## css3新特性

css3选择器

背景和边框

文字特效

2D/3D转换

动画

多列布局

用户界面

# 第一章 css3选择器

## ■基本选择器

|  |  |  |  |
| --- | --- | --- | --- |
| 选择器 | 名称 | 简介 | 版本 |
| \* | 通配选择器 | 匹配所有元素 | css2 |
| element | 标记选择器 | 匹配指定标记的元素 | Css1 |
| .class | 类别选择器 | 匹配class属性值相匹配的元素 | Css1 |
| #id | ID选择器 | 匹配唯一标识id属性值相匹配的元素 | Css1 |

## ■关系选择器

|  |  |  |  |
| --- | --- | --- | --- |
| 选择器 | 名称 | 简介 | 版本 |
| selector1selector2 | 包含选择器 | 选择被selector1包含的selector2元素 | Css1 |
| selector1,selector2,select  or3 | 选择器分组 | 选择匹配，selector1,selector2,selector3的所有元  素 | css1 |
| selectorl>selector2 | 子元素选择器 | 选择所有属于selector1元素的子元素selector2 | Css2 |
| selector1+selector2 | 下一个相邻选择器 | 选择紧挨着selector1后面的selector2元素 | Css2 |

## ■属性选择器

|  |  |  |
| --- | --- | --- |
| 选择器 | 描述 | 版本 |
| E[attr] | 选取拥有此属性的E元素 | Css2 |
| E[attr="value"] | 选取属性的值为value的E元素 | Css2 |
| E[attr^="value"] | 选取属性的值为value开始的E元素 | Css3 |
| E[attr$="value"] | 选取属性的值以value结束的E元素 | Css3 |
| E[attr\*="value"] | 选取属性的值含有value的E元素 | Css3 |
| E[attr~="value"] | 选取多个属性值中包含value值的E元素 | Css2 |
| E[attr|=“value"] | 选择指定属性具有指定值开始的E元素。 |  |

## ■结构伪类选择器

|  |  |  |
| --- | --- | --- |
| 选择器 | 描述 | 版本 |
| :root | 选择匹配所在的文档的根元素 | Css3 |
| E:empty | 匹配没有任何子元素(包括文本节点)的元素E | css3 |
| E:first-child | 匹配父元素的第一个子元素E | Css2 |
| E:last-child | 匹配父元素的最后一个子元素E | Css3 |
| E:nth-child(n) | 匹配父元素的第n个子元素E | Css3 |
| E:nth-last-child(n) | 匹配父元素的倒数第n个子元素E | css3 |
| E:only-child | 匹配父元素仅有的一个子元素E | css3 |
| E:first-of-type | 匹配同类型中的第一个同级兄弟元素E | Css3 |
| E:last-of-type | 匹配同类型中的最后一个同级兄弟元素E | Css3 |
| E:only-of-type | 匹配同类型中的唯一一个同级兄弟元素E | Css3 |
| E:nth-of-type(n) | 匹配同类型中的第n个同级兄弟元素E | Css3 |
| E:nth-last-of-type(n) | 匹配同类型中倒数第n个同级兄弟元素E | css3 |

## ■状态伪类选择器

|  |  |  |
| --- | --- | --- |
| 选择器 | 描述 | 版本 |
| E:link | 超链接访问前的样式 |  |
| E:visited | 超链接被访问过后的样式 |  |
| E:hover | 设置元素在其鼠标悬停时的样式 |  |
| E:active | 设置元素在被用户激活时的样式 |  |
| E:focus | 设置元素在成为焦点时的样式 |  |
| E:checked | 匹配表单中，处于选中状态的元素E  (用于type=radio|checkbox) |  |
| E:enable | 匹配表单上处于可用状态的元素E |  |
| E:disabled | 匹配表单上处于禁用状态的元素E |  |

## ■伪类元素选择器

|  |  |  |
| --- | --- | --- |
| 选择器 | 描述 | 版本 |
| E:first-letterE::first-  letter | 设置对象内的第一-个字符的样式 |  |
| E:first-line/E::first-line | 设置对象内的第一行的样式 |  |
| E:before/E::before | 设置在对象前发生的内容，与  content属性-起使用 |  |
| E:afer/E::after | 设置在对象后发生的内容，与  content属性一起使用 |  |

## Content属性、before和after伪类

### 1. content属性

**■content- -般和:before,:after- 起使用，用来生成内容(img和input没有该属性)**

■content的内容一 般可以为以下四种 (IE6-7不支持) :

--none:不生成任何值。

--attr:插入标签属性值

**--url:使用指定的绝对或相对地址插入- 一个外部资源(图像，声频,视频或浏览 器支持的其他任何资源)**

**--string:插入字符串**

### 1.1 content:attr(属性)

**■content: attr(属性);将标记的属性插入到当前文本指定位置，后面的样式color和 background-color只对插入的文字生效，因为是p:before.**

**例:**

**<style>**

**div:after{**

content:attr(data-line);

color:f#ff;

**background-color:#090;**

}

**</style>**

<!--data-line=这是自定义属性，也可以使用标记原有属性-->

**<div data-line= "这是插入的文本">**

**测试哈哈**

**</div>**

### 1.2 content:url(链接资源地址)

**●content:url (链接资源地址);使用指定的绝对或相对地址插入-个外部资源(图像, 声频，视频或浏览器支持的其他任何资源)**

<style>

p:before(content:urlgive\_ advice .png);}

</style>

<body>

</body>

**<p>你好吗，好好学习，天天向上!你做到了吗? </p>**

### 1.3 content:字符串

●content:字符串:在指定位置插入字符串, 后面的样式color和background-color 只对插入的文字生效，因为是p:before.

例:

p:before{

content:"寄语哈哈:";

color:#fff;

background-color:#090;

<body>

<p>你好吗，好好学习，天天向上!你做到了吗? </p>

</body>

### 1.4 content:none

**引发的问题,因为p:before或p:after选择器让所有的p前面或后面插入文本?**

**●解决方法，如果不想让指定p前面或后面插入文本,那么可以使用content:none;**

**例:**

**<style>**

**p:before{**

**content:"寄语哈哈:";**

**olor:fff;**

**background-color:#090;**

**}**

**p.special:before{**

**content:none;**

**}/\*等价于content:normal\*/**

</style>

**<body>**

**<p>你好吗，好好学习，天天向上!你做到了吗? </p>**

**<p class="special">勤能补拙，奋斗不息</p>**

**<p>没有付出，没有回报</p>**

</body>

### 1.5 content属性插入项目符号

**before和after选择器和content属性配合，不仅可以在标签中插入文本,还可以 为多个项目插入项目符号**

#### 1.5.1添加连续的编号

**语法:**

**<元素> :before或after{**

content:counter(计数器);

}

**<元素>{**

**counter-increment:计数器;  
}**

#### 编号嵌套-复位计数器

**语法:**

**<元素> :before或after{**

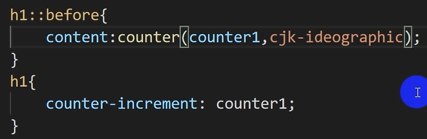
**content:counter(计数器编号种类):/\*编号种类是list-style-type属性值\*/}**

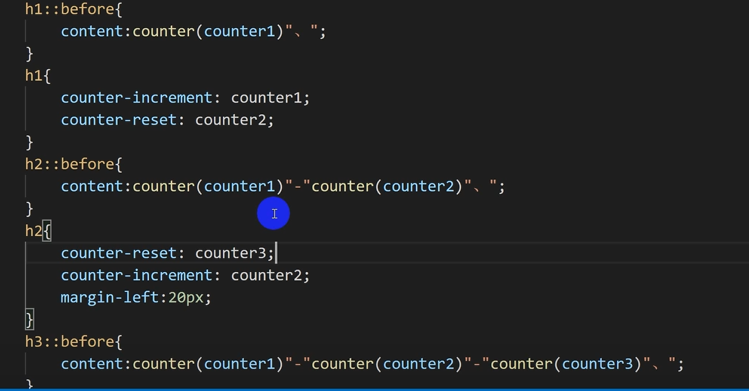
**<元素>{**

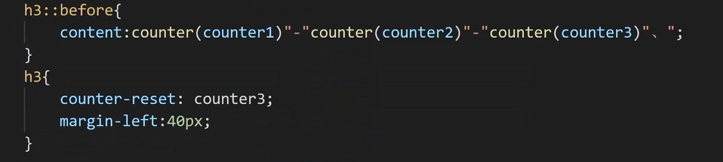
**counter-increment:计数器;**

**counter-reset:计数器1；/复位计数器\*/**

}







复位要给外层的元素加，不能给里层元素加，用他之前就复位的。

#### 1.5.2在字符串两边添加嵌套文字符号

■如果要在字符串两边添加诸如、单引号和双引号之类的文字符号，可以使用

content属性open-quote属性值、close-quote属性值和quotes属性进行设置

语法:

<元素>:before {

content:open-quote;

color:#f00;

font-family:黑体;

}

<元素> :after{

content:close-quote;

color:#f00;

font-family:黑体;

}<元素>{

quotes:"《""》";

}

## CSS3属性的兼容问题

■使用不同浏览器内核的私有属性解决低版本浏览器对css3支持不好的问题(具体哪些css3属性浏览 器不支持还需要查询css3手册)

■-ms-trident内核

■ -moz-gecko内核

■-webkit-webkit内核

■-o-presto内核

■用法:

■-ms-border-radius:50%;

■-moz-border-radius:50%;

■-webkit- border-radius:50;

■-o-border-radius:50%;

border-radius:50%:

# 第二章 新增样式（背景和边框）

## 背景色渐变：

线性渐变

语法规则:

. background: linear-gradient (方向，颜色1,颜色2， 颜色3， ..;



●说明:

设置线性渐变，方向有五个参数to top |to bottom | to left | to right还可以是对角的形式两个组合的方式to left top |to right top

.方向上除了使用指定的上下左右这种形式，还可以执行具体的角度0deg或45deg或90deg. 指定具体 的角度。

径向渐变

**语法规则:**

. background:adialgradient (颜色1,颜色2.颜色3， ...

## 边框—圆角：

语法规则：

border-radius:

说明：使用CSS3 border-radius属性,你可以给任何元素制作“圆角“

如果你在border-radius属性中只指定一个值,那么将生成4个圆角。

但是，如果你要在四个角上一一指定，可以使用以下规则：

四个值：第一个值为左上角，第二个值为右上角，第三个值为右下角，第四个值为左下角。

三个值：第一个值为左上角，第二个值为右上角和左下角，第三个之为右下角。

两个值：第一个值为左上角于右下角，第二个值为右下角于左下角

一个值：四个圆角值相同

## 边界图片：

给元素边框指定图像

语法规则：

border-image:source slice width outset repeat;

说明：

url()设置边界图片的切片，设置图像的边界向内的便宜长度

slice 定义边界图像的切片，设置图像的边界向内的偏移长度

width定义图像边框的宽度

outset定义边框图像的偏移位置

repeat定义边框图像的展现方式stretch(拉伸，默认值) repeat(平铺) round（自动缩放图像进行平铺）



## 背景图片：

--background-size:设置背景图片的大小，单位可以事px和%

--background-clip:content-box | padding-box | border-box

图片左上角在盒子左上角（超出内容的部分裁剪掉）

Background-origin是图片在内容部分显示。

--设置背景图片覆盖的范围，属性值：border-box:覆盖border区域padding区域.content区域padding-box：覆盖padding区域，content区域content-box覆盖content区域

Background-origin:content-box | padding-box | border-box

说明：在默认情况下使用background-position定位背景图片都是以元素边框以内的左上角位远点来定位，使用该属性可以改变背景图片远点位content部分左上角位原先padding部分左上角位原点，或者时border左上角为原点

## 阴影：

box-shadow用来给盒子模型添加阴影效果

说明:box-shadow属性有两个必须的值，还有三个可选值

h-shadow必须，阴影的水平设置，允许负值

v-shadow必须，阴影的垂直位置，允许负值

blur可选，模糊距离

spread 可选 阴影的尺寸

color 可选 阴影的颜色

inset 可选 将外部阴影改为内部阴影。



# 第三章css3文本效果和字体：

## 文本阴影效果

语法规则：

text-shadow:10px 5px 5px #ccc;

说明：

第一个参数表示阴影的水平位置

第二个参数表示阴影的垂直位置

第三个参数表示阴影的模糊距离

第四个参数表示阴影的颜色

补充：box-shadow，设置盒子（元素）阴影，用法同上

## 强制文本换行

Word-wrap:normal | break-word

说明:默认情况下单词或者是url地址都是在断点处换行，使用word-wrap:break-work;允许长单词或者是url地址换行到下一行，会从中剪断。

## 当文本溢出时发生的事

Text-overflow:clip | ellipsis；

说明：clip修剪文本，超出的部分隐藏

ellipsis用省略号表示被修剪得文本

注意：text-overflow属性要配合overflow:hidden一起使用才会生效

## 设置文本不进行换行

White-space:normal | pre | nowrap | pre-wrap | pre-line

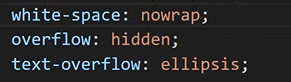
white-space：normal:默认处理方式。

Pre:用等宽字体显示预先格式化得文本，不合并文字间得空白距离，当文字超出边界时不换行。可查阅pre对象

nowrap:强制在同一行内显示所有文本，直到文本结束或者遭遇br对象。

Pre-wrap:用等宽字体显示预先格式化得文本，不合并文字间得空白距离，当文字碰到边界时发生换行。

pre-line:保持文本得换行，不保留文字间得空白距离，当文字碰到边界时发生换行。

(超出显示省略号。)

## Css3字体:

语法规则；

@font-face{

font-family:name;

src:url(字体路径)；

font-weight:normal | bold;

}

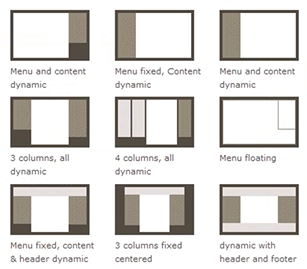
## Css3弹性盒子（flex box）（新）：

### 网页布局(layout):

网页布局(layout) 是css的一个重点应用，

Position属性+float属性，他对于那些特殊布局非常不方便，比如，垂直居中就不容易实现，

2009年，W3C提出了一种新的方案---Flex布局，可以简便、完整、响应式地实现各种页面布局。



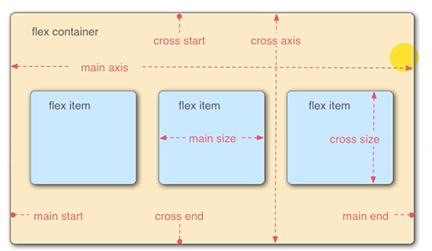
### Flex布局：

Flex布局是什么？

Flex是Flexible Box的缩写，意为“弹性盒子”，用来为盒状模型提供最大地灵活性。

任何一个容器都可以指定为Flex布局

采用Flex布局的元素，称为Flex容器(flex container),简称”容器”。它地所有子元素自动成为容器成员，称为Flex项目(flex item),简称”项目”。



容器默认存在两根轴:水平的主轴(main axis)和垂直的交叉轴(cross axis)。主轴的开始位置(与边框的交叉点)叫做mainstart,结束位置叫做main end;交叉轴的开始位置叫做cross start,

结束位置叫做cros send。

项目默认沿主轴排列。单个项目占据的主轴空间叫做main size, 占据的交叉轴空间叫做cross size.

任何一个容器都可以指定为Flex布局。

display：flex；

行内元素也可以使用flex布局。

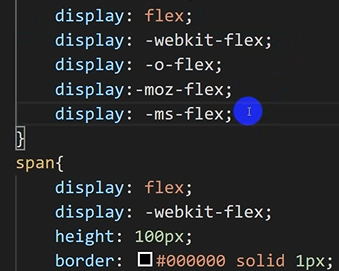
display:inline-flex;

Webkit内核地浏览器，必须加上-webkit-前缀

display:-webkit-flex;/\*safari\*/

display:flex;

父元素和子元素同时设置成弹性盒子，那么在显示的时候，里边的子元素的宽度由内容来决定，不管里边的子元素是否是块级元素

W3c

谷歌

欧朋

火狐

ie

容器的属性：

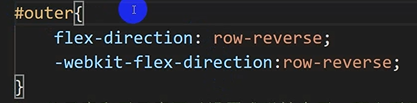
1. flex-direction 决定主轴的方向(即项目的排列方向)

row :默认值主轴为水平方向，起点在左端

row-reverse :主轴为水平方向，起点在右端

column:主轴为垂直方向，起点在上沿

column-reverse :主轴为垂直方向，起点在下沿





使用flex-direction的时候，row:盒子宽度是由内容决定的，column：盒子高度就是由内容来决定的。 \*\*这写属性都是给父元素加的。

2. flex-wrap定义如果一条轴线排不下， 如何换行（给父元素加）

nowrap (默认)不换行

**wrap换行，第一行在上方**

wrap-reverse换行，第一行在下方

3. flex-flow该属性是flex-direction属性和flex-wrap属性的简写形式，默认值是row nowrap

flex-flow: <flex-direction> || <flex-wrap>;

row-reverse排列顺序倒过来 nowrap代表不换行

4. justify-content属性定义项目在主轴上的对齐方式(水平对齐)

flex-start(默认值)左对齐

.flex-end右对齐

center居中

space-between两端对齐，项目之间的间隔都相等

space-around 每个项目两侧的间隔相等，所以项目之间的间隔比项目与边框的间隔大一倍

align-items属性定义项目在交叉轴(垂直轴)上如何对齐（垂直对齐）

flex-start 垂直居上

flex-end垂直居下

center 垂直居中

baseline 项目的第一行文字的基线对齐

stretch(默认值)如果项目未设置高度或设为auto，将占满整个容器的高度。



水平居中

垂直居中

（全都给父元素加）

页面中就一个div怎样居中？面试题：

给html加宽高，给body加居中属性

如果只想设置水平垂直都居中的话不用给里面的元素设置弹性盒子，想改变里面的东西例如排列顺序，换行等属性时把里面的设置为弹性盒子

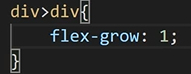
### 项目属性

■1. order属性定义项目的排列顺序，数值越小，排列越靠前，默认值为0

order : <integer>值为数字可以是 负数（调换里面子元素的排列数序）

■2. flex-grow属性定义项目的放大比例，默认为0，即如果存在剩余空间，也不放大

**flex-grow : <number>**



■如果所有项目的flex-grow属性都为1，则他们将等分剩余空间。如果一个项目的flex-grow属性为2, 其他项目为1，则前者占据的剩余空间将比其他项目多一倍

■3. flex-shrink属性定义了项目的缩小比例，默认为1，即如果空间不足，该项目将缩小（不能用了）

**flex-shrink:<number>**

如果所有项目的flex-shrink属性都为1，当空间不足时，都讲等比例缩小，如果一-个项目的flex- shrink属性为0，其他项目都为1，则空间不足时，前者不缩小，负值无效

■4. flex-basis属性定义了在分配多余空间之前，项目占据的主轴空间。浏览器根据这个属性，计算主 轴是否有多余空间。它的默认值为auto，即项目的本来大小

flex-basis : <length> | auto;

它可以设为跟width或height属性--样的值(比如350px)，则项目将占据固定空间

■5. flex属性是flex- grow,flex-shrink和flex-basis的简写， 默认为0 1 auto后两个属性可选。(复合属性)

auto相当于(1 1 auto)

none相当于(0 0 auto)

建议优先使用这个属性，而不是单独写三个分离的属性，因为浏览器会推算相关值

■6. align-self属性允许单个项目与其他项目不一样的对齐方式，可覆盖align-items属性， 默认值为auto, 表示继承父元素的align-items属性，如果没有父元素，则等同于stretch

align-self : auto | flex-start | flex-end | center | baseline | stretch

## 盒布局和界面设置（老盒子）

### 盒子布局

1、开启盒子布局

**display:-webkit-box;**

**display:-moz-box;**

**display:box;**

**2、 元素的布局方向--box-orient属性**

**-webkit- -box-orient: horizontal | vertical**

**horizontal:设置弹性盒模型对象的子元素为水平排列**

**vertical:设置弹性盒模型对象的子元素为纵向排列**

3、元素的布局顺序-webkit-box-direction属性

box-direction: normal | reverse

normal:设置弹性盒模型对象的子元素按正常顺序排列

reverse:反转弹性盒模型对象的子元素的排列顺序

4、调整元素的位置-webkit-box-ordinal-group属性

box-ordinal-group: <integer>（他是给子元素加的）

<integer>:用整数值来定义弹性盒模型对象的子元素显示顺序

5、弹性空间分配-webkit-box-flex属性

**box-flex: <number>**

<number>:使用浮点数指定对象所分配其父元素剩余空间的比例

**6、元素对其方式box-pack和box-align属性**

**box-pack: start| center | end | justify（水平方向对齐方式）**

start:设 置弹性盒模型对象的子元素从开始位置对齐(大部分情况等同于左对齐)

center:设 置弹性盒模型对象的子元素居中对齐

end:设 置弹性盒模型对象的子元素从结束位置对齐(大部分情况等同于右对齐) justify:设 置或弹性盒模型对象的子元素两端对齐

**box-align: start | end | center| baseline| stretch（垂直方向对齐方式）**

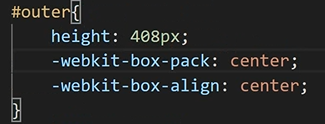
start:设置弹性盒模型对象的子元素从开始位置对齐

center:设置弹性盒模型对象的子元素居中对齐

end:设置弹 性盒模型对象的子元素从结束位置对齐

baseline:设 置弹性盒模型对象的子元素基线对齐

stretch:设 置弹性盒模型对象的子元素自适应父元素尺寸



### 增强的盒子模型

1、盒子阴影-- box-shadow属性

**box-shadow: inset? && [ <length>{2,4} && <color>?];**

none:无阴影

<length>①:第1个 长度值用来设置对象的阴影水平偏移值。可以为负值

<length>②:第2个 长度值用来设置对象的阴影垂直偏移值。可以为负值

<length>③:如果提供 了第3个长度值则用来设置对象的阴影模糊值。不允许负值

<length>④:如果提供 了第4个长度值则用来设置对象的阴影外延值。不允许负值

<color>:设置对象的阴影的颜色。

inset:设置对象的阴影类型为内阴影。该值为空时，则对象的阴影类型为外阴影

2、盒子尺寸的计算方法--box-sizing属性

**box-sizing: content-box | border-box**

content-box: padding和border不被包含 在定义的width和Iheight之内。

border-box: padding 和border被包含在定义的width和heigh之内。

3、溢出内容处理- overflow-x和overflow-y属性

**overflow-x: visible | hidden | scroll | auto**

**overflow-y: visible | hidden | scroll | auto**

visible: 不剪切内容。

hidden: 将超出对象宽度的内容进行裁剪。将不出现滚动条。

scroll: 将超出对象宽度的内容进行裁剪，并以滚动条的方式显示超出的内容。

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

## 多列布局

### display取值

**display: none | inline | block | list-item | inline-block | table | inline-table | table-caption | table-cell | table-row | table-row-group | table-column | table- column-group | table-footer-group | table-header-group | compact | run-in| ruby | ruby-base | ruby-text | ruby-base-group | ruby-text-group | box | inline-box**

**用法:**

none:隐藏对象。与visibility属 性的hidden值不同，其不为被隐藏的对象保 留其物理空间.

inline:指定对象为内联元素。

block:指定对象为块元素。

list-item:指定 对象为列表项目。

inline-block:指定对象为内联块元素。(CSS2)

table:指定对象作为块元素级的表格。类同于html标签<table> (CSS2)

inline-table:指定对象作为内联元素级的表格。类同于html标签<table>(CSS2 )

table-caption:指定对象作为表格标题。类同于html标签<caption> (CSS2)

table-cell:指定对象作为表格单元格。类同于html标签<td> (CSS2)

table-row:指定对象作为表格行。类同于html标签<tr> (CSS2)

table-row-group:指定 对象作为表格行组。类同于html标签<tbody>(CSS2)

table-column:指定 对象作为表格列。类同于html标签<col> (CSS2)

table- column-group:指定 对象作为表格列组显示。类同于htm|标签

<colgroup> (CSS2 )

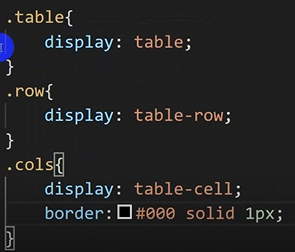


table-header-group:指定对象作为表格标题组。类同于html标签<thead>(CSS2)

table-footer-group:指定对象作为表格脚注组。类同于html标签<tfoot>(CSS2)

compact:分配对 象为块对象或基于内容之.上的内联对象(CSS3)

run-in:分配对 象为块对象或基于内容之上的内联对象(CSS3)

ruby:将对象作为表格脚注组显示(CSS3)

ruby-base:将对 象作为表格脚注组显示(CSS3)

ruby-text:将对象作为表格脚注组显示(CSS3)

ruby-base-group:将对象作为表格脚注组显示(CSS3)

ruby-text-group:将对 象作为表格脚注组显示(CSS3)

box:将对象作为弹性盒模型显示(CSS3)

inline-box:将对象作为内联块级弹性盒模型显示(CSS3）

### 多列属性：

1、多列属性columns

columns: [ column-width ] II [ column-count ] 

[ column-width]:设置或检索对象每列的宽度

[ column-count]:设置或检索对象的列数

2、列宽属性column-width

column-width: <length>| auto

<length>:用长度值来定义列宽。不允许负值 auto:根据column-count自定分配宽度

**3、列数属性column-count**

column-count: <integer>| auto

<integer>:用整数值来定义列数。不允许负值

auto:根据column-width自定分配宽度

**4、列间距属性column-gap**

column-gap: <length>| normal

<length>:用长度值来定义列与列之间的间隙。不允许负值

normal:与font-size大小相同。假设该对象的font-size为16px，则normal值为16px。

**5、定义列分割线--column-rule属性**

column-rule: [ column-rule-width ] || [ column-rule-style ] || [ column-rule- color ]

[ column-rule-width]:设置或检索对象的列与列之间的边框厚度。

[ column-rule-style ]:设置或检索对象的列与列之间的边框样式。

[ column-rule-color]:设置或检索对象的列与列之间的边框颜色。

**6、定义横跨所有列--column-span属性**

**column-span: none | all**

none:不跨列

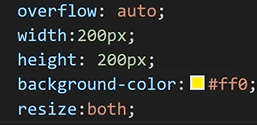
all:横跨所有列

### 鼠标类型：

cursor: auto | default | none | context-menu | help | pointer | progress | wait| cell | crosshair| text| vertical-text| alias | copy | move | no-drop| not- allowed | e-resize | n-resize| ne-resize| nw-resize| s-resize| se-resize| sw-resize| w-resize| ew-resize| ns-resize| nesw-resize| nwse-resize| col- resize | row-resize| all-scroll]

## 用户界面设计

### resize属性

resize: none | both | horizontal | vertical

none:不允许用户调整元素大小。

both:用户 可以调节元素的宽度和高度。

horizontal:用户可以调节元素的宽度

vertical:用户 可以调节元素的高度。

### 2、outline属性

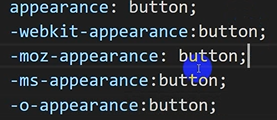
**outline: [ outline-width ] II [ outline-style ] II [ outline-color ]**

[ outline-width]:指定轮廓边框的宽度。

[ outline-style]:指定轮廓边框的样式。

[ outtine-color ];指定轮廓边框的颜色。(轮廓边框是在border外面的，不包含margin)

### 3、伪装元素--appearance属性



**appearance: normal | icon | window | button | menu | field;**

**normal:正常地修饰元素**

icon:把元素修饰得像一个图标

window:把元素修饰得像- - 个视图

button:把元素修饰得像一个按钮

menu:把元素修饰得像菜单

field:把元素修饰的向一个输入框

### 4、content属性

content: none | normal | <string> | counter(<counter>) | attr(<attribute>)| url (<ur|>) | inherit ;

normal:默认值。表现与none值相同

none:不生成任何值。

<string>:插入字符串

<attr> :插入标签属性值

<url>:使用指定的绝对或相对地址插入一一个外部资源(图像，声频，视频或浏 览器支持的其他任何资源)。

## box-reflect属性

**说明:**

盒子倒影属性:可以对盒子模型进行倒影设置。

格式:

**none|**

**below:下方**

**above:.上方**

**left:左边**

**right:右边**

# 第四章2D和3D转换

## 2D变形

### Transform属性

■css3新增的transform属性，可用于元素的变形，实现元素的旋转，缩放，移动，倾斜等变形 效果

■基于webkit内核的替代私有属性: -webkit-transform

■基于gecko内核的替代私有属性: -moz-transform

■基于presto内核的替代私有属性: -o-transform

■基于ie内核的替代私有属性: -ms-transform

#### ■基本语法:

**■transform: none| transform-functions;**

■说明:

■none:默认值，不设置变形

■transform-functions:设置一个或多 个变形函数。变形函数包括旋转rotate()、缩放 scale()、移动translate()、 倾斜skew()、 矩阵变形matrix()，设置多个变形函数时，用空 格间隔。

#### ■rotate()旋转

语法格式: .

**transform: rotate(30deg);**

**说明: rotate()方法， 使用参数表示旋转的角度，单位是deg,可以是负数表示逆时针**

#### ■scale()方法缩放

语法规则: .

**transform:scale(x,y);**

说明: x,y表示的是宽度和高度的倍数，以原图的中心点进行缩放



#### ■translate()移动

语法说明:

transform: translate(dx,dy);

说明: translate()用于定义元素在二维空间的偏移

dx，表示元素在水平方向上的偏移距离

dy,表示元素在垂直方向上的偏移距离

dx,dy使用带有单位的长度，可以是正数也可以是负值

Trsansform:translate(10px, 10px);

#### ■skew()倾斜

skew()函数用于定义元素在二维空间的倾斜变形

语法规则: .

skew(angleX,angleY);

说明:参数表示在x轴和y轴上的倾斜角度，单位是deg

#### ■Transform-origin:x y;(设置圆心点)

说明：x和y的默认值都是50%,也就是元素的中心位置，设置新的圆心，可以使用百分数的形式，也可以使用x:left | center | right , y:top | middle | bottom

## 3D变形

Perspective属性设置镜头到元素平面的距离

取值：像素(px)

Transform-style：规定如果在3D空间中呈现被嵌套的元素

取值：

-flat:子元素将不保留3D位置。

-preserve-3d:子元素将保留其3D位置。

# 第五章 css3动画

## 过渡效果transition

■说明: css3过渡是元素从-种样式逐渐改变为另-种的效果

■transition 简写属性，用于在一个属性中设置四个过渡属性

■transition-property规定应用过渡的CSS属性的名称

■transition-duration定义过渡效果花费的时间。默认是0。单位S

■transition-timing-function规定过渡效果的时间曲线。默认是"ease"。

■transition-delay 规定过渡效果何时开始(延迟时间)。默认是0。单位s

■transition-timing-function属性指定切换效果的速度。  
■linear 规定以相同速度开始至结束的过渡效果。  
■ease 默认值，规定慢速开始，然后变快，然后慢速结束的过渡效果。  
■ease-in 规定以慢速开始的过渡效果。  
■ease-out 规定以慢速结束的过渡效果。  
■ease-in-out 规定以慢速开始和结束的过渡效果。

■例:

■div { width: 100px;height: 100px;background: red;

■ transition: width 2s, height 2s, transform 2s;

■ }

■ div:hover {width: 200px;height: 200px;

■ transform: rotate(180deg);

■ }

## 动画

关键帧动画--@keyframes规则

语法规则： 语法规则：

@keyframes name{ @keyframes name{

from{css样式} 0% {css样式}

to{css样式} 25% {css样式}

} 50% {css样式}

75% {css样式}

100% {css样式}

}

■animation-name指定动画的名称

■animation duration指定动画多少秒或亳秒完成

■animation-timing-function设置动画将如何完成一个周期

linear动画从头到尾的速度是相同的

ease默认，动画以低速开始，然后加快，在结束前变慢

ease-in动画以低速开始

ease-out动画以低速结束

ease-in-out动画以低速开始和结束

■animation-iteration-count规定动画被播放的次数。默认是1, infinite循环播放

■animation-delay定义动画什么时候开始(动画开始前等待的时间)单位是s或者ms

■animation-direction规定动画是否在下一周期逆向地播放。

normal默认值

reverse动画反向播放

alternate奇数次正向播放，偶数次反向播放

■animation-fill-mode指定动画在播放之前或之后，其动画效果是否可见，默认值none，不改变默认行为; forwards,当前动画完成后，保持最后一个属性值(在最后一个关键帧中定义)

■animation-play-state 指定动画是否正在运行或暂停，默认值running，暂停是paused

/\* 创建动画：

@keyframs 动画名{

from{} 开始状态

to{} 最终状态

}

调用动画:

animation:动画名执行时间forwards ( 是否定位到最终得状态) ;

transform:ratote scale skew translate;

perspective: ; //设置镜头到平 面得距离

transform-style:是否让子元素保留3D效果

transition:过渡效果

过度得属性过渡时间过渡曲线延迟时间

\*/