HTML5+CSS3

复习

CSS2 复习

```
.box{
        /*文本颜色*/
        color:red;
        /*文本字体大小，浏览器默认大小是 16px,最小 12px*/
        font-size:16px;
        /*字体的粗细：lighter  normal  bold*/
        font-weight: lighter;
        /*字体的样式 italic 斜体字  oblique 倾斜的 normal 默认*/
        font-style:oblique;
        /*行高可以设置成字体大小的倍数。*/
        line-height:1.3;
        /*首行缩进 2 个字符*/
        text-indent:2em;
        width: 300px;
        text-align: center;
        /*可以设置两个值 第一个值：线的颜色   第二个值：underline 下划线
         line-through 中划线  overline 上划线*/
        text-decoration: #34A853 overline;
        /*规定段落中的文本不进行换行*/
        white-space: nowrap;
        /*允许在单词内换行*/
        word-break:break-all;
        /*
        em 也可以设置字体的大小，em 指的是文字的高度。
        但对英文来说
        顶线 中线 基线 底线行高：基线到基线的距离。
        */
    }
```

第 1 章 选择器

1.1. 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，.....

```
/*父子选择器*/
 ul > li {
     margin-top: 2px;
     background-color: #90ee90;
 }

 /*相邻兄弟选择器，不包括上面的兄弟元素*/
 .second + li {
```

```
9.      background-color: yellow;
10. }
11.
12. /*所有兄弟选择器，不包含上面的兄弟元素*/
13. .second ~ li {
14.     background-color: #dd0000;
15. }
16.
17. .second {
18.     background-color: #ff4a5f;
19. }
20.
21. .first {
22.     background-color: pink;
23. }
24.
25. /*群组选择器，一次给多个标签或者 class 或者 id 设置样式。*/
26. .first, .second {
27.     background-color: #dd0000;
28. }
```

1.2. 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']

```
/*    选择拥有 title 属性的 li 标签*/
ul > li[title] {
    /*background-color: firebrick;*/
    margin-top: 2px;
}

/*    选择拥有 title 属性，并且属性值为 bbox 的 li 标签*/
li[title="bbox"] {
    /*background-color: #90ee90;*/
    margin-top: 2px;
}

/*选择拥有 title 属性，并且属性值中包含单词 box 的 li 标签*/
li[title~="box"] {
    background-color: pink;
    margin-top: 2px;
}

/*选择拥有 title 属性，并且属性值以 a 字母开头的 li 标签*/
```

```
li[title^="a"] {
    background-color: gray;
}

/*选择拥有 title 属性，并且属性值以 x 字母结尾的 li 标签*/
li[title$="x"] {
    background-color: deepskyblue;
}

/*选择拥有 title 属性，并且属性值中包含字符串 box 的 li 标签*/
li[title*="box"] {
    background-color: purple;
}

input[type="text"] {
    width: 200px;
    height: 200px;
    background-color: #90ee90;
}
```

1.3. 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）

01.目标伪类选择器（轮播）.gif

2) checked 状态类案例（自定义选框）

3) CSS3 结构类案例（尺子）

1.4. 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)

1.5. 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

1.6. 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

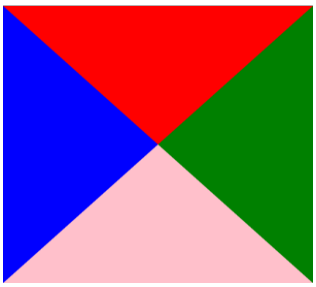
当拿 50%时， 宽等于高 ---- 圆形

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

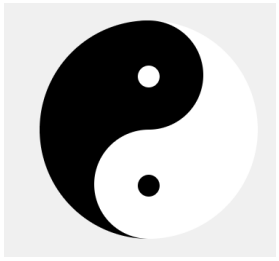


2)搜狐评论框

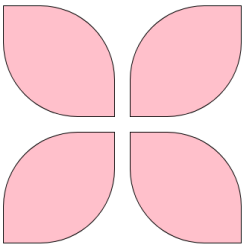
3)半圆

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

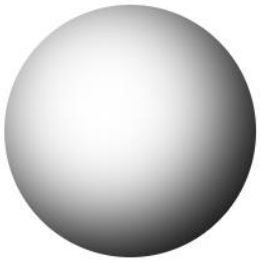
inset(默认没有，也就是默认是外阴影)

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)

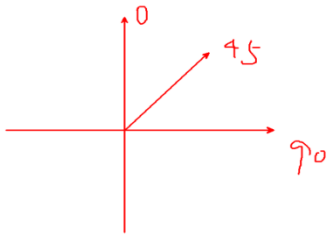
5.1.3 方向分类

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);

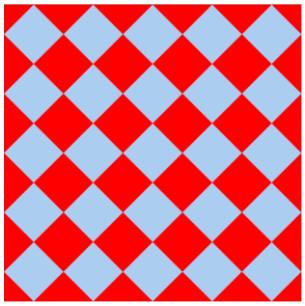
5.1.6 重复渐变

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

15.渐变（发廊灯）.gif

3)光斑动画

15.渐变（光斑动画）.gif

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

16.幽灵按钮.gif

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

17.2d变换（瓶体旋转）.gif

2)菜单按钮

18.2d变换（菜单按钮）.gif

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

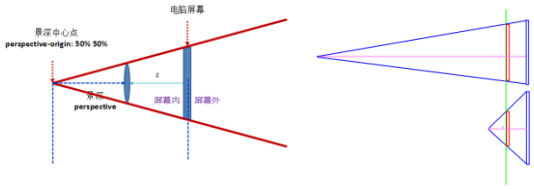
1)定义：实现元素在 3D 空间中的近大远小的效果

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



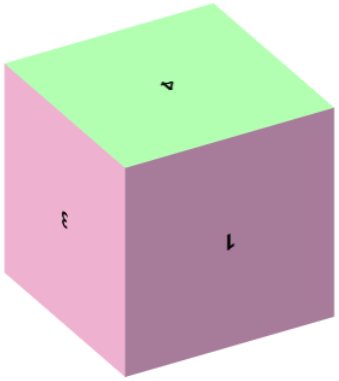
7.3.4 变换基点

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

20.3d变换（旋转盒子）.gif

2)3d 相册

21.3d变换（旋转相册）.gif

3)天猫商品墙

22.3d(天猫商品墙).gif

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 form 状态保持一致

both : 让元素一开始与 form 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：

23.动画（兔斯基）.gif

2)自行车手：

24.动画（自行车手）.gif

3)开机动画：

26.动画（开机动画）.gif

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

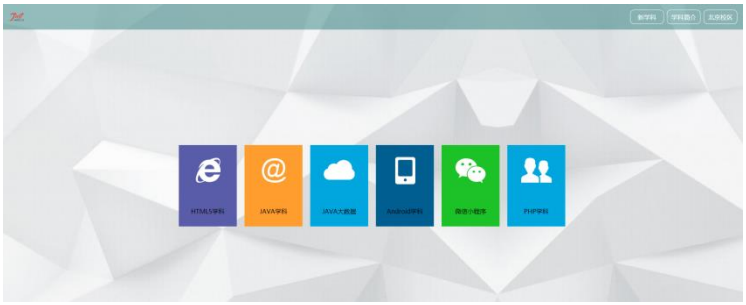
flex-basis：0

9.3 案例

1)天猫墙（自适应布局）：

2)配送图标（自适应布局）：

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

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-fill

1)定义：规定如何填充列

2)语法：column-fill: balance | auto;

10.2.3 column-gap

1)定义：规定列之间的间隔

2)语法：column-gap: 像素值;

10.2.4 column-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

列之间规则的样式：column-rule-style

列之间规则的宽度：column-rule-width

10.2.5 column-span

1)定义：元素应该横跨的列数

2)语法：column-span:number

10.2.6 column-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

习近平在亚太经合组织工商领导人峰会上的演讲

布斯塔曼特主席先生，亚太工商界各位代表，女士们，先生们，朋友们：
很高兴同大家相会在美丽的利马。中国和秘鲁相距遥远，《英汉大词典》有一个词语，叫“从中国到秘鲁”，意思是走遍天下。今天，我们不远万里来到利马，共同的目标是探讨推进亚太发展思路和举措。
两个月前，二十国集团领导人杭州峰会成功举行。我同各国领导人一道，深入讨论世界经济面临的突出问题，形成许多重要共识。大家普遍认为，全球经济复苏仍然乏力，增长动力不足，经济全球化遇到波折，贸易和投资低迷，全球性挑战加剧世界经济不确定性。面对风险和挑战，各方要发扬同舟共济、合作共赢的伙伴精神，加强宏观政策协调，创新经济增长方式，构建开放型世界经济，推动强劲、可持续、平衡、包容增长。

当前，亚太总体保持平衡发展态势，但也面临挑战，处在发展关键当口。作为全球经济规模最大、最具发展活力的地区，亚太要勇于担当，发挥引领作用，采取有力协调行动，为世界经济复苏注入新动力，为世界经济增长开辟新道路。
第一，促进经济一体化，建设开放型经济。开放是亚太经济的生命线。20多年来，亚太经合组织成员坚持贸易自由化和便利化，贸易量年均增长8%，是同期经济增速的两倍多，为亚太经济增长提供了稳定动力。近年来，国际贸易发展进入低迷期。世界贸易组织预计，今年全球贸易增幅可能连续第五年低于经济增速。亚太面临相同压力，也亟待解决区域经济合作碎片化等挑战。任何区域贸易安排要获得广泛支持，必须坚持开放、包容、普惠、共赢。我们应该构建平等协商、共同参与、普遍受益的区域合作框架，封闭和排他性安排不是正确选择。
建设亚太自由贸易区，是事关亚太长远繁荣的战略举措，工商界朋友称之为“亚太经合组织之梦”。我们要坚定推进亚太自由贸易区建设，为亚太开放型经济提供制度保障。要重振贸易和投资的引擎作用，增强自由贸易安排开放性和包容性，维护多边贸易体制。
当前，围绕经济全球化有很多讨论，支持者有之，质疑者亦有之。总体而言，经济全球化符合经济规律，符合各方利益。同时，经济全球化是一把双刃剑，既为全球发展提供强劲动能，也带来一些新情况新挑战，需要认真面对。新一轮科技和产业革命正孕育兴起，国际分工体系加速演变，全球价值链深度重塑，这些都给经济全球化赋予新的内涵。亚太经合组织成立于经济全球化不断推进的时期，亚太取得的发展成就同经济全球化密不可分。我们要认识和把握自身发展和外部环境的互动变化，捕捉新机遇，定位新角色，创立新优势。同时，全球化也提出需要深入研究的新问题，我们要积极引导经济全球化发展方向，着力解决公平公正问题，让经济全球化进程更有活力、更加包容、更可持续，增强广大民众参与感、获得感、幸福感。
第二，促进互联互通，实现联动发展。互联互通是释放发展潜力的重要手段，也是实现联动发展的基础前提。我们要推动建立覆盖整个亚太的全方位、复合型互联互通网络。今年，亚太经合组织会议时隔8年重回拉美举行，我们要把握这一契机，推动太平洋两岸互联互通建设彼此对接，在更广范围内辐射和带动实体经济发展。要深入落实北京会议制定的互联互通蓝图，完善基础设施、制度规章、人员交流三位一体的互联互通架构，确保2025年实现全面联接的目标。
3年前，我提出“一带一路”倡议，就是要以互联互通为着力点，促进生产要素自由便利流动，打造多元合作平台，实现共赢和共享发展。截至目前，共有100多个国家和国际组织积极参与和支持，结成志同道合、互信友好、充满活力的“朋友圈”。亚洲基础设施投资银行开业运营，丝路基金顺利组建，一大批重大项目付诸实施，产生巨大经济社会效益。中国将同各方一道，秉持共商、共建、共享原则，推进政策沟通、道路联通、贸易畅通、货币流通、民心相通，实现发展战略对接，深化互利合作，为区域经济发展和民生改善注入强大动力。我们欢迎各方参与到合作中来，共享机遇，共迎挑战，共谋发展。

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

header元素

header 元素代表“网页”或“section”的页眉。通常包含h1-h6元素或hgroup，作为整个页面或者一个内容块的标题。也可以包裹一节的目录部分，一个搜索框，一个nav，或者任何相关logo。整个页面没有限制header元素的个数，可以拥有多个，可以为每个内容块增加一个header元素

```
<header>
    <h1>网站标题</h1>
    <h2>网站副标题</h2>
</header>
```

注意事项

1. 可以是“网页”或任意“section”的头部部分；
2. 没有个数限制。
3. **如果hgroup或h1-h6自己就能工作的很好，那就不要用header。**

11.3.2 footer

footer元素

footer元素代表“网页”或“section”的页脚，通常含有该节的一些基本信息，譬如：作者，相关文档链接，版权资料。如果footer元素包含了整个节，那么它们就代表附录，索引，提拔，许可协议，标签，类别等一些其他类似信息。

```
<footer>
  地址：昌平区平西王府公交站东尚硅谷教学楼邮箱：info@atguigu.com 微博：weibo.com/u/3272253032
</footer>
```

注意事项

1. 可以是“网页”或任意“section”的底部部分；
2. 没有个数限制，除了包裹的内容不一样，其他跟header类似。

11.3.3 hgroup

hgroup元素

hgroup元素代表“网页”或“section”的标题，当元素有多个层级时，该元素可以将h1到h6元素放在其内，譬如文章的主标题和副标题的组合

```
<header>
  <hgroup>
    <h1>网站标题</h1>
    <h2>网站副标题</h2>
  </hgroup>
</header>
```

注意事项

1. 如果只需要一个h1-h6标签就不用hgroup
2. 如果有连续多个h1-h6标签就用hgroup
3. 如果有连续多个标题和其他文章数据，h1-h6标签就用hgroup包住，和其他文章元数据一起放入header标签。

11.3.4 nav

nav元素

nav元素代表页面的导航链接区域。用于定义页面的**主要导航部分。**

```
<nav>
  <ul>
    <li>HTML 5</li>
    <li>CSS3</li>
    <li>JavaScript</li>
  </ul>
</nav>
```

注意事项

1. 用在整个页面主要导航部分上。

11.3.5 section

section元素

section元素代表文档中的“节”或“段”，“段”可以是指一篇文章里按照主题的分段；“节”可以是指一个页面里的分组。section通常还带标题，虽然html5中section会自动给标题h1-h6降级，但是最好手动给他们降级

```
<section>
  <h1>section是啥？</h1>
  <article>
    <h2>关于section</h2>
    <p>section的介绍</p>
    <section>
      <h3>关于其他</h3>
      <p>关于其他section的介绍</p>
    </section>
  </article>
</section>
```

注意事项

1. 一张页面可以用section划分为简介、文章条目和联系信息。不过在文章内页，最好用article。**section不是一般意义上的容器元素，如果想作为样式展示和脚本的便利，可以用div。**
2. 表示文档中的节或者段；
3. **article、nav、aside可以理解为特殊的section，所以如果可以用article、nav、aside就不要用section，没实际意义的就用div**

11.3.6 aside

aside元素

aside元素被包含在article元素中作为主要内容的附属信息部分，其中的内容可以是与当前文章有关的相关资料、标签、名词解释等。（特殊的section）

在article元素之外使用作为页面或站点全局的附属信息部分。最典型的是侧边栏，其中的内容可以是日志串连，其他组的导航，甚至广告，这些内容相关的页面。

```
<article>
  <p>文章内容</p>
  <aside>
    <h1>作者简介</h1>
    <p>张三，网络写手。</p>
  </aside>
</article>
```

注意事项

1. aside在article内表示主要内容的附属信息，
2. **在article之外则可做侧边栏，没有article与之对应，最好不用。**
3. 如果是广告，其他日志链接或者其他分类导航也可以用

11.3.6 article

article元素

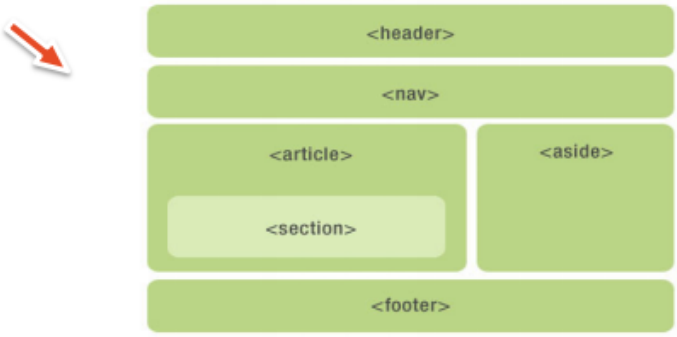
article元素最容易跟section和div容易混淆，其实article代表一个在文档，页面或者网站中自成一体的内容，其目**的是为了让开发者独立开发或重用**。譬如论坛的帖子，博客上的文章，一篇用户的评论，一个互动的widget小工具。

```
<article>
  <h1>一篇文章</h1>
  <p>文章内容..</p>
  <footer>
    <p><small>版权：尚硅谷，作者：xxx</small></p>
  </footer>
</article>
```

注意事项

1. 一张页面可以用section划分为简介、文章条目和联系信息。不过在文章内页，最好用article。section不是一般意义上的容器元素，如果想作为样式展示和脚本的便利，可以用div。
2. 表示文档中的节或者段；
3. article、nav、aside可以理解为特殊的section，所以如果可以用article、nav、aside就不要用section，没实际意义的就用div

11.4 语义化标签布局



第 12 章 表单新属性

文本输入	
email	电子邮箱输入框
tel	电话号码输入框（即使输入字母也不会校验）
url	网页URL输入框
search	搜索输入框（自动填充默认打开）
number	数字选择
number - min、max、step	最小，最大取值，间隔取值
数据选择	
range	特定范围内的数值选择器
range - min、max、step	最小，最大取值，间隔取值
color	颜色选择
datetime-local	日期+时间选择
time	时间选择
date	日期选择
week	周选择
month	月选择
新的属性	
placeholder	输入框提示信息
autocomplete	是否保存用户输入值。**默认为on**，关闭提示选择off
autofocus	自动获取输入焦点
required	验证类：不能为空值。

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

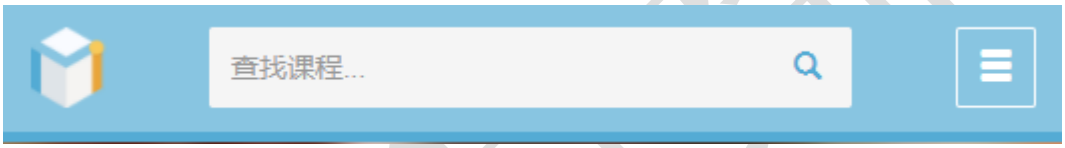
```
@media screen and (min-width: 768px){
        #box{
                width: 200px;
    }
    }
```

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

播放 暂停

播放 暂停

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

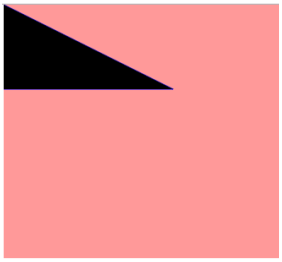
round： 圆角

bevel： 斜角

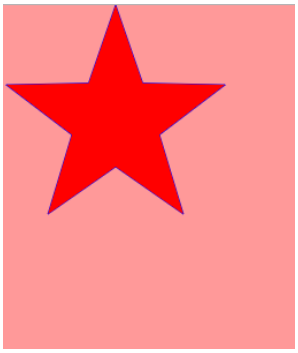
miter： 直角

6) 线段案例

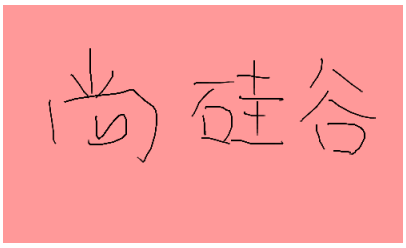
三角形



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

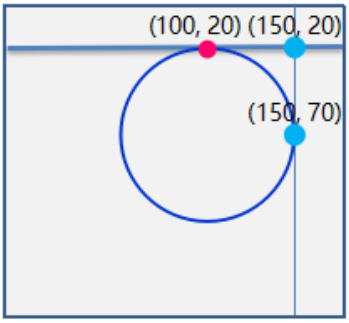
参数 3：半径

参数 4，参数 5：圆的起始弧度与结束弧度

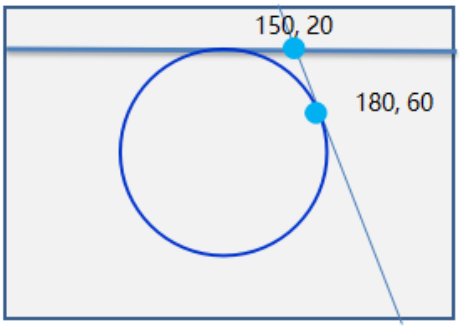
参数 6：绘制图形是否为顺时针

15.6.2 圆弧

- 绘制路径-弧线：arcTo（x1, y1, x2, y2, raidus）在画布上创建介于**两个切线之间的弧/曲线。**




```
ctx.moveTo(20,20);          // 创建开始点
ctx.lineTo(100,20);         // 创建水平线
ctx.arcTo(150,20,150,70,50); // 创建弧
ctx.lineTo(150,120);        // 创建垂直线
```




```
painting.moveTo(20, 20);
painting.arcTo(150, 20, 180, 60, 70);
painting.lineTo(200,120);
```

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

33.canvas（飞鸟）.gif

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：

值	描述
source-over	默认。在目标图像上显示源图像。
source-atop	在目标图像顶部显示源图像。源图像位于目标图像之外的部分是不可见的。
source-in	在目标图像中显示源图像。只有目标图像内的源图像部分会显示，目标图像是透明的。
source-out	在目标图像之外显示源图像。只会显示目标图像之外源图像部分，目标图像是透明的。
destination-over	在源图像上方显示目标图像。
destination-atop	在源图像顶部显示目标图像。源图像之外的目标图像部分不会被显示。
destination-in	在源图像中显示目标图像。只有源图像内的目标图像部分会被显示，源图像是透明的。
destination-out	在源图像外显示目标图像。只有源图像外的目标图像部分会被显示，源图像是透明的。

3) 案例：刮刮卡

35.canvas(刮刮卡).gif