css3样式

浏览器前缀

chrome+safari+opera(新版):-webkit-

firefox: -moz-

IE:-ms-

**1.border-radius**

border-radius 圆角(半径)

border-top-left-radius

border-top-right-radius

border-bottom-right-radius

border-bottom-left-radius

border-radius:10px(四个角);

border-radius:10px(左上，右下) 30px(右上、左下);

border-radius:10px(左上) 30px(右上、左下) 50px(右下);

border-radius:10px(左上) 30px(右上) 50px(右下) 70px(左下);

**2.box-shadow**

inset (可选参数)

x偏移

y偏移

blur模糊半径

扩展半径(可选参数)

color 颜色

阴影可以有多层，每层阴影中间以“,”隔开，先写的在上边有写的在下边

3.text-shadow:2px 2px 1px #000, 文字阴影

-webkit-text-stroke 文字描边 描边宽度 描边颜色（目前只有webkit浏览器支持）

**4.rgba()**

css3新增颜色模式

rgb+alpha

rgba 用在边框透明的时候，会看到边框也会显示元素的背景，（背景颜色透明，但元素内容不透明）

rgba(0,0,0,.5)

**5.选择符**

结构性伪类

.box p:nth-child(1){background:Red;}

找到box下的p标签 的父级下，第一个子元素,并且标签类型还必须是p

找到奇数行 .box \*:nth-child(odd) {background: red;}

找到偶数行 .box \*:nth-child(even) {background: red;}

----------------------------------------------

.box h1:nth-of-type(2) {background: red;}

找到.box下的h1标签的父级下的第2个h1标签

----------------------------------------------

.box p:first-child 相当于 nth-child(1)

.box p:last-child 相当于 nth-last-child(1)

.box h1:first-of-type 相当于 nth-of-type(1)

.box h1:last-of-type 相当于 nth-last-of-type(1)

----------------------------------------------

.box:before {content: "妙味趣学";}

.box:after {content: "妙味课堂";}

.box:first-letter {font-size: 50px;}

.box:first-line {color: red;}

content添加的内容 叫做伪元素

.box:first-letter 找到元素中的第一个字符

.................................................

**属性选选择符**

E[attr]只使用属性名，但没有确定任何属性值

E[attr="value"]指定属性名，并指定了该属性的属性值

E[attr~="value"]指定属性名，并且具有属性值，此属性值是一个词列表，并且以空格隔开，其中词列表中包含了一个value词，而且等号前面的“?”不能不写

E[attr^="value"]指定了属性名，并且有属性值，属性值是以value开头的

E[attr$="value"]指定了属性名，并且有属性值，而且属性值是以value结束的

E[attr\*="value"]指定了属性名，并且有属性值，而且属值中包含了value

E[attr|="value"]指定了属性名，并且属性值是value或者以“value-”开头的值（比如说zh-cn）

**6.背景渐变**

背景渐变是background-image的样式

1）**线性渐变 linear-gradient**

方位 默认top，left, left top

background: -webkit-linear-gradient(left top, red, blue, yellow);

background: -moz-linear-gradient(left top, red, blue, yellow);

background: -ms-linear-gradient(left top, red, blue, yellow);

background: linear-gradient(left top, red, blue, yellow);

（使用方位来确定方向时要加内核前缀）

角度

background: linear-gradient(0deg, red, blue, yellow);

0deg 从下到上 顺时针旋转

渐变点

background: linear-gradient(45deg, red 20%, blue 60%, yellow 100%);

渐变点:1. 具体数值 2. 百分比 （从何处渐变过渡到下一个颜色，到下一个渐变点完全变为下一个颜色）

IE6 7 8的背景渐变，只能设置2个颜色，且只能设置两个方向

filter:progid:DXImageTransform.Microsoft.gradient(startColorstr='red',endColorstr='yellow',GradientType='1');

方向只有0和1这两个值：GradientType='1'

线性渐变重复

repeating-linear-gradient(90deg, red 0, red 20px, #EEE 20px,#EEE 40px);

2）径向渐变 radial-gradient

radial-gradient([<起点>]? [<形状> || <大小>,]? <点>, <点>…);

起点：可以是关键字（left,top,right,bottom），具体数值或百分比

形状： ellipse、circle

大小 :具体数值或百分比（即x轴和y轴的半径），也可以是关键字（最近端，最近角，最远端，最远角，包含或覆盖 (closest-side, closest-corner, farthest-side, farthest-corner, contain or cover)）;注意firefox目前只支持关键字

例：

background: -webkit-radial-gradient(left top,100px 80px, red 10%, blue 100%);

background: -ms-radial-gradient(left top,100px 80px, red 10%, blue 100%);

background: -moz-radial-gradient(left top,farthest-corner, red 10%, blue 100%);

**7.background 背景**

同一个元素可以添加多张背景图，先写的在上边，后写的在下边，如果还有背景颜色，背景颜色写在最后一个的后面

例：background: linear-gradient(0deg, rgba(0, 0, 0, .8), rgba(0, 0, 0, 0)), url(img/2.jpg) no-repeat 0 0 red;

1）background-clip 背景裁切

border-box 从边框区域显示背景

padding-box 从padding区域显示背景

content-box 从content区域显示背景

另： -webkit-text 文字区域显示背景 （webkit内核私有样式）

2 )background-size 背景大小

contain 包含 （元素包含整张背景图）

cover 覆盖 （铺满元素）

auto 例：background-size: 400px auto;即高度会等比放大

补充：background: url(img/2.jpg) no-repeat 0 0 / 400px auto;这样的写法有些移动端不支持。

3 )background-origin 背景基点

背景基点的0，0点默认是从padding的左上角。

border-box 基点为边框的左上角

padding-box 基点为padding的左上角 （默认值）

content-box 基点为内容的左上角

**8.animation 动画**

1）关键帧

@keyframes move {

0% {

left: 0;

top: 0;

}

100% {

left: 300px;

top: 300px;

}

move是关键帧的名称。

0%是初始位置，100%是结束位置,在关键帧中没有起始位置或者结束位置，默认会都以元素的初始位置为准。

（若用于移动端，需要加内核前缀）

2）调用关键帧

E｛animation: move 2s 1s infinite cubic-bezier(0.825, -0.590, 0.320, 1.495);｝

1）animation （若用于移动端，需要加内核前缀）

必要属性animation-name 动画名称（关键帧名称)

animation-duration 动画持续时间

可选属性animation-delay 动画延迟 （动画延迟）

animation-iteration-count 重复次数 （数字 或者 infinite-无限次）

animation-direction 偶数次的播放顺序 （alternate 倒序播放;normal顺序播放：默认值）

animation-timing-function 动画运动形式

linear匀速

ease 缓冲

ease-in 由慢到快

ease-out 由快到慢

ease-in-out 由慢到快再到慢。

cubic-bezier(number, number, number, number)：特定的贝塞尔曲线类型，4个数值需在[0, 1]区间内

animation-fill-mode 动画播放之前或之后，其动画效果是否可见

backwards 动画执行之前，位置停留在第一帧

forwards 动画执行之后，位置停留在最后一帧

both 动画执行之前，位置停留在第一帧,动画执行之后，位置停留在最后一帧

2） animation-play-state:paused; 动画暂停 （注意：此属性不是animation的子属性）

**8.transition 过渡**

transition 用在给元素的数值类的样式发生改变的时候，添加过渡动画的

1）transition属性

transition-property 要运动的样式 （all || [attr] || none）

transition-duration 运动时间

transition-delay 延迟时间

transition-timing-function 运动形式 （与animation-timing-function的属性值相同）

例： transition:｛ 2s 1s width, 1s height;} （注意：多条样式的加过渡时中间用“，”隔开）

**9.transform 变换**

rotate 旋转

scale 缩放 （单位：数字）

scaleX X轴缩放

scaleY Y轴缩放

skewX X轴斜切 （单位：deg)

skewY Y轴斜切

skew(skewX,skewY);

translateX X轴位移 (单位：px)

translateY Y轴位移

translateZ Z轴位移

translate(translateX,translateY);

transform 3D变换

rotateX 围绕着x轴翻转

rotateY 围绕着y轴翻转

rotateZ 围绕着Z轴翻转

;adfkl

perspective 景深

perspective-origin 景深基点 （可以为像素或者百分比）

transform-origin: center center; 变换基点 transform-origin：left/right/top/bottom 上下左右边为变换基点。

transform-style: preserve-3d; 把元素的内容 变换成3d空间

transform 不脱离文档流

transform 的层级比普通元素高，但是比定位元素低

transform 移走之后，原始位置一样会被保留

transform 不会影响元素的占位大小