在布局设置的时候，font-size 影响超级大；建议在body中，font-size = 62.5%，后面可以用em当作单位，1em = 10px，可以进行响应式设计;默认正常下字体大小为16px，62.5% = 10px 方便调试。

CSS的坐标轴体系，默认原点在元素的左上角

CSS 3D坐标轴体系，Z轴以观看者为参照，x轴从左到右，y轴从上到下，z轴从屏幕到观看者（正值移动表示元素向自己方向移动）

CSS定位

在html中，一切以框的形式出现

元素分为块级元素和行内元素，块级自带框，行内元素只有设置display：block；可以像块级元素一样显示，同时，display：none；可以让元素没有框，则不显示框和内容，不占用内容空间。

内容空间就类似于积木，由一个个框搭建成

无名框

<div>some text<p>Some more text</div>,其中，红色部分表示无名框，无法对其应用样式

定位机制：

普通流

按照在html中的顺序，块级框一个一个竖直排列

行内元素水平依次排列，一行称为一个行框，行框高度是能够包含所有行框内元素的高度

CSS position属性

Static：正常文档流方式生成框，行框在其父元素内

Relative：元素框偏离原来位置一定距离，原本所占的空间依然保留

Absolute：脱离文档流，相对于其包含块（可能是初始包含块）定位，原来所占空间释放,说白了就top，left起作用，定义这两个就可以

Fixed：类似于absolute，包含块为视窗

CSS overflow属性

对于固定的height和width，如果内容超出了固定宽和高度，通过overflow可以对该情况进行处理

Overflow：heidden；表示隐藏超出的部分

Overflow：scroll；表示给内容添加滚动条，通过滚动条来看溢出的内容

Overflow：auto；交给浏览器来决定如何处理

CSS clip属性

针对绝对定位的元素，定义一个矩形（绝对定位），在矩形内的可见矩形外的由overflow处理

默认不应用任何裁剪auto

垂直排列图像

vertical-align：text-top 图像顶部与文字平行

: text-bottom 图像底部与文字平行

CSS Z-index属性

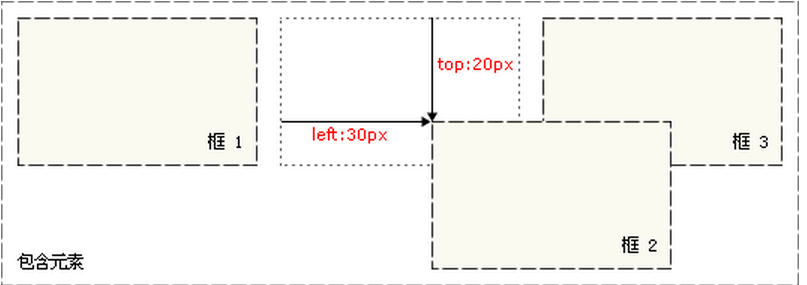
设置堆叠属性，Z轴方向，即垂直显示器平面的方向

属性值高的浮在低的上面，效果类似于word中的图片衬与文字下方

属性值可以为负值

仅能在定位元素上使用

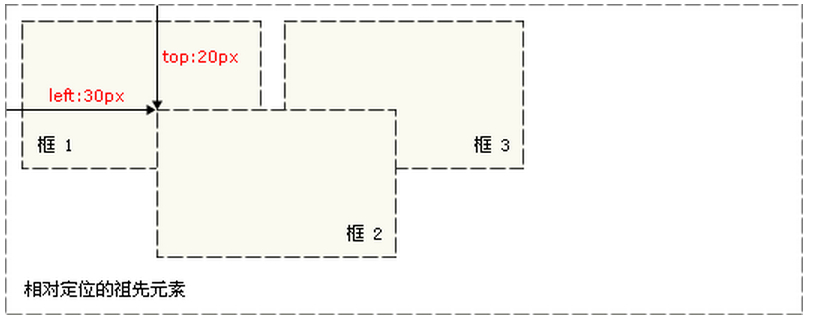
CSS 相对定位



元素仍占据原来的空间，移动可能造成元素覆盖

CSS绝对定位

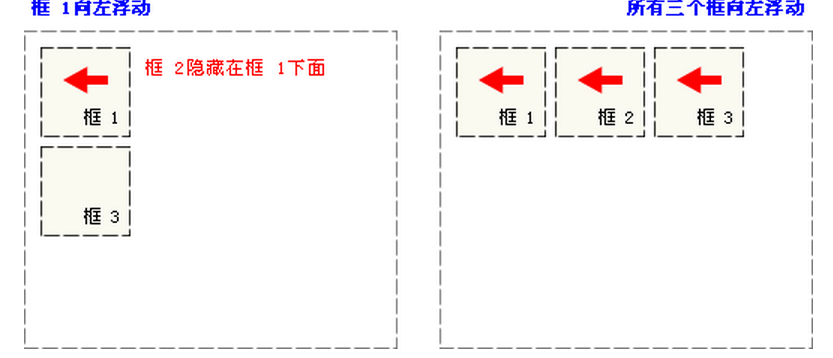
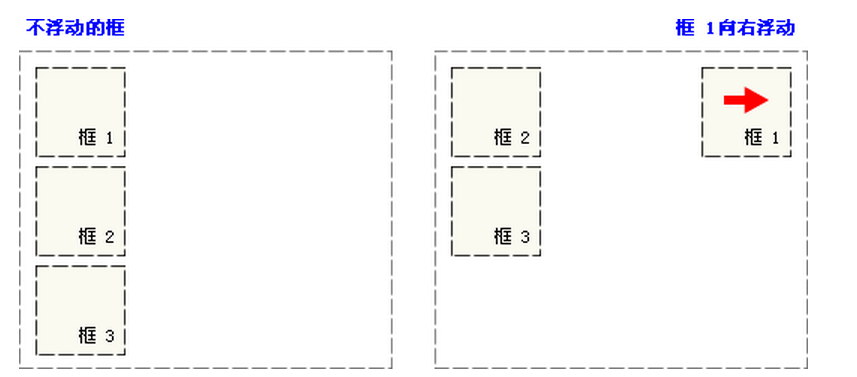
元素脱离原来文档流，之前占据的空间被释放，相对于**最近的已定位祖先元素**，如果元素没有已定位的祖先元素，那么它的位置相对于**最初的包含块，可以任意覆盖**

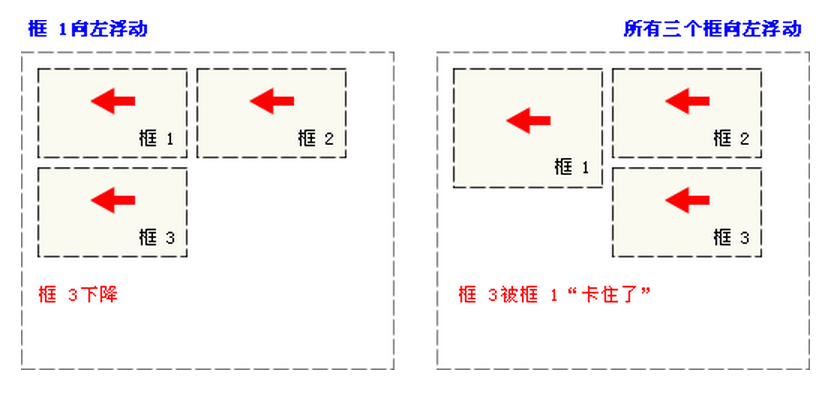


CSS float属性

Float后，元素脱离文档流，直到外边缘碰到包含框或者另一个浮动框的框为止

Right，left两个值

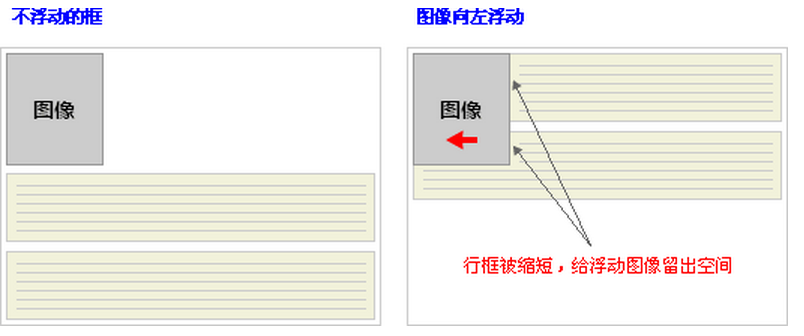




一行位置不够，后面的浮动元素向下浮动，元素高度不同时，可能造成卡住的情况

行框的清理

设置浮动框可以使行框围绕浮动框，效果类似文字环绕图片



取消文字环绕效果，设置clear属性

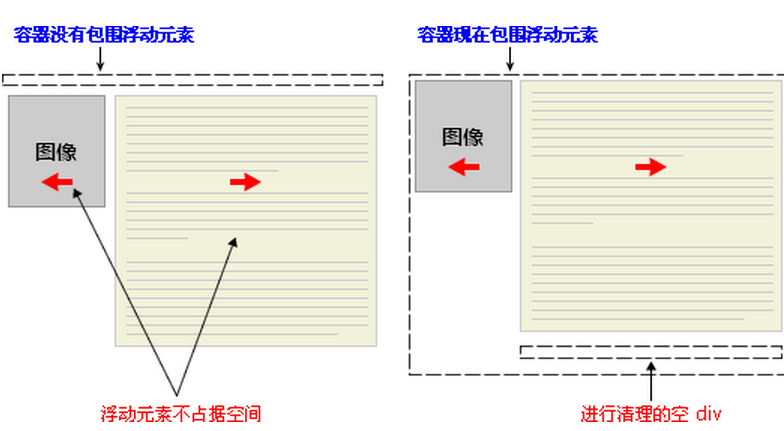
Clear: left, right 表示被设置该属性的行框哪边不应挨着浮动框

增加上外边距达到该效果



子元素全部设置浮动，会使元素脱离容器，增加一个元素专门用来设置clear来避免这种情况

直接让容器浮动同样可以避免这种情况，浮动会得到继承



CSS对齐

块元素，占据全部可用宽度，前后会自动换行P, div…

Margin: auto 设置成居中效果，均分可用外边距

必须声明!doctype

Width不能是100%

Position 属性 relative，absolute，然后设置水平位置

必须声明！doctype

取消body默认设置body{margin：0px; padding: 0px;}

解决兼容问题

Float 属性 right，left设置水平位置

CSS尺寸

图像尺寸

图像也是元素，改变图像的方法同样适用其他块模型

Img{height: 30px; width: 30px;} 像素形式

Img{ height:30% ; width: 30%;} 百分比形式

百分比是相对于视窗（父元素）的高度和宽度

最大宽度和最大高度

Max-width: 300px; max-width:50%

Max-height: 300px; max-height: 50%;

元素最多占据最大高度和最大宽度决定的空间，超出部分可能会被后面的块覆盖

最小高度和最小宽度

Min-height: 100px; min-height:100%

Min-width:100px; min-width:100%

元素至少占据最小宽度和最小高度决定的空间

行间距

Line-height : 100%; 表示行间距是字体大小的100%，一般默认为110%~120%

50px; 表示行间距为固定的50px；

5; 表示行间距为: 字体大小\*5；

CSS分类

显示为内联或者块级

Display: inline; 显示为内联元素，形成行框

Display：none； 不显示元素

Display：block； 显示为块级元素，按照在html中位置从上到下竖直排列，中间自动换行

元素可见/不可见

Visibility: visible; 可见

Visibility: hidden; 不可见

Visibility：collapse；表格对应的行或列重叠在一起，内容消失，原来的边框消失

给表格专用的visiblity：hidden；

改变光标

Cursor：光标的样式

Cursor：Pointer； 手

Cursor: crosshair; 十字

Cursor: auto; 箭头

个人觉得没必要改变光标，因为用起来一定很烦

制作导航栏

用ul和li，a元素构建基本的导航栏内容

ul{ list-style-type:none; margin:0; padding:0;}

消除浏览器默认样式和内外边距

竖直导航栏

a{display: block; width: 60px;}

所有的链接竖直表示，整个块都可以点击而不仅仅是链接文字；必须给框设置宽度，否则会出现意想不到的错误

水平导航栏

a{ display: inline;}

链接不一定有相同长度，因为链接文字长度的不同

Li{ float: left;}

a{ display: block; width:60px; }

block使得链接以块的形式显示，且整个块都可以点击

块元素默认width是整个宽度，浮动无法使彼此相邻，必须设置宽度

CSS图像透明

Opacity：0.4；

该属性规定图像的透明程度 取值 0~1

Filter：alpha（opacity = 40）；

针对IE8和以前的用法 取值0~100

CSS应用于不同媒介类型

@media 媒介类型{ CSS内容 }

媒介类型：print 打印 screen 显示器 tv电视机。。。

CSS3

选择器

框模型

背景和边框

文字样式

2-D，3-D转化

动画

多列布局

用户界面

边框

圆角边框

Border-radius: 25px;

Moz-border-radius : 25px; 对应老版本的firefox

属性值越大，圆角越接近圆

默认为0

边框阴影

box-shadow: h，v，blur，spread，color，inset

H:水平阴影的位置，允许负值，相对于块的左上角（原点）的距离

V：竖直阴影的位置，允许负值

Blur：可选属性 模糊距离，从外向内开始模糊

Spread：可选，阴影尺寸，在默认阴影大小的基础上扩展的尺寸

Color：可选，阴影颜色

Inset：可选，外部阴影变成内部阴影

Border-image

把元素包在这个图片里面

IE不支持，opera需要前缀 -o-，safari需要-webki-前缀

Border-image: url, slice, width, outset, repeat

Slice: 图片边框向内偏移，默认100%

Width：图片边框宽度

Outset：图片超出所包的元素的边框的的量

Repeat：repeat平铺，rounded：铺满，stretched：拉伸

举例：border-image: url(), 30, 30, round;

背景

规定背景图片尺寸大小

Background- size: 100px,100px;

Background-size: 100%,100%; 相对于其父元素

规定背景图片的定位区域

Background-origin: border-box; 元素border的左上角作为原点

Padding-box; 和borer-box的效果基本一样

Content-box; 元素内容区域的左上角作为原点

设置多个背景图片

Background-image: url(1), url(2);

Background-clip

规定背景的绘制区域

Background-clip: border-box padding-box content-box

把背景图片剪裁到指定区域，背景区域一定比border-box大

CSS文本效果

IE9以前的版本不支持

文本阴影

Text-shadow : h, v, blur, color

H: 水平阴影的位置 必选 相对于块的左上角（原点）的距离

V：垂直阴影的位置 必选

Blur：模糊距离 选填 从外向内开始模糊

Color：阴影颜色 选填

CSS自动换行

Word-wrap: break-word;

长文本不会超出区域，而是自动换行，保证所有文字都在区域内

Text-overflow:

文本长度超过包含元素时的处理方法

Text-overflow：clip；直接剪掉超出的文本

Text-overflow: ellipsis 用省略号代替超出的部分

CSS文本

@font-face

可以把需要的字体放到服务器上，如果用户的浏览器没有该字体，可以从服务器上下载这个字体

Firefox，safari, chrome支持.tff和.otf字体。IE9支持.eot字体；IE8和以前的版本不支持@font-face

<style>

@font-face{

Font-family: yourFont;

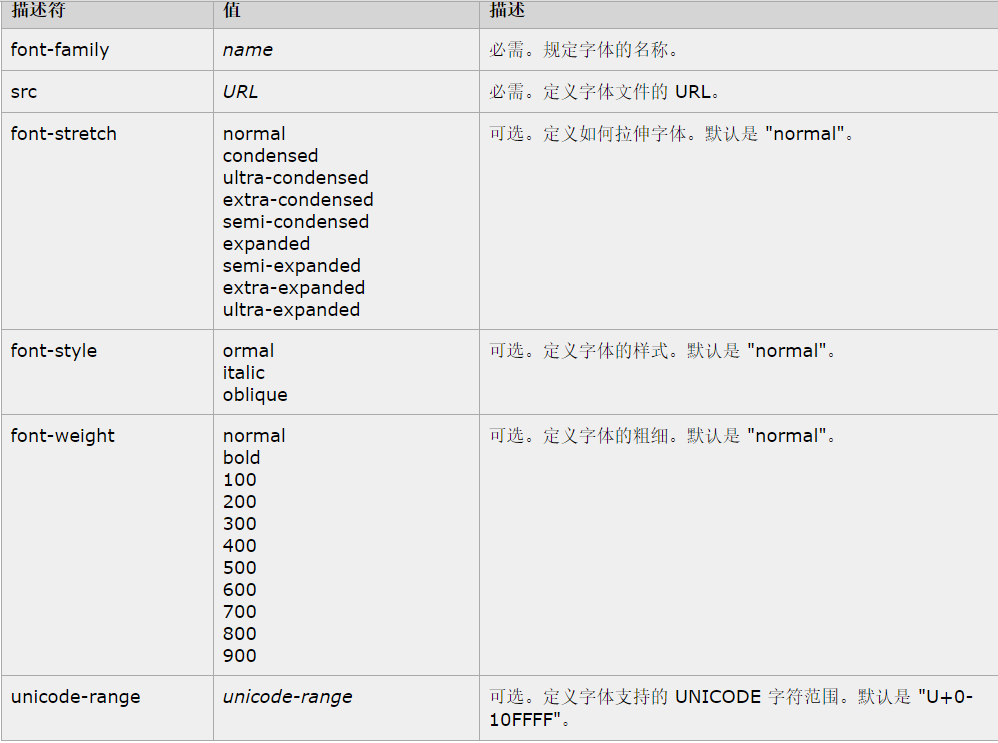
Src: url(‘xxx.ttf’);

Src: url(‘xxx.eot’); 保证兼容性

Font-weight: bold;

...只要是设置字体的属性都可以添加进来

}



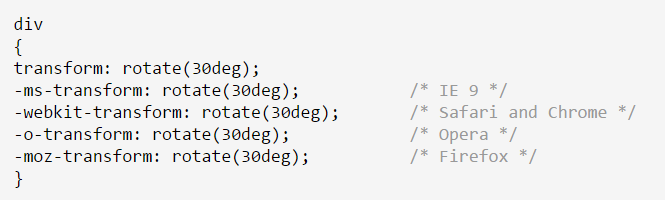
CSS 2D转换

Transform：

移动，缩放，转动，拉长，拉伸。

二维平面上的元素操作

兼容性：



Translate(100px,50px)

元素向右移动100px，向下移动50px；

translateX 沿x轴移动

translateY 沿Y轴移动

Rotate（30deg）

元素顺时针旋转30度

可以是负值，表示逆时针旋转

Scale(2,4)

元素宽度变成原来两倍，高度变为原来4倍

scaleX 仅宽度变化

scaleY 仅高度变化

Skew(20deg,30deg)

元素相对x轴逆时针旋转20度，相对y轴逆时针旋转30度

视窗左上角为原点，水平向右为x轴，竖直向下为y轴

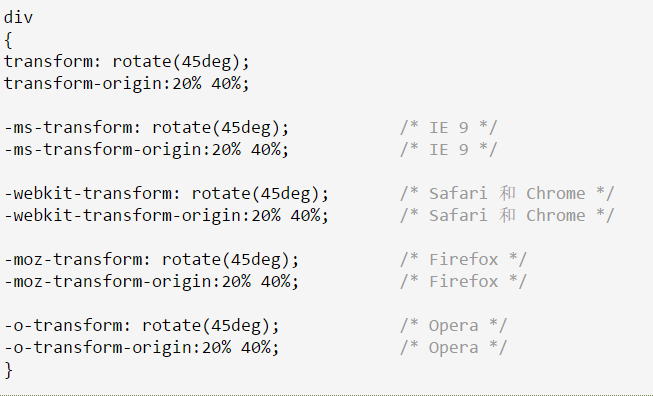
skewX 仅相对X轴旋转

skewY 仅相对Y轴旋转

transform-origin

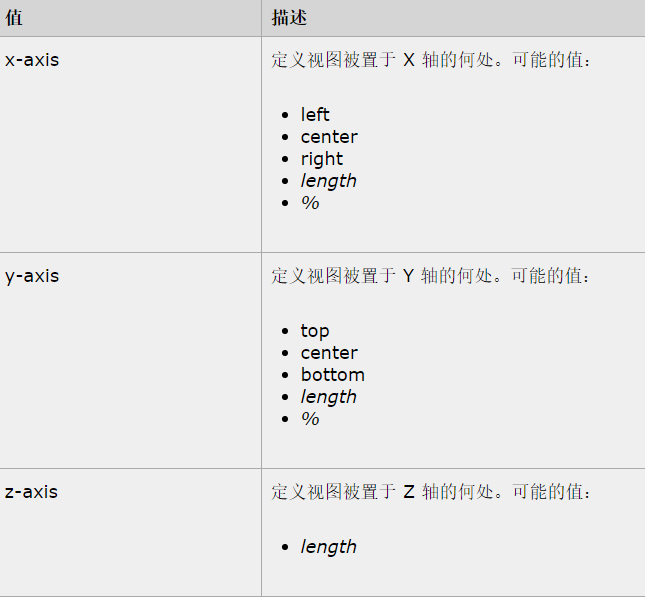
改变元素变化的原点位置

兼容性：



默认

Transform-origin: 50% 50% 0;



3D转换

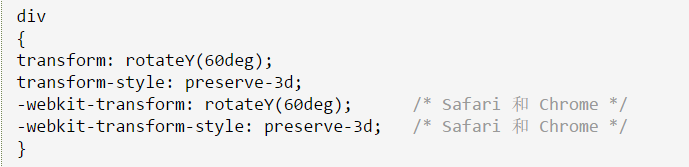
transform:rotateX(120deg) 元素围绕其x轴旋转120度

transform:rotateY(120deg) 元素围绕其Y轴旋转120度

兼容性：



Transform-style:





被转换的子元素是否保留其3D转换，如何在3D空间中呈现子元素

Transform-style:flat 不保留3D位置

Transform-style:preserve-3d 保留3D位置

perspective

主流浏览器都不支持，chrome和safari支持webkit-perspective:

元素距离视图的距离，Z轴方向的距离

只对3D元素起作用

默认为none

perspective:500 距离视图500像素

子元素会产生透视效果，元素本身没啥反应

perspective-origin：

主流浏览器都不支持，chrome和safari支持webkit-perspective:

设置3D元素的基点位置

用法和transform-origin一样

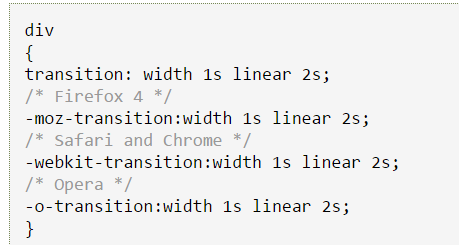
必须和perspective一起用，子元素产生透视效果

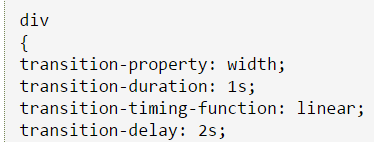
默认perspective-origin：50%，50%

transition

属性值变化的时候，从一种样式过度到另一种样式，设置变化过程

IE9及以前的版本不支持，chrome25及更早版本需添加-webkit-

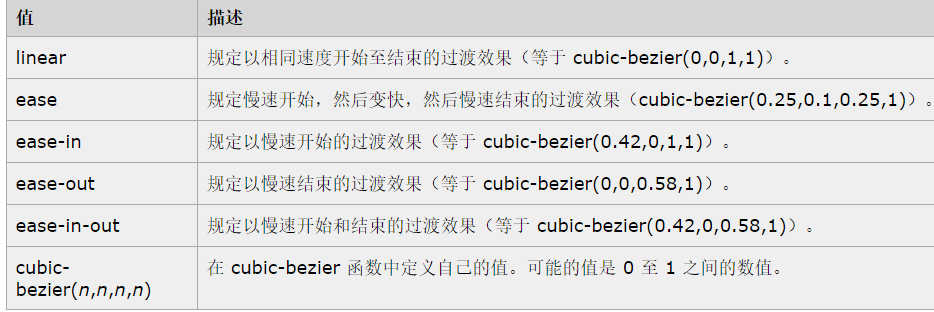




transition-timing-fucntion

规定过度效果的曲线





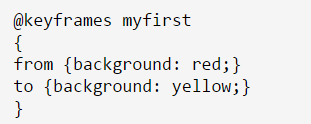
CSS动画

@keyframes + animation

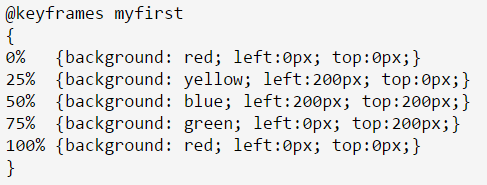


IE9及以前的版本不支持这俩个

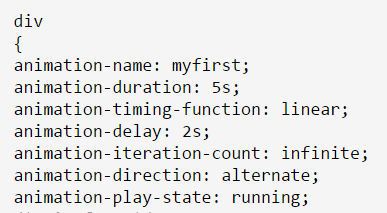
@keyframe创建动画

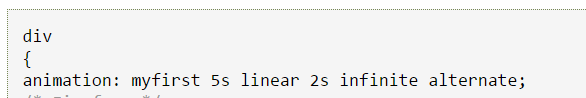


动画即从一种样式变化成另一种样式，可以设置任意多的样式和任意多的变化次数，用百分比规定变化的时间，from表示0%，动画开始；to表示100%，动画结束。为了得到最佳浏览器支持，必须设置0%和100%



用animation与元素绑定





动画名字和时长必须定义（前两项必须定义，后面的可省略）

animation-iteration-count:

表示动画执行次数



animation-iteration-count: infinite表示无限循环执行

animation-direcion:

规定是否轮流反向播放动画



animation-direction: normal 正常播放

animation-direction: alternate 135正常246反向

次数必须大于1该属性才有效

animation-play-state

规定动画运行还是暂停



animation-play-state： paused动画暂停

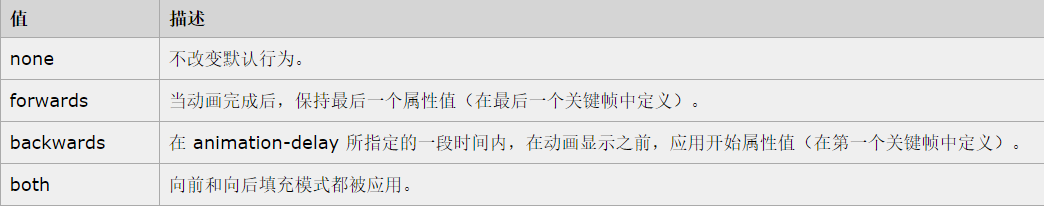
animation-play-state：running动画播放

animation-fill-mode

动画播放前或者播放后元素状态



animation-fill-mode：可以多个关键词用逗号分隔



CSS多列

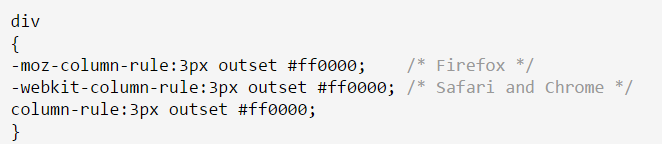
创建多个列来对文本进行布局，类似报纸



column-count规定列的个数

column-gap规定列之间的间隔

column-rule规定列之间的间隔，样式，颜色



第二个属性表示列之间的规则

默认为none





column-span

表示元素横跨列的个数



column-span : 1

column-span : all

只有这两个值

column-width

规定列的宽度

column-width : auto 浏览器自己决定

column-width : length 规定列的宽度

CSS3 用户界面



resize：

用户可以调整元素大小



outline-offset：

轮廓偏移

