# HTML5+CSS3

尚硅谷前端研究院

版本：V 1.0

# 第1章 选择器

## 基本选择器

### 1.1.1子元素选择器

1)定义：只能选择某元素的子元素

2)语法：父元素>子元素

### 1.1.2相邻兄弟选择器

1)定义：可以选择紧接着在另外一个元素后的元素，而且他们具有一个相同的父元素

2)语法：元素 + 相邻兄弟元素

### 1.1.3通用兄弟选择器

1)定义：选择某元素后边的所有兄弟元素，而且他们具有一个共同的父亲

2)语法：元素 ~ 后面所有兄弟元素

### 1.1.4群组选择器

1)定义：将具有相同样式的元素分组在一起，每个选择器之间用逗号 ， 隔开

2) 语法：元素1，元素2，元素3，.......

## 属性选择器

### 1.2.1 element[attribute]

1)定义： 为带有attribute属性的元素设置样式

2)语法：element[attribute]

### 1.2.2 element[attribute='value']

1) 定义：为attribute='value'属性的元素设置样式

2)语法：element[attribute='value']

### 1.2.3 element[attribute~='value']

1) 定义：选择attribute属性包含单词value的元素,并设置样式

2) 语法：element[attribute~='value']

### 1.2.4 element[attribute^='value']

1) 定义：设置attribute属性值，以value开头的所有元素样式

2)语法：element[attribute^='value']

### 1.2.5 element[attribute$='value']

1) 定义：设置attribute属性值，以value结尾的所有元素样式

2) 语法：element[attribute$='value']

### 1.2.6 element[attribute\*='value']

1)定义：设置attribute属性值，包含value的所有元素，并为其设 置样式

2)语法：element[attribute\*='value']

## 动态伪类

### 1.3.1 锚点伪类

1):link

2):visited

### 1.3.2 用户行为伪类

1) :hover

2) :active

3) :focus

### 1.3.3 目标伪类

:target 当我们点击锚链接时，对应ID的元素会显示在视口

### 1.3.4 checked状态伪类

1) checkbox:只能设置宽度和高度，不能设置背景颜色和边框

2) 清除input的默认样式 appearance: none;

### 1.3.5 CSS3结构类

1) first-child：选择属于其父元素的首个子元素的每个element元素，并为其设置样式（element:first-child）

2) last-child：选择属于其父元素的最后一个子元素的每个element元素，并为其设置样式（element:last-child）

3) nth-child(n)：选择某元素下第number个element元素（n是一个简单的表达式，不能用其他字母代替，n从0开始计算）

1.nth-child(odd) : 可用于匹配下标是奇数的元素的关键字

2.nth-child(even):可用于匹配下标是偶数的元素的关键字

4) nth-last-child()：匹配属于其元素的第n个元素的每个元素，从最后一个子元素开始计数(element:nth-last-child(n))

5) nth-of-type()：匹配属于父元素的特定类型的第n个子元素（element:nth-of-type()）

6) nth-last-of-type()：选匹配属于父元素的特定类型的第n个子元素，从最后一个开始计数(element:nth-last-of-type()）

7) first-of-type()：匹配属于其父元素的特定类型的首个子元素的每个元素（element：first-of-type()）

8) last-of-type()：匹配属于其父元素的特定类型的最后一个子元素的每个元素（element：last-of-type()）

9) :only-child：匹配属于其父元素的唯一子元素的每个元素（element:only-child）

10) :only-of-type：匹配属于其父元素特定类型的唯一子元素的每个元素（element :only-of-type）

11) :empty：匹配没有子元素（包括文本节点）的每个元素（element :empty ---- div:empty）

### 1.3.6 案例

1) 目标伪类案例（图片切换）



2) checked状态类案例（自定义选框）

未标题-1

3) CSS3结构类案例（尺子）



## 否定选择器

### 1.4.1 定义

匹配非 元素或者选择器 的每个元素

### 1.4.2 语法

父元素：not(子元素或者选择器)

### 1.4.2 案例

ul :not(span)

## 伪元素

### 1.5.1 element::first-line

对元素的第一行文本进行设置，只能用于块级元素

### 1.5.2 element::first-letter

用于向文本的首字母设置特殊样式，只能用于块级元素

### 1.5.3 element::before

在元素的内容前面插入新内容，常与content配合使用

### 1.5.4 element::after

在元素的内容后面插入新内容，常与content配合使用

### 1.5.5 element::selection

用于设置浏览器中选中文本后的背景色与前景色

### 1.5.6 伪元素与元素的区别

无法通过JS获取其DOM，无法通过浏览器直接查看

伪元素默认是 inline

### 1.5.7 使用伪元素注意事项

1)使用伪元素before,after必须设置content

2)使用伪元素before,after显示背景图，一定要使用display设置为块元素

3)使用伪元素before,after设置为display:inline\_block,需要再次设置vertcal-align:middle

### 1.5.8 案例

文章列表



## CSS权重（优先级）

### 1.6.1 定义

当很多规则被应用到某一个元素上时，权重是一个决定哪种规则生效，或者是优先级的过程

### 1.6.2 权重的等级与权值

行内样式（1000）> ID选择器（100）>类，属性选择器和伪类选择器（10）> 元素选择器和伪元素选择器（1）>通配符选择器（0）

### 1.6.3 CSS权重规则

1) 当多个选择器发生冲突时，会选择权重高的选择器来显示，权重越高越优先显示

2) 比较时需要将多个选择器的权重进行相加在进行比较，如果权重一样，后面的会覆盖前面的样式

3) 权重相加不可能超过他的最大数量级，例如无论多少个元素组成的选择器，都没有一个class选择器权重高

4) 可以在样式后边添加一个！important ，这样该样式将会拥有最大的权重，其他样式都不能将其覆盖（注意：尽量不要使用！important）

# 第2章 边框圆角

## 2.1 Border-radius

### 2.1.1定义

可以为元素添加圆角边框（块元素，行内块元素，行内元素）

### 2.1.2 语法使用

1) 四个值：左上角 右上角 右下角 左下角

2) 三个值：左上角 右上角和左下角 右下角

3)两个值：左上角和右下角 右上角和左下角

4)一个值：4个角都生效

## 2.2圆形与椭圆形

1) 一旦使用百分比，参照的是元素本身的高度与宽度

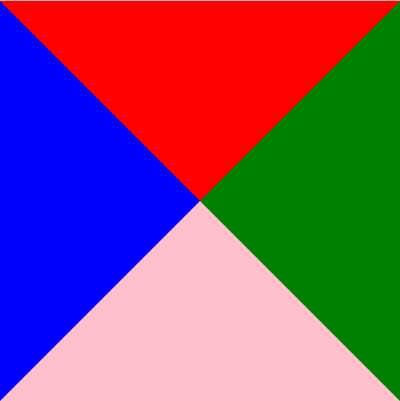
当拿50%时， 宽等于高 ---- 圆形

宽不等于高 --- 椭圆形

2) 椭圆形：border-radius: x轴半径 / y轴半径

## 2.3案例

1)边框实现三角形



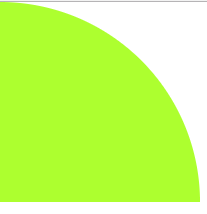
2)搜狐评论框



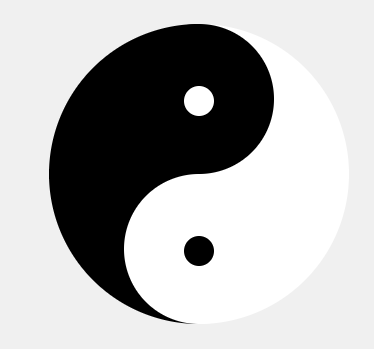
3)半圆



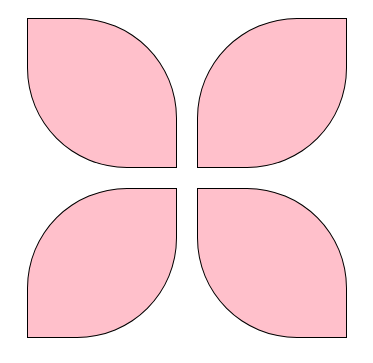
4)扇形



5)太极图



6)风车



# 第3章 盒阴影

## 3.1 Box-shadow

### 3.1.1定义

可以控制一个或多个下拉阴影的框

### 3.1.2 语法使用

box-shadow: 水平方向的偏移量 垂直方向的偏移量 模糊程度 扩展程度 颜色 是否具有内阴影

如果设置多个阴影，各项参数使用都好分开

## 3.2内阴影

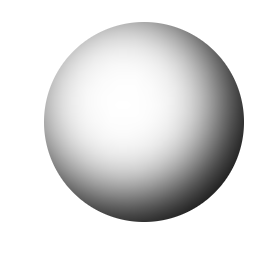
inset(默认没有，也就是默认是外阴影)

加上inset,由元素本身的位置先里挤效果

注意问题：扩展程度可为负值，但是模糊不可以

## 3.3案例

1)立体球



2)重复盒子阴影



3)模糊盒子阴影



# 第4章 背景

## 4.1背景裁剪

## 4.1.1定义

background-cilp：指定背景的绘制区域

### 4.1.2 语法使用

background-cilp：border-box / padding-box / content-box

border-box:默认值

## 4.2背景原始起始位置

### 4.2.1定义

background-origin：设置背景图像的原始起始位置

### 4.2.2 语法使用

background-origin：border-box / padding-box / content-box

padding-box:默认值

### 4.2.3 注意问题

background-position:定义背景图片的位置，水平与垂直方向上面的偏移量（参考点与这三个有关系）

## 4.3背景图像的大小

### 4.3.1定义

background-size：指定背景图像的大小

### 4.3.2 语法使用

background-size：number / % / cover / contain

### 4.3.3 属性说明

background-size： 宽度 高度（如果只写一个数值，第二个数值默认auto）

百分比： 0% - 100% 之间的任何值，此时的百分比参照于元素div的大小

cover：将背景图片以容器最远边进行缩放，如果高度达到一定比例100%，宽度多出的会溢出

contain：将背景图片以容器最近边进行缩放，如果高度达到一定比例100%，宽度部分就会出现空白

## 4.4多重背景

### 4.4.1定义

background-image：CSS3允许您为元素使用多个背景图片

### 4.4.2 语法使用

background-image: url('1.jpg),url('2.jpg') ... 使用逗号把图片分开

### 4.4.3 注意问题

元素引入多个背景图片，前面图片会覆盖后面的图片

## 4.5案例

背景模糊



# 第5章 渐变

## 5.1线性渐变

### 5.1.1定义

background-image：是沿着一根轴线改变颜色，从起点到终点进行顺序渐变

### 5.1.2 语法使用

background-image:linear-gradient(方向，开始颜色，结束颜色)

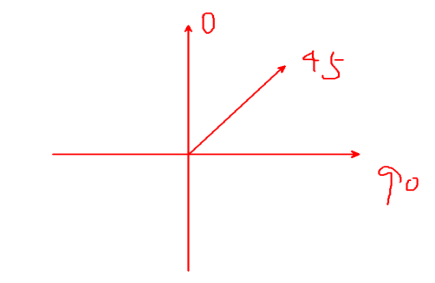
### 5.1.3 方向分类

1) 从上到下（默认）：background: linear-gradient(red,blue);

2) 从左到右：background: linear-gradient(to right,red,blue);

3) 对角：background: linear-gradient(to right bottom,red,blue);

4) 角度：



### 5.1.4 颜色结点

默认每个颜色均匀分布

1) background: linear-gradient(red 10%,blue 20%,green 30%,yellow 40%);

从0%到10%，为红色，从10%到20%为红色到蓝色的渐变，从20%到30%为蓝色到绿色的渐变，从30%到40%，为绿色到黄色的渐变

2) background: linear-gradient(red 10%,blue);

从0%到10%，为红色，从10%到100%为红色到蓝色的渐变，最后如果不写具体数值，默认到100%

3) background: linear-gradient(red,blue 30%);

从0%到30%，为红色到蓝色的渐变，如果第一个不写，默认数值是 0%

4) background: linear-gradient(rgba(255,0,0,0),rgba(255,0,0,1));

由透明色变为不透明色

### 5.1.5 重复渐变

background: repeating-linear-gradient(90deg,red 0%,blue 20%);

或者

background: repeating-linear-gradient(90deg,red 0%,blue 10%,red 20%);

### 5.1.6 注意问题

渐变本质绘制的是一张图片（背景图片），所以使用background或者使用background-image

百分比：把元素渐变方向的整体长度看成100%

## 5.2径向渐变

### 5.2.1定义

background-image：从起点到终点，颜色从内向外进行圆形渐变

### 5.2.2 语法使用

background:radial-gradient(形状尺寸，开始颜色，结束颜色)

### 5.2.3 形状分类

1) 圆形circle

2) 椭圆形ellipse

3) 注意：当元素的高和宽一样时，参数无论设置谁，都是圆形

### 5.2.4 颜色结点

background: radial-gradient(circle,red 30% ,blue 70%);

注意：此时的百分比,指的是圆心到元素最远端的距离（角度）

### 5.1.5尺寸大小

1) closest-side最近边

background: radial-gradient(closest-side circle,red , blue);

2) farthest-side 最远边

background: radial-gradient(farthest-side circle,red , blue);

3) closest-corner最近角

background: radial-gradient(closest-corner circle,red , blue);

4) farthest-corner最远角

background: radial-gradient(farthest-corner circle,red , blue);

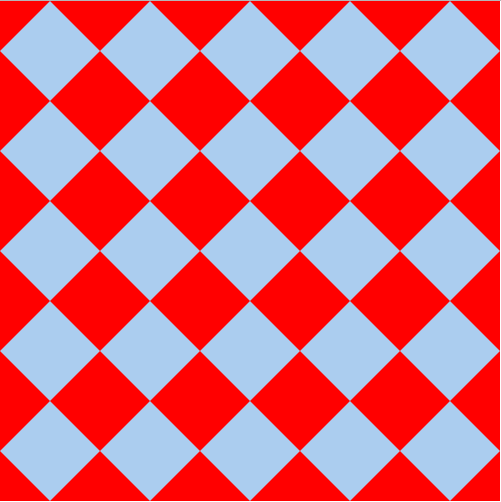
### 5.1.6 重复渐变

background: repeating-radial-gradient(red 0%,blue 20%);

background: repeating-radial-gradient(red 0%,blue 10%,red 20%);

## 5.3案例

1)马赛克地砖



2)发廊灯



3)光斑动画



# 第6章 过渡

## 6.1定义

允许css的属性值在一定时间区间内平滑的过渡，在鼠标点击，鼠标滑过或对元素任何改变中触发，并圆滑地以动画改变css的属性值

## 6.2属性

### 6.1.1 transition-property

1) 定义：设置对象中的参与过渡的属性

2) 语法：transition-property：none | all | property

3) 参数说明：

none： 没有属性改变

all : 默认值，所有属性都改变

property： 元素的属性名 color等

### 6.1.2 transition-duration

1) 定义：设置对象过渡的持续时间

2) 语法：transition-duration：time

3) 参数说明：

规定完成过渡效果需要花费的时间，以秒或者毫秒计，默认值0

### 6.1.3 transition-timing-function

1) 定义：设置对象中过渡的动画类型

2) 语法：transition-timing-function：动画类型（只能使用一种）

3) 参数说明：

linear:线性过渡（匀速）cubic-bezier(0,0,1,1)

ease:平滑过渡（慢--快--慢），默认值 cubic-bezier(0.25,0.1,0.25,1)

ease-in:慢--快 cubic-bezier(0.42,0,1,1)

ease-out:快--慢 cubic-bezier(0,0,0.58,1)

ease-in-out:慢--快--慢 cubic-bezier(0.42,0,0.58,1)

贝塞尔曲线

### 6.1.4 transition-delay

1) 定义：设置对象延迟的过渡时间

2) 语法：transition-delay：time

3) 参数说明：

指定秒或者毫秒数之前要等待切换效果的开始，默认是0

### 6.1.5 transition

1) 定义：设置对象变换时的过渡

2) 语法：transition：property duration timing-function delay；

3) 参数说明：

时间顺序不能乱，其他参数位置不限

如果想给多个属性添加不同的过度，参数之间使用逗号分开

## 6.3案例

幽灵按钮



# 第7章 变换

## 7.1定义

让一个元素在一个坐标系统中变形，这个属性包含一系列的变形函数，可以移动，旋转，缩放元素

## 7.2 2D变换

### 7.2.1 旋转

1) 定义：通过指定一个角度参数，对元素指定一个2D的旋转

2) 语法：transform：rotate(angle) 单位deg

3) 注意：angle指旋转角度，正数表示顺时针旋转，负数表示逆时针旋转

### 7.2.2 平移

1) 定义：根据X轴和Y轴的位置给定参数，使当前元素位置移动

2) 语法：

transform：translateX() --- 仅水平方向移动

transform：translateY() --- 仅垂直方向移动

transform：translate( X, Y) --- 水平方向和垂直方向同时移动

单位px

注意：

如果只写一个参数，第二个默认是0，也就是只设置了水平方向上的位移

### 7.2.3 缩放

1) 定义：设置元素的缩放程度

2) 语法：

transform：scaleX() --- 仅水平方向缩放

transform：scaleY() --- 仅垂直方向缩放

transform：scale(x,y) --- 使元素垂直和水平方向同时缩放

没有单位

3) 语法：

如果只写一个参数，元素垂直和水平方向同时缩放

### 7.2.4 扭曲/倾斜

1) 定义：设置元素的倾斜状态

2) 语法：

transform：skewX() --- 仅使元素在水平方向上扭曲变形

transform：skewY() --- 仅使元素在垂直方向上扭曲变形

transform：skew(x,y) --- 使元素在水平方向和垂直方向上扭曲变形

单位deg

3) 语法：

0deg与180deg 效果一样

### 7.2.5 变换基点

1) 定义：元素变换的基准点

2) 语法：

transform-origin：水平方向 垂直方向

3) 默认值：

rotate 几何中心点

skew 几何中心点

scale 几何中心点

translate 本身位置

### 7.2.6 案例

1)瓶体旋转



2)菜单按钮



3)钟表



## 7.3 3D变换

### 7.3.1 开启3D空间

transform-style: preserve-3d（一般对父元素设置）

### 7.3.2 3d变换设置

rotateX()：指对象在X轴上的旋转角度（变换基点： 50% 50% 0）

rotateY()：指对象在Y轴上的旋转角度（变换基点： 50% 50% 0）

rotateZ()：指对象在Z轴上的旋转角度（变换基点： 50% 50% 0）

translateZ()：指对象在Z轴上面的平移（变换基点： 50% 50% 0）

scaleZ():指对象在Z轴上面的缩放（变换基点： 50% 50% 0）

### 7.3.3 景深设置

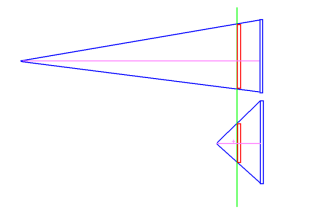
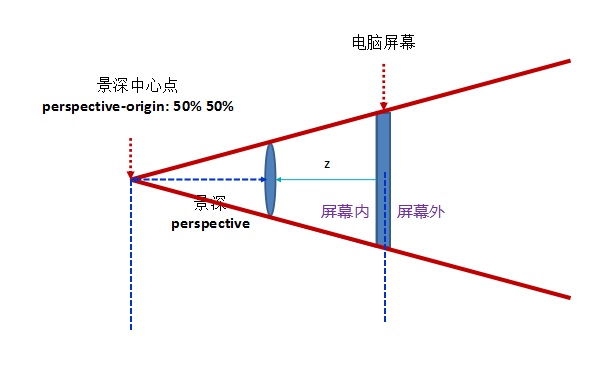
1)定义：实现元素在3D空间中的近大远小的效果

2)设置：

父元素设置景深：perspective: 300px;

子元素设置景深：transform:perspective(300px) translateZ(-200px);

3)景深分析图



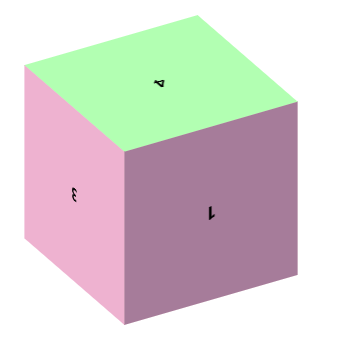
### 7.3.4 变换基点

1)默认值： 50% 50% 0

2)语法使用：可以使用关键字（top,bottom,left,right）,百分比，具体像素值等

3)注意：立体3d盒子 Z：只能使用具体的长度，不能使用百分比和关键字

4)案例：旋转的立体盒子



### 7.3.5 景深中心点

1)定义：改变观察者视角

2)语法使用：

perspective-origin: top right;

perspective-origin: top;

### 7.3.6 元素背面是否可见

backface-visibility:visible ;（默认值：可见）

backface-visibility: hidden; 不可见

### 7.3.7 案例

1)旋转盒子相册



2)3d相册



3)天猫商品墙



# 第8章 动画

## 8.1定义

使元素从一种样式逐渐变化到另外一种样式的效果

## 8.2原理

视觉暂留原理：

人类具有”视觉暂留“的特征，人的眼睛在看到一幅画或一个物体后，在0.34s 内不会消失

动画原理：

通过把人物的表情，动物变化等动作，分解成许多动作瞬间的画幅，利用视觉原理，在一幅画还没消失前播放下一副画，就会给人造成一种流畅的视觉变化效果

## 8.3关键帧@keyframes

1)定义：keyframes关键帧，用来决定动画变化的关键位置

（注意：keyframes 控制关键位置，并不是所有的位置）

2)语法：

@keyframes animationname{

keyframes-selector{

cssStyles;

}

}

3)参数说明：

animationname：必写项，定义动画的名称

keyframes-selector：必写项，动画持续时间的百分比

0% - 100%之间， 或者使用form和to关键字也可以设置，form代表0%，to代表100%

## 8.4 animation属性

### 8.4.1 animation-name

1)定义：设置对象所应用的动画名称

2)语法：animation-name：keyframename | none

3)参数说明：

keyframename：指定要绑定到选择器的关键帧的名称

### 8.4.2 animation-duration

1)定义：设置对象动画的持续时间

2)语法：animation-duration：time

3)参数说明：

指定对象播放完成需要花费的时间，默认值是0

### 8.4.3 animation-timing-function

1)定义：设置对象动画的过渡类型

2)参数说明：

linear:线性过渡（匀速）

ease:平滑过渡（0--慢--快--慢），默认值

ease-in:慢--快

ease-out:快--慢

ease-in-out:慢--快--慢

贝塞尔曲线

### 8.4.4 animation-delay

1)定义：设置对象动画的延迟时间

2)语法：animation-delay：time

3)参数说明：

可选值，定义动画开始前等待的时间，以秒或者毫秒数计数，默认值是0

### 8.4.5 animation-iteration-count

1)定义：设置对象动画的循环次数

2)语法：animation-iteration-count ： infinite | number

3)参数说明：

number为数字，其默认值是1

infinite：无限循环次数

### 8.4.6 animation-direction

1)定义：设置对象动画是否反向运动

2)语法：

animation-direction：normal , reverse , alternate , alternate-reverse

3)参数说明：

Normal:正常方向

reverse :反向运动

alternate:先正常运动在反向运动，并持续交替运行， 需要配合循环属性使用

alternate-reverse:先反向运动在正常运动，并持续交替运行， 需要配合循环属性使用

### 8.4.7 animation-play-state

1)定义：设置对象是否正在运行或已暂停

2)语法：animation-play-state：paused | running

3)参数说明：

paused ： 指定暂停动画

running : 默认值，制定正在运行的动画

### 8.4.8 animation-fill-mode

1)定义：设置对象动画外的状态

2)语法：animation-fill-mode：backwards | both | forwards

3)参数说明：

backwards ： 让元素一开始与 from 状态保持一致

both : 让元素一开始与 from 状态保持一致,结束时候与to状态保持一致

forwards: 结束时候与to状态保持一致

### 8.4.9 animation

1)定义：设置对象所应用的动画特效

2)语法：

animation ： name duration timing-function delay interation-count direction play-state

## 8.5案例

1)兔斯基动画：



2)自行车手：



3)开机动画：



# 第9章 伸缩盒模型

## 9.1新版本与老版本对比

### 9.1.1 flex容器

新版:

display: flex;

display: -webkit-flex;

老版:

display: box;

display: -webkit-box; --- 切记，在移动端使用

**9.1.2 主轴的布局方向**

新版:

flex-direction: row; --- 主轴默认值

flex-direction: column; --- 主轴与侧轴发生对调

老版:

-webkit-box-orient: horizontal; --- 主轴默认值

-webkit-box-orient: vertical; --- 主轴与侧轴发生对调

**9.1.3 主轴的排列方向**

新版:

flex-direction: row-reverse; --- 主轴从左到右，start与end对调

flex-direction: column-reverse; --- 主轴与侧轴发生对调，start与end对调

老版:

-webkit-box-direction: normal; --- 元素排从左到右，默认方向

-webkit-box-direction: reverse; --- 元素排从右到左，但是元素整体都在左边

**9.1.4 富裕空间的管理（主轴）**

新版:

justify-content: flex-start; --- 富裕空间在右侧

justify-content: flex-end; --- 富裕空间在左侧

justify-content: center; --- 富裕空间在两边

justify-content: space-around; --- 富裕空间在左侧包含每一个伸缩项目

justify-content: space-between; --- 每一个伸缩项目包含富裕空间

老版:

-webkit-box-pack: start; --- 默认值：富裕空间在右边

-webkit-box-pack:end; --- 富裕空间在左边

-webkit-box-pack:center; --- 富裕空间包含伸缩项目的整体，使伸缩项目整体在中间，富裕空间在两边

-webkit-box-pack:justify; --- 伸缩项目包含富裕空间

**9.1.5 富裕空间的管理（侧轴）**

新版:伸缩项目的高度又自身内容撑开

align-items: flex-start; --- 富裕空间在下边

align-items: flex-end; --- 富裕空间在上边

align-items: center; --- 富裕空间在两边

align-items: baseline; --- 富裕空间被基线分开

align-items: stretch; --- 拉伸，默认值

老版:

-webkit-box-align: start; --- 富裕空间在下边

-webkit-box-align: end; --- 富裕空间在上边

-webkit-box-align: center; --- 富裕空间在上下两边，富裕空间包含伸缩项目的整体，使伸缩项目整体在中间

**9.1.6 弹性空间（伸缩项目）**

新版:

flex-grow: 1; --- 将富裕空间分配到项目上

老版:

-webkit-box-flex: 1; --- 弹性空间，将富裕空间分配到项目上

**9.2新版本特有属性**

**9.2.1 项目实现换行（flex容器）**

flex-wrap: nowrap; --- 默认值，父元素宽度不够，子元素自身宽度会被压缩

flex-wrap: wrap; --- 父元素宽度不够，子元素会进行换行

flex-wrap: wrap-reverse; --- 子元素换行的同时，侧轴的start与end发生对调

出现flex-wrap：wrap；之后，出现单行的富裕空间

注意：align-items 每一行的赋予空间，align-items: flex-start;

**9.2.2 控制整体侧轴的富裕空间（flex容器）**

align-content: flex-start; --- 项目整体进行打包，放在整体侧轴的start处

align-content: flex-end; --- 项目整体进行打包，放在整体侧轴的end处

align-content: center; --- 项目整体进行打包，放在整体侧轴的center处

注意：align-items 与 align-content 发生冲突时，看元素是否换行

如果没有换行align-items 生效

如果有换行align-content 生效 --- 打包

align-content 生效条件：

1)在伸缩容器中产生换行flex-wrap: wrap;

2)同时设置足够高的容器高度（因为需要整体打包才能看见效果，所以需要高度）

**9.2.3 控制主轴和侧轴的位置及方向（flex容器）**

flex-flow 是 flex-wrap 与 flex-direction 的缩写

flex-flow:wrap-reverse column-reverse;

与flex-wrap: wrap-reverse;flex-direction: column-reverse;实现效果一样

**9.2.4 项目的排列顺序（flex项目）**

order: 1;

order排序，把排序元素先单独拿出来，让剩余元素先正常排列，排完之后，order元素在其后边再进行顺序排列

order是沿着主轴方向进行排序的

**9.2.5 项目自身侧轴的富裕空间（flex项目）**

每一个项目控制自身的侧轴

align-self: flex-start;

align-self: flex-end;

align-self: center;

**9.2.6 收缩率（flex项目）**

当伸缩项目设置宽度，比容器宽度还要大时，元素并没有发生溢出的情况，那此时就出现一个东西叫 收缩因子（收缩率）

flex-grow:拉伸银子 0 ---不拉伸

flex-shrink:收缩因子 默认值：1 --- 收缩

**9.2.7 基准值（flex项目）**

子元素的基准值

flex-basis：1

## 9.3案例

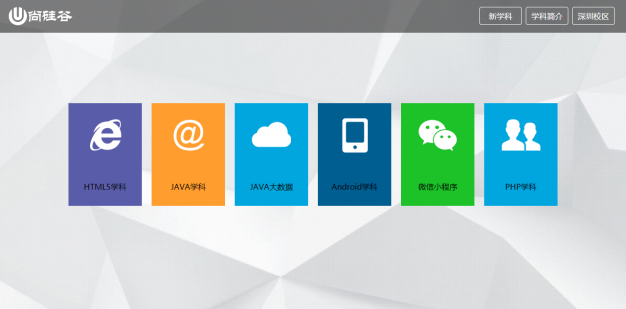
1)天猫墙（自适应布局）：



2)配送图标（自适应布局）：

2018-03-07_120341

3)页面练习：



# 第10章 多列

## 10.1定义

使用CSS3，能够创建多个列来对文本进行布局 ( 就像报纸那样)

## 10.2属性

### 10.2.1 [column-count](http://www.w3school.com.cn/cssref/pr_column-count.asp" \o "CSS3 column-count 属性)

1)定义：规定元素应该被分隔的列数

2)语法：column-count: number;

### 10.2.2 [column-f](http://www.w3school.com.cn/cssref/pr_column-count.asp" \o "CSS3 column-count 属性)ill

1)定义：规定如何填充列

2)语法：column-fill: balance | auto;

### 10.2.3 [column-](http://www.w3school.com.cn/cssref/pr_column-count.asp" \o "CSS3 column-count 属性)gap

1)定义：规定列之间的间隔

2)语法：column-gap: 像素值;

### 10.2.4 [column-](http://www.w3school.com.cn/cssref/pr_column-count.asp" \o "CSS3 column-count 属性)rule

1)定义：所有 column-rule-\* 属性的简写属性(颜色，样式，宽度)

2)语法：column-rule: 颜色，样式，宽度;

3) 列之间规则的颜色：[column-rule-color](http://www.w3school.com.cn/cssref/pr_column-rule-color.asp" \o "CSS3 column-rule-color 属性)

列之间规则的样式：[column-rule-style](http://www.w3school.com.cn/cssref/pr_column-rule-style.asp" \o "CSS3 column-rule-style 属性)

列之间规则的宽度：[column-rule-width](http://www.w3school.com.cn/cssref/pr_column-rule-width.asp" \o "CSS3 column-rule-width 属性)

### 10.2.5 [column-](http://www.w3school.com.cn/cssref/pr_column-count.asp" \o "CSS3 column-count 属性)span

1)定义：元素应该横跨的列数

2)语法：column-span:number

### 10.2.6 [column-](http://www.w3school.com.cn/cssref/pr_column-count.asp" \o "CSS3 column-count 属性)width

1)定义：列的宽度

2)语法：column-width:像素值

**10.3案例**

1)多列新闻



2)瀑布流照片



# 第11章 语义化标签

## 11.1定义

在HTML 5出来之前，我们用div来表示页面头部，章节，页脚等。但是这些div都没有实际意义。各大浏览器厂商分析了上百万的页面，从中发现了DIV名称的通用id名称大量重复。  
例如，很多开发人员喜欢使用div id="footer"来标记页脚内容，所以Html5元素引入了语义化标签（一组新的片段类元素）

## 11.2语义化标签意义



## 11.3语义化标签

### 11.3.1 header

**11.3.2 footer**



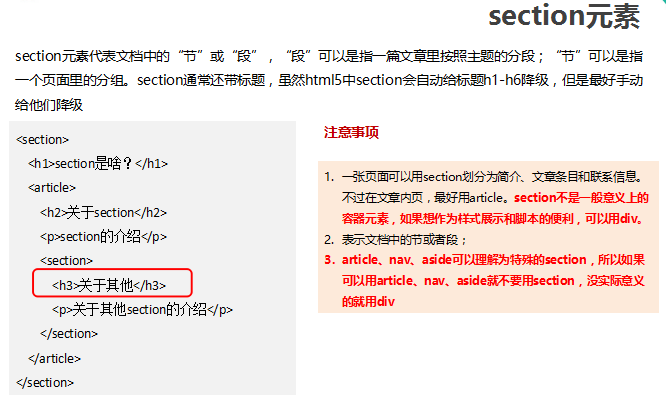
### 11.3.3 hgroup



### 11.3.4 nav



### 11.3.5 section



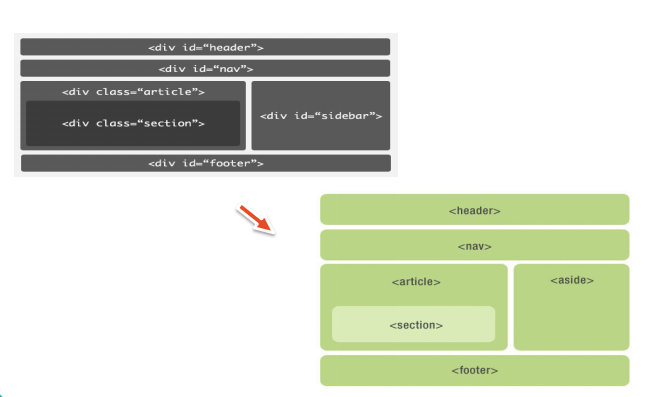
### 11.3.6 aside



### 11.3.6 article



## 11.4语义化标签布局



# 第12章 表单新属性



# 第13章 响应式布局

## 13.1定义

随着浏览器窗口的调整，页面结构发生改变

## 13.2媒体选择器

@media screen and (min-width: 768px){  
 #box{  
 width: 200px;  
 }  
 }

## 13.3案例

大屏幕



中等屏幕



小屏幕



# 第14章 多媒体

## 14.1音频audio

1) 标签：<audio><audio>

2) 属性介绍：

src :引入音频路径

controls="controls" 音频播放

loop="loop" 循环播放

autoplay="autoplay" 自动播放

preload="none" -- 不加载多媒体文件

preload="metadata" -- 加载基本的播放信息

preload="auto" -- 预加载一部分多媒体资源

## 14.2视频video

1) 标签：<video><video>

2) 属性介绍：

src :引入音频路径

controls="controls" 音频播放

autoplay="autoplay" 自动播放

poster : 引入一张图片，视频一上来的预览图

3)常见JS方法：

paly() 从当前位置播放  
 pause() 如果音频在播放中，则暂停播放。

4)常见JS事件：

onloadedmetadata 当音频元数据加载完毕时触发。  
 ontimeupdate 播放过程中实时触发。  
 onvolumechange 声音改变时触发

5)常见JS属性

duration 音频总时长（返回未格式化的秒）  
 currentTime 音频已经播放时长（返回未格式化的秒）  
 volume： 0~1的任意值。控制音量。  
 muted： 布尔值。静音。（ture表示静音，false表示非静音）  
 paused：布尔值。音频文件是否暂停。（ture表示暂停，false表示播放）  
 ended：布尔值。音频文件播放结束（ture表示播放结束，false表示播放中或者暂停）

## 14.3案例



# 第15章Canvas

## 15.1介绍

1)canvas 元素用于在网页上绘制图形

2) 标签:<canvas><canvas>

3) 默认大小：300\*150 （不能css中设置宽度和高度）

## 15.2基本使用步骤

1)获取画布

var canvas = document.querySelector('canvas')

2)获取画笔，或者上下文

var painting = canvas.getContext('2d');

3)开始绘制新路径（从新起笔）

painting.beginPath();

4)绘制图形

## 15.3矩形

1)填充矩形：

方法一：

painting.fillRect(0,0,100,100);

fillRect参数：  
 参数1， 参数2 矩形左上角在画布中的坐标点（坐标的原点画布的左上角）  
 参数3，参数4 矩形宽高

方法二：

painting.rect(50,50,100,100);

注意绘制一个矩形：但是必须配合描边或者填充方法一起使用，否则无法显示图像

painting.fill();

2)描边矩形：

方法一：

painting.strokeRect(120,0,100,100)

strokeRect参数：  
 参数1， 参数2 矩形左上角在画布中的坐标点（坐标的原点画布的左上角）  
 参数3，参数4 矩形宽高

方法二：

painting.rect(50,50,100,100);

painting.stroke();

3)相关设置

填充样式设置：

填充颜色：painting.fillStyle = 'rgba(255,0,0,0.3)';

描边样式设置：

设置线的宽度：painting.lineWidth = 20;  
 设置描边颜色：painting.strokeStyle = 'rgba(0,0,255,0.3)';

描边注意问题:在原有盒子基础上，描边会里外均等分布

## 15.4橡皮擦

绘制一个清除矩形，必要时候一般清空整个画布

painting.clearRect(120,0,100,100);

## 15.5线段

1) 画笔移动的位置，画笔起始位置：

painting.moveTo(200,100);

2) 画笔移动画笔画到的目标位置：

painting.lineTo(200,200);  
3) 描边：

painting.stroke();

4) 设置线段末端：lineCap

butt :线段末端以方形结束。(默认值)  
 round :线段末端以圆形结束  
 square:线段末端以方形结束，但是增加了一个宽度和线段相同，高度是线段厚度一半的矩形区域

5) 设置线段连接处：lineJoin

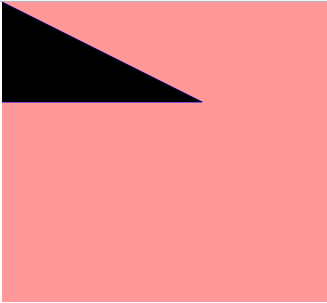
round : 圆角

bevel : 斜角

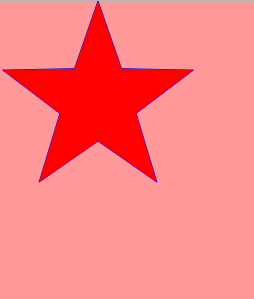
miter : 直角

6) 线段案例

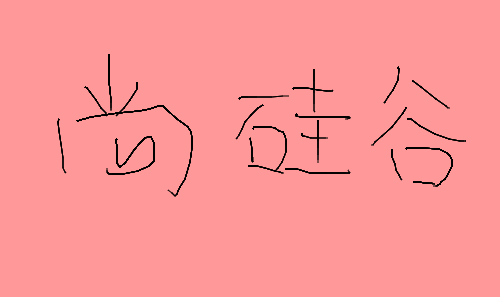
三角形



五角星



签名



## 15.6圆形与圆弧

### 15.6.1圆形

1) 语法：

painting.arc(200,200,100,0,360/180\*Math.PI,false);

2) 参数说明：

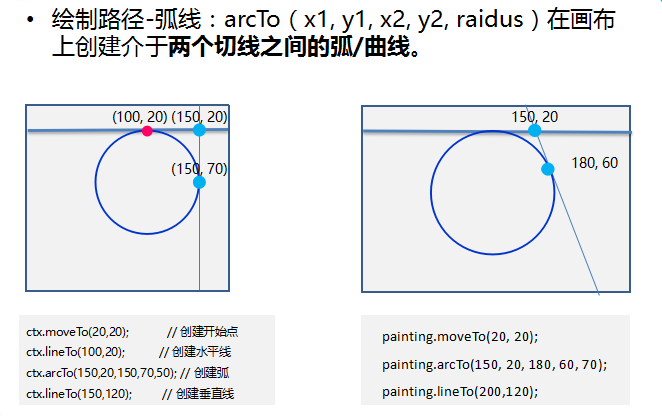
参数1，参数2：圆心坐标

参数3：半径

参数4，参数5：圆的起始弧度与结束弧度

参数6：绘制图形是否为顺时针

### 15.6.2圆弧



## 15.7 变换

1) 位移：painting.translate(100,100);

2) 位移注意问题：位移元素:改变图形的所有坐标点，相当于重新绘制坐标点，位移坐标写好之后，后续的图形坐标要参照此时的新坐标，但是，之前的盒子不会受影响

3) 缩放：painting.scale(2,0.5);

4) 旋转：painting.rotate(5/180\*Math.PI)

5)案例钟表



## 15.8 加载图片

1) 加载图片：painting.drawImage();

2) 步骤：

首先，需要使用图片就得由img标签

其次，引入图片路径

然后，等图片加载完成后再去设置图片显示

最后，图片显示

drawImage():

参数1：图片的dom对象

参数2，参数3：图片在画布中显示的初始位置

参数4，参数5：图片在画布中的宽度与高度

3) 案例：飞鸟



## 15.9 渐变

1) 线性渐变：painting.createLinearGradient(20,20,100,100);

参数1，参数2，参数3，参数4 的连线决定了渐变的方向和区间

2) 添加渐变色：

addColorStop(0,'red')

参数1 ： 只能是0--1的小数 渐变的起始位置  
 参数2 ：颜色

3) 径向渐变：

painting.createRadialGradient(200,200,50,130,200,100);

参数1，参数2，参数3，第一个小圆的圆心和半径  
 参数4，参数5，参数6，第二个大圆的圆心和半径

## 15.10 文字

1) 填充文字：painting.fillText('你好啊',50,50);

2) 镂空文字：painting.strokeText('你好啊',50,50);

3) 文字样式：painting.font = 'bold 40px 微软雅黑';

4)水平对齐方式

painting.textAlign = 'start'; --- 默认值

painting.textAlign = 'center';

painting.textAlign = 'end';

5)垂直对齐方式

painting.textBaseline = 'top'; 文字顶部有线有缝隙  
 painting.textBaseline = 'hanging'; 文字顶部没有线有缝隙  
 painting.textBaseline = 'middle';  
 painting.textBaseline = 'bottom'; 文字底部有线有缝隙  
 painting.textBaseline = 'alphabetic'; 默认值

## 15.11 阴影

painting.shadowColor = 'blue'; 阴影颜色设置  
 painting.shadowBlur = 20; 阴影模糊程度  
 painting.shadowOffsetX = 100; 阴影水平方向偏移量  
 painting.shadowOffsetY = 50; 阴影垂直方向偏移量

## 15.12 像素操作

1)读取像素：提取矩形中的信息  
 painting.getImageData(100,100,100,100);  
 返回值：返回 40000个像素点，及图形的宽度与高度

width:该区域横向上像素点的个数  
 height:该区域纵向上像素点的个数  
 data:该区域所有像素点的rgba信息

2)写入像素：

painting.putImageData(填入的对象,填充区域水平坐标点,填充区域垂直坐标点);

3)案例：



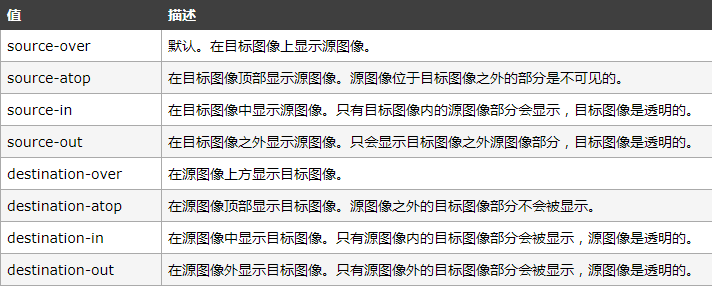
## 15.13 图片合成

1)定义：globalCompositeOperation属性设置或返回如何将一个源（新的）图像绘制到目标（已有）的图像上

源图像 = 您打算放置到画布上的绘图

目标图像 = 您已经放置在画布上的绘图

2) 属性值：



3) 案例：刮刮卡

