样）

语义化标签指的是，像H1~H6,列表(li),强调(strong em)等等这样的，标签有自己的含义的。

常见语义化标签：

<title>：页面主体内容

<hn>：h1~h6，分级标题,<h1>与<titlte>协调有利于搜索引擎优化

<ul>:无序列表

<ol>:有序列表

<header>:页眉通常包括网站标志、主导航、全站连接以及搜索框。

<nav>:标记导航，仅对文档中重要的链接群使用。

<main>:页面主要内容，一个页面只能使用一次。如果是web应用，则包围其主要功能。

<article>定义外部内容，

<section>定义文档中的内容的节(section、区段)。比如章节、页眉、页脚、或文档中的其他部分。

<aside>：定义其所处内容之外的内容。如侧栏、文章的一组链接、广告、友情链接、相关产品列表等。

<footer>：页脚，只有当父级是body时，才是整个页面的页脚。

<strong>：和 em 标签一样，用于强调文本，但它强调的程度更强一些。

<em>：将其中的文本表示为强调的内容，表现为斜体。

<progress>：定义运行中的进度（进程）。

而标签语义化指的是，尽量使用有相对应结构的含义的HTML的标签。

作用是：结构更好、更理由搜索引擎的抓取(SEO的优化) 和开发人员的维护可维护性更高，因为结构清晰,so易于阅读)。

更有利于特殊终端的阅读(手机，个人助理等）

**有效使用css选择器**

有效使用 css 选择器，因遵循以下原则

*  保持简单，不要使用嵌套过多过于复杂的选择器。
*  通配符和属性选择器效率最低，需要匹配的元素最多，尽量避免使用。
*  不要使用类选择器和 ID 选择器修饰元素标签。
*  不要为了追求速度而放弃可读性与可维护性
*  避免使用 CSS 表达式。

选择器嵌套应少于 3 级

/\* bad \*/

.page .header .login #username input {

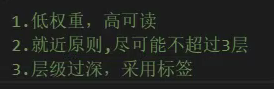
}

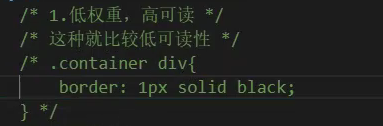
/\* good \*/

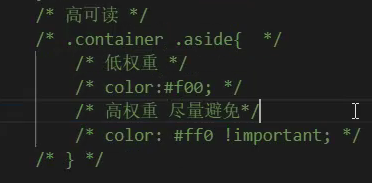
#username input {

}

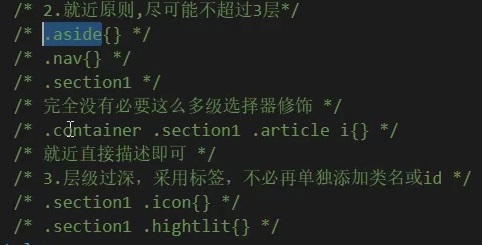
低权重，是优先级有关的内容







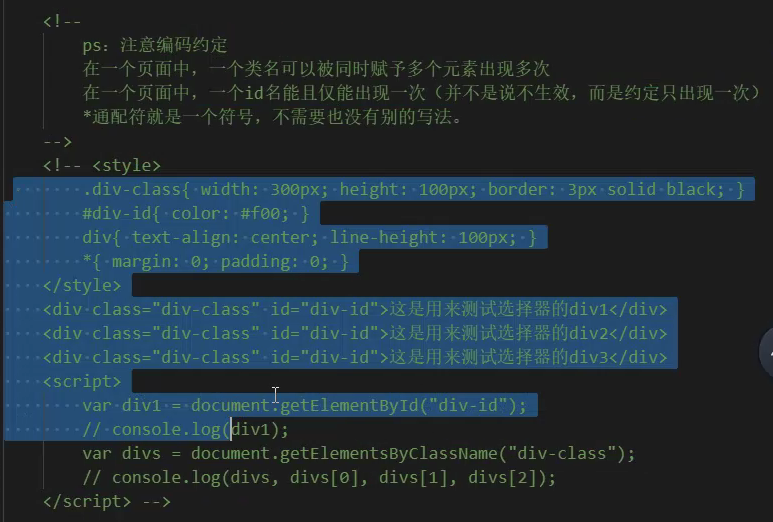
如果添加了!import表示必然生效，因为它是高权重。但是要尽量避免写它。



关于id选择器的问题：

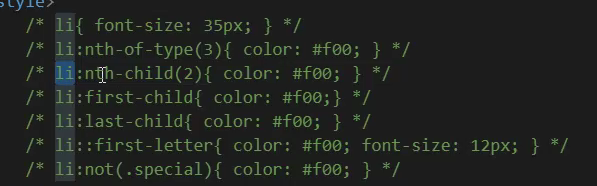
在一个页面中，一个类名可以被同时赋予多个元素出现多次。

在一个页面中，一个id名能且仅能出现一次（并不是不生效，而是约定只出现一次）



包含选择器和子选择器的区别：

选中某个元素，这个元素是其在父元素的第几个子元素



查看以上选择器之间的区别。

:nth-child(n)强调的是父元素下的子元素，并不在意元素的类型，然后再从这些严肃中进行筛选。

:nth-of-type(n)强调是同类型兄弟元素，然后再从这些元素里进行筛选。

静态类选择器

交互类选择器

1.悬停选择器：hover

触发选择器 ：active

关于hover的简单使用。



2.伪元素选择器：

::after向选定的元素之后插入内容

::before 向选定的元素前插入内容

content是伪元素里面必要的属性。

content：“这是before的内容”



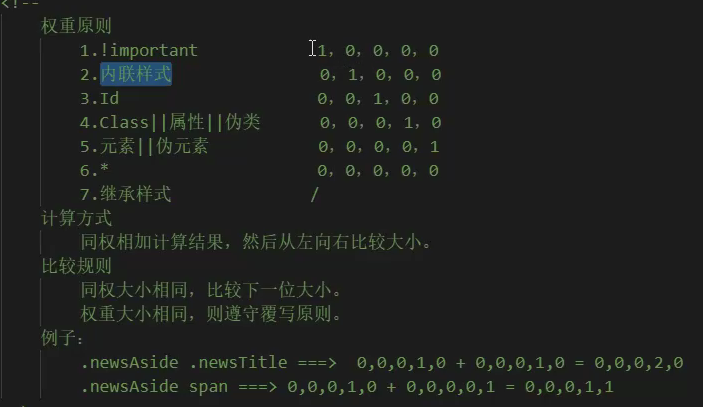
C:\Users\Administrator\Documents\Tencent Files\1027230111\Image\SharePic\20201102101831.png

伪元素清除浮动

用::after

伪元素是默认行内元素

覆写原则-权重问题



权重大小不同，谁的权重大谁生效。

权重大小相同，遵守覆写原则。

权重：！import>div选择器>class选择>标签选择器>\*通配符选择器