**ID,对某个目标的样式进行改变，单一目标，标识为#**

<style>

#s{color:#900}

</style>

rel=”shortcut icon” **意为该文档与被连接文档之间的关系**

<h1 id="s">第一个标题</h1>

<id=“XX”>(在显示区域中)  **Class对某一组目标的样式进行改变可重复使用标识为 .（点）**

<style>

.ad{color:#039}

.cente{ text-align:center} 对子class后的子class前需要加 . （点）

</style> 对class下的定义的标签无需加 . （点）

<h1 class="ad">第一个标题</h1>

<h1 class="ad">第一个标题</h1>

<h1 class="cente" class="ad" >第一个标题</h1>

. bb{width:100%; height:600px; background-image:url(2460.JPG); background-repeat:repeat-y}

**对插入图片的排列方式为沿着X或者Y轴**

Background-color  **设置背景颜色**

Background-image 50px 10px **设置图片为背景 ；俩定位水平 垂直**

.bb{width:100%; height:10000px; background-image:url(2460.JPG);Background-attachment：fixed} **背景图片是否固定或者随着页面其他部分滚动**

.bb{width:100%;height:10000px;background-image:url(2460.JPG);Background-position: left right top bottom cent } 如果直接输入大小也可以指定位置 **设置背景的起始位置(上下左右中)**

Background-repeat **设置背景是否如何重复**

文本

Color  **更改颜色**

Letter-spacing:20px  **更改字间距**

Line-height:30px  **更改行高（上下文字30若字为12则上下各5）**

Text-align:center **文字水平对齐方式**

font-weight **文字加粗**

Text-decoration: underline **添加下划线的位置**

（underline 下划线）

（overline 上划线）

（link-through 穿过文本的一条线）

（blink 闪烁的文本）

（inherit 从父元素继承text-decoration的值）

Text-indent:50px  **首行文字的缩进**

.box{Text-shadow：10px 10px 10px #003c} **（1为左2为右3为模糊度 颜色）文本阴影**

Span{font-size:12px;vertical-align:sub;}

<h1 class=”box”>这个<span>文本</span>有下标</h1>  **对文字的进行下标上标的对齐**

Text-decoration：none **清除下划线**

White-space:nowrap  **对文字空白的处理**

(nowrap 不换行还在同一行继续，直到遇到<br>)

(pre-wrap 保留空白符序列，正常换行。)

(pre-line 合并空白符序列，保留换行符)

(inherit 规定 应该从父元素继承white-space属性的值）

Word-spacing：30px **减少单词与单词之间的留白而不是字母之间的适用于单词**

padding-left:500px; **移动框体，只移动上层，不可移动背景**

<style>

a:link{color:rgba(0,255,0,1)} 未访问过

a:visited{color:rgba(0,0,255,1)} 已访问过  **设置链接颜色**

a:hover{color:rgba(0,0,0,1)} 鼠标放在连接上 a:hover必须跟在a:link和a:visited后面

a:active{color:rgba(153,255,0,1)} 鼠标被点击那一刻 a:active必须跟在a:hover后面

</style> 也可设置鼠标经过时背景颜色改变

Border:0 none; **去除图片链接蓝框**

ul li {list-style-image:url(s4002.ico) ; width:10px; height:10px} **替换表格序号方法一**

<style>

ul{ list-style:none} **首先删除样式，然后设置背景图片并设置大小，然后设置合适的文字位置**

ul li { background:url(01.PNG) no-repeat;; background-size:20px 20px; text-indent:20px} **方法二**

</style>

List-style-position **列表标记与文本的关系**

Inside 列表项目在文本以内，且环绕文本根据标记对齐。

Outside 默认值，标记位于文本的左侧。项目标记在文本以外，且环绕文本不根据标记对齐。

Inherit 规定从父元素继承list-style-position属性的值。

List-style-type **设置列表标记的类型**

None 无标记 Disc 默认实心圆 Circle 空心圆 Square 数字

Decimal-leading-zero 0开头的数字标记（01.02.03等） Lower-roman 小写罗马数字

Upper-roman 大写罗马数字

盒子

padding:50px 10px 20px 30px； **填充属性定义元素边框与元素内容之间的空间**

可以分别设置padding-top:25px padding-bottom:25px padding-right:50px padding-left:50px

四个值：上，右，下，左

三个值：上，左右，下

两个值：上下，左右

Border:5px solid #003c； （**border）边框属性可以更改边框的大小，形状，颜色，也可以单独调试同上**

**如果是简写无标识与上值相同**

border-radius:50% **正圆角**

outline:50px solid #000000 **（outline）轮廓属性是绘制与元素周围的一条线，位于边框的外围，可以起到突出元素的作用使用方法与边框相同**

margin:100px 80px 90px 50px **（margin）外边距属性是用于与其他元素进行分开的透明区域**

**使用方法与填充属性相同**

注明：如果希望设计稿是XXpx，并设定了边框与内边距需要设定的宽为XX ---（二倍的边框加内边距）。长与此相同。

轮廓属性是向内增加的

H1,h2,h3,p{ color:#003c} **对一组标签进行统一更改仅需要加上，（逗号）即可**

子优先级高于全组优先级

Max-height **最大长度**

Max-widh **最大宽度**

Min-height **最小长度**

Min-widh **最小宽度**

Display: none隐藏空间 display :inline 转换块元素为行元素 disply:block将行元素转换为块元素

例 .aa:hover p{display:none} （：的作用是表明对分区内的某个标记所做的动作）

visbility:hidden 隐藏元素 visbility可见性

.aa:hover p{ visibility: hidden} **意为在鼠标经过aa时就会隐藏p标记内容** ***冒号***

.aa ul li a span:hover { } **意为在鼠标经过span时会隐藏span内容**

Display可以转化块元素和内联元素，block可以将内联元素转为块元素，当为inline的时候就将块元素转为内联元素。Inline-block它具有行元素和块元素共同的属性特点。

<span></span> 仅代表一个内联标记行元素，不换行，无其他属性

**内联元素没宽高不会另起一行，块元素有宽高会另起一行inline-block有宽高，但不另起一行**

Float：right 对齐仅对于所指定的标签有效，对标签内的无效 **向右浮动对元素有效常用于块元素**

**系统默认会配置一些参数，常需要在头部清零padding-marging等值，列表的小点也应该取消样式重做。**

Img{cal-align:middle} **图片垂直居中**

**行内块元素垂直居中**

vertical-align: middle

块状填充剩余空间并出现滚动条。

overfload:hiden;

height:100%;

position:abserlot;

定位

Position:static标准流

Position:relative相对定位对象遵循常规流，并且参照自身在常规流中的位置通过top，right，bottom，left这4个定位偏移属性进行偏移时不会影响常规流中的任何元素。

Position:absolute 绝对定位对象脱离常规流，此时偏移属性参照的是离自身最近的定位祖先元素，如果没有定位的祖先元素，则一直回溯到body元素。盒子的偏移位置不影响常规流中的任何元素，其margin不与其他任何margin折叠。Left top right bottm

假如无父级作为参照物，浏览器的坐上角就是他的0点（坐标原点）

Position:fixed; 与absolute一致，但偏移定位是以窗口为参考。当出现滚动条时，对象不会随着滚动。

z-index:XX设置定位图层显示优先级，数越大越顶层，越小越底层，必须有定位存在此属性才有效

假如父级需要排版在子元素顶层，父级不设定，子级需要z-index:-1；

平时做案例时，建议先对父级设置相对定位（参照物，他左上角就是原点），在对子级设置绝对定位，left top right bottom

过渡动画

transition:过度，综合属性：属性，过度时间，动画形式，延迟时间

transition-property:属性 all所有; 某一个属性width: 多个属性：逗号隔开，width，opactiy,color,boder;

transition-duration:过渡时间；0.5s;简写 .5s

transition-timimg-runction:ease 开始和结束慢，中间快，平滑 ease-in 开始慢 ease-out 结束慢

ease-in-out 和ease类似，但比ease幅度大。 linear匀速

transition-delay:1s 鼠标放上多久后出现动画

transition:all 1s ease-out过渡属性写在对象本身和写在经过出现过渡效果上面不同，往返不同

Transform:scale(1.1);缩放，大于1放大，小于1缩小

opacity:1 把整个对象（包含文本，图片，边框，）同时改变不透明度  
position:absolute;top:50%;left50%;transform:translate(-50%)无论块大小，css3中的水平居中，垂直居中的方法。

transition-delay:0.5s 延时播放

选择符

E:first-of-type 匹配同类型父元素的第一个子元素E

E:last-of-type 匹配同类型中的最后一个同级兄弟元素E。

E:only-of-type 匹配同类型中的唯一的一个同级兄弟元素E。

E:only-of-type 匹配同类型中的唯一的一个同级兄弟元素E。

E:nth-of-type(n) 匹配同类型中的第n个同级兄弟元素E

E:nth-last-of-type(n) 匹配同类型中的倒数第n个同级兄弟元素E。

li:nth-child(2n){color:#f00;} /\* 偶数 \*/ 对奇偶进行分别管理

li:nth-child(2n+1){color:#000;} /\* 奇数 \*/ 对其他方法也可

li:nth-child(n+4):nth-child(-n+8){color:#000;} 从4到8的选择区间。

超出文本隐藏

对文本区域设置宽高，溢出隐藏。多行文本显示 溢出出现省略号

overflow:hidden;

text-overflow:ellipsis;/\*超出文本出现省略号\*/

display:-webkit-box;/\*设置成一个弹性的盒子\*/

-webkit-line-clamp: 3;/\*只显示三行\*/

-webkit-box-orient:vertical;/\*垂直方向 竖向显示\*/

width:510px;

单行文本出现省略号

overflow:hidden;

text-overflow:ellipsis;

white-space:nowrap;/\*不换行\*/

width:510px;

cursor:pointer; 变成小手状

white-space-nomor

超出换行

word-wrap: break-word;  
word-break: normal;

flex属性

可以像bootstrap一样对内部的东西进行自由的编写。控制排列方式，控制排序方式。

使用flex-grow 可以比例分文件大小，若都为一则平分剩下的空间。

进度条

::-webkit-scrollbar{  
 width:0  
}

透视（用于3d效果）

**perspective**：1000，效果必须写在父级上，只能作用于3D效果。透视的3D效果

**perspective-origin**：**默认值**：50% 50%，效果等同于center center，更改旋转的固定位置支点。

**transform-style**：flat | preserve-3d 最重要的，指定某元素的子元素是（看起来）位于三维空间内，还是在该元素所在的平面内被扁平化。

**backface-visibility：**visible | hidden 背面不可见，设置，旋转后的背面时看不到的。

选择器权重

!importent权重最高。

其他是累加的形式计算的。

去除滚动条

::-webkit-scrollbar {

width: 0px;

height: 1px;

}::-webkit-scrollbar-thumb {

border-radius: 5px;

-webkit-box-shadow: inset 0 0 5px rgba(0, 0, 0, 0.2);

background: rgba(0, 0, 0, 0.2);

}

修改滚动条

::-webkit-scrollbar-track-piece {

background-color:#f5f5f5;

border-left:1px solid #d2d2d2;

}

::-webkit-scrollbar {

width:13px;

height:13px;

}

::-webkit-scrollbar-thumb {

background-color:#c2c2c2;

background-clip:padding-box;

border:1px solid #979797;

min-height:28px;

}

::-webkit-scrollbar-thumb:hover {

border:1px solid #636363;

background-color:#929292;

}

**IE滚动条**

scrollbar-arrow-color: color; /\*三角箭头的颜色\*/

scrollbar-face-color: color; /\*立体滚动条的颜色（包括箭头部分的背景色）\*/

scrollbar-3dlight-color: color; /\*立体滚动条亮边的颜色\*/

scrollbar-highlight-color: color; /\*滚动条的高亮颜色（左阴影？）\*/

scrollbar-shadow-color: color; /\*立体滚动条阴影的颜色\*/

scrollbar-darkshadow-color: color; /\*立体滚动条外阴影的颜色\*/

scrollbar-track-color: color; /\*立体滚动条背景颜色\*/

scrollbar-base-color: color; /\*滚动条的基色\*/

**相关内容**

IE是最早提供滚动条的样式支持，好多年了，但是其它浏览器一直没有支持，IE独孤求败了。

这些样式规则很简单：

scrollbar-arrow-color: color; /\*三角箭头的颜色\*/

scrollbar-face-color: color; /\*立体滚动条的颜色（包括箭头部分的背景色）\*/

scrollbar-3dlight-color: color; /\*立体滚动条亮边的颜色\*/

scrollbar-highlight-color: color; /\*滚动条的高亮颜色（左阴影？）\*/

scrollbar-shadow-color: color; /\*立体滚动条阴影的颜色\*/

scrollbar-darkshadow-color: color; /\*立体滚动条外阴影的颜色\*/

scrollbar-track-color: color; /\*立体滚动条背景颜色\*/

scrollbar-base-color: color; /\*滚动条的基色\*/

webkit不再是用简单的几个CSS属性，而是一坨的CSS伪元素：

-webkit-scrollbar 滚动条整体部分

-webkit-scrollbar-button 滚动条两端的按钮

-webkit-scrollbar-track 外层轨道

-webkit-scrollbar-track-piece 内层轨道，滚动条中间部分（除去）

-webkit-scrollbar-thumb （拖动条？滑块？滚动条里面可以拖动的那个，肿么翻译好呢？）

-webkit-scrollbar-corner 边角

-webkit-resizer 定义右下角拖动块的样式

通过这些伪元素，可以完全的重写一个网站的滚动条样式。

当然webkit提供的不止这些，还有很多伪类，可以更丰富滚动条样式：

:horizontal – horizontal伪类应用于水平方向的滚动条

:vertical – vertical伪类应用于竖直方向的滚动条

:decrement – decrement伪类应用于按钮和内层轨道（track piece）。它用来指示按钮或者内层轨道是否会减小视窗的位置（比如，垂直滚动条的上面，水平滚动条的左边。）

:increment – increment伪类和decrement类似，用来指示按钮或内层轨道是否会增大视窗的位置（比如，垂直滚动条的下面和水平滚动条的右边。）

:start – start伪类也应用于按钮和滑块。它用来定义对象是否放到滑块的前面。

:end – 类似于start伪类，标识对象是否放到滑块的后面。

:double-button – 该伪类以用于按钮和内层轨道。用于判断一个按钮是不是放在滚动条同一端的一对按钮中的一个。对于内层轨道来说，它表示内层轨道是否紧靠一对按钮。

:single-button – 类似于double-button伪类。对按钮来说，它用于判断一个按钮是否自己独立的在滚动条的一段。对内层轨道来说，它表示内层轨道是否紧靠一个single-button。

:no-button – 用于内层轨道，表示内层轨道是否要滚动到滚动条的终端，比如，滚动条两端没有按钮的时候。

:corner-present – 用于所有滚动条轨道，指示滚动条圆角是否显示。

:window-inactive – 用于所有的滚动条轨道，指示应用滚动条的某个页面容器（元素）是否当前被激活。（在webkit最近的版本中，该伪类也可以用于：：selection伪元素。webkit团队有计划扩展它并推动成为一个标准的伪类）

另外，:enabled、:disabled、:hover 和 :active 等伪类同样可以用于滚动条中。