# CSS学习笔记二

1. 如何设置滚动条和如何设置滚动条的样式。

overflow

语法：

overflow : visible | auto | hidden | scroll ;

参数：

1.visible :不剪切内容也不添加滚动条。假如显式声明此默认值，对象将被剪切为包含对象的window或frame的大小。并且clip属性设置将失效

2.auto :此为body对象和textarea的默认值。在需要时剪切内容并添加滚动条

3.hidden :不显示超过对象尺寸的内容

4.scroll :总是显示滚动条

滚动条的样式。还有待测试验证。

scrollbar-face-color: #FFFFFF; ’滚动条凸出部分的颜色scrollbar-highlight-color: #FFFFFF; ’滚动条空白部分的颜色scrollbar-shadow-color: #FFFFFF; ’立体滚动条阴影的颜色scrollbar-3dlight-color: #FFFFFF; ’滚动条亮边的颜色scrollbar-arrow-color: #FF0000; ’上下按钮上三角箭头的颜色scrollbar-track-color: #FFFFFF; ’滚动条的背景颜色scrollbar-darkshadow-color: #FFFFFF; ’滚动条强阴影的颜色scrollbar-base-color: #FFFFFF; ’滚动条的基本颜色 如上设置出现的效果即为滚动条除上下三角外的其他位置都为白色，上下三角为红色。大家可以根据具体的需要来对滚动条的颜色进行修改！

1. 所有有关字体的样式：

3、所有有关背景的样式：

4、CSS样式中的 !important 的使用。

使用!important的css定义是拥有最高的优先级的。比内联样式都高。

同一个样式在一个类里定义两次。表示会优先调用后面加了 !important的样式。

.testClass{

color:blue !important;

color:red;  //还是现实蓝色，不会覆盖。

}

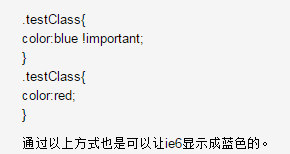
但是对于IE浏览器时情况如下：

.testClass{

color:blue !important;

color:red;  //后一个会覆盖前一个。

}



如上图。如果是分开写两次同一个类样式里的同一个属性，是不会覆盖的。

5、文件类型描述（Document Type Definition，DTD）

6、[**彻底弄懂css中单位px和em,rem的区别**](http://www.cnblogs.com/leejersey/p/3662612.html)

国内的设计师大都喜欢用px，而国外的[网站](http://www.html5cn.org/)大都喜欢用em和rem，

PX特点

1. IE无法调整那些使用px作为单位的字体大小；

2. 国外的大部分网站能够调整的原因在于其使用了em或rem作为字体单位；

3. Firefox能够调整px和em，rem，但是96%以上的中国网民使用IE浏览器(或内核)。

px像素（Pixel）。相对长度单位。像素px是相对于显示器屏幕分辨率而言的。(引自CSS2.0手册)

em是相对长度单位。相对于当前对象内文本的字体尺寸。如当前对行内文本的字体尺寸未被人为设置，则相对于浏览器的默认字体尺寸。(引自CSS2.0手册)

       任意浏览器的默认字体高都是16px。所有未经调整的浏览器都符合: 1em=16px。那么12px=0.75em,10px=0.625em。为了简化font-size的换算，需要在css中的body选择器中声明Font-size=62.5%，这就使em值变为 16px\*62.5%=10px, 这样12px=1.2em, 10px=1em, 也就是说只需要将你的原来的px数值除以10，然后换上em作为单位就行了。

EM特点

1. em的值并不是固定的；

2. em会继承父级元素的字体大小。

所以我们在写CSS的时候，需要注意两点：

1. body选择器中声明Font-size=62.5%；

2. 将你的原来的px数值除以10，然后换上em作为单位；

3. 重新计算那些被放大的字体的em数值。避免字体大小的重复声明。

rem特点

        rem是[CSS3](http://www.html5cn.org/portal.php?mod=list&catid=16)新增的一个相对单位（root em，根em），这个单位引起了广泛关注。这个单位与em有什么区别呢？区别在于使用rem为元素设定字体大小时，仍然是相对大小，但相对的只是HTML根元素。这个单位可谓集相对大小和绝对大小的优点于一身，通过它既可以做到只修改根元素就成比例地调整所有字体大小，又可以避免字体大小逐层复合的连锁反应。目前，除了IE8及更早版本外，所有浏览器均已支持rem。对于不支持它的浏览器，应对方法也很简单，就是多写一个绝对单位的声明。这些浏览器会忽略用rem设定的字体大小。