

[笔记][LIKE-CSS3][01-CSS3边框和渐变]

前端

[笔记][LIKE-CSS3][01-CSS3边框和渐变]

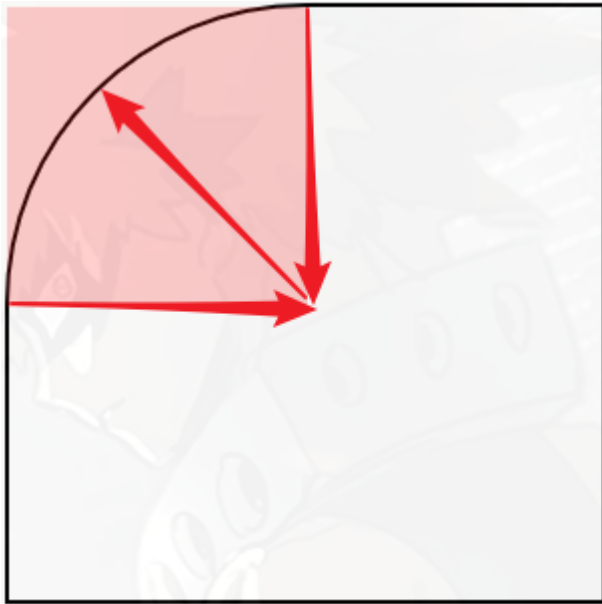
01. 课程对应代码资料下载
02. 边框圆角基本使用(掌握)
03. 边框圆角注意点 (掌握)
04. 绘制半圆和椭圆(理解)
05. 边框圆角练习1(理解)
06. 边框圆角练习2(理解)
07. 边框图片
08. 边框图片补充(掌握)
09. 滑动门(掌握)
10. 边框图片按钮(掌握)
11. vertical-align(掌握)
12. 线性渐变上(掌握)
13. 线性渐变注意点(掌握)
14. 线性渐变练习(理解)
15. 卡拉OK效果(理解)
16. 径向渐变(掌握)

01. 课程对应代码资料下载

略

02. 边框圆角基本使用(掌握)

边框圆角快捷键： `bdrs`



绘制圆角的原理：`border-radius: 100px 0 0 0;`

- 四个参数分别是：左上角，右上角，右下角，左下角
- 左上角位置设置为 `100px`，则会从左上角向右向下偏移 `100px`，并以 `100px` 为半径做 `1/4` 个圆。

如果要画一个圆：`border-radius: 50%;`

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>001-边框圆角</title>
  <style>
    *{
      margin: 0;
      padding: 0;
    }

    .father{
      width: 200px;
      height: 200px;
      border: 1px solid #000;
      box-sizing: border-box;
      margin: 100px auto;
      /* border-radius: 100px 100px 100px 100px; */
      border-radius: 100px 0 0 0;
    }

    .son{
      width: 100px;
      height: 100px;
      background-color: rgba(255, 0, 0, 0.2);
    }
  </style>
```

```
</head>
<body>
<!--
1. 什么是边框圆角?
将直角的边框变为圆角的边框

2. 边框圆角的格式?
border-radius: 左上 右上 右下 左下;

3. 将正方形变为圆形的技巧
border-radius: 50%;

4. 系统如何绘制圆角?
首先根据指定的值找到圆心
按照指定的值作为半径绘制圆弧

-->
<div class="father">
  <div class="son"></div>
</div>
</body>
</html>
```

03. 边框圆角注意点（掌握）

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>119-边框圆角注意点</title>
  <style>
    *{
      margin: 0;
      padding: 0;
    }

    .one{
      width: 200px;
      height: 200px;
      border: 1px solid #000;
      box-sizing: border-box;
      margin: 100px auto;
      /* border-radius: 100px 100px 100px 100px; */
      /* border-radius: 100px 50px 100px 100px; */
    }
  </style>

```

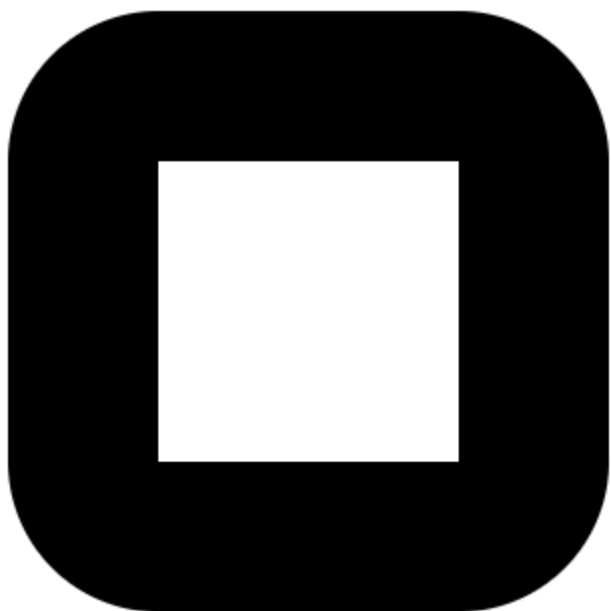
```

        /* 当省略了某一个角的值之后，系统会自动参考对角的值。 */
        /* border-radius: 100px 50px 100px; */
        border-radius: 100px 50px;
    }

    .two {
        width: 200px;
        height: 200px;
        border: 50px solid #000;
        box-sizing: border-box;
        margin: 100px auto;
        /* 当只设置了一个值的时候，其它三个角都会参考这个值。 */
        /* border-radius: 100px; */
        /* 当边框圆角的值 > 边框宽度的时候，外边框和内边框都会变成圆角 */
        /* 当边框圆角的值 <= 边框宽度的时候，外边框是圆角，内边框是直角 */
        border-radius: 50px;
    }
</style>
</head>
<body>
    <!-- <div class="one"></div> -->
    <div class="two"></div>
</body>
</html>

```

下图是 `border-radius` 等于 `50px` 的时候，外边框有圆角，内边框无圆角
一旦大于 `50px`，内外都会有圆角



04. 绘制半圆和椭圆(理解)

单独设置某一个角的边框圆角快捷键 `btls`，这是 `border-top-left-radius` 的缩写。

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    * {
      margin: 0;
      padding: 0;
    }

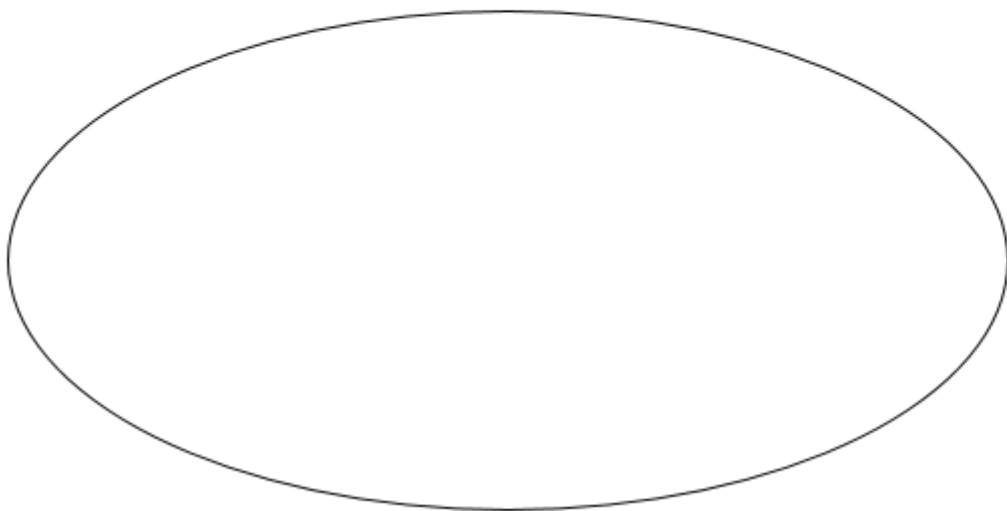
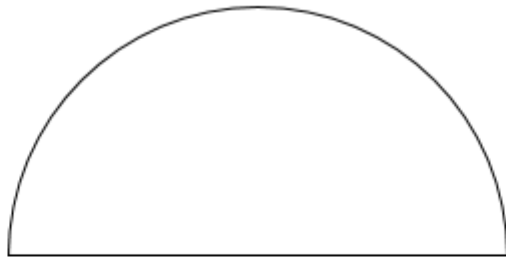
    .one{
      /* width: 200px;
      height: 100px;
      border: 1px solid #000;
      box-sizing: border-box;
      margin: 100px auto;
      border-radius: 100px 100px 0 0; */
      width: 200px;
      height: 100px;
      border: 1px solid #000;
      box-sizing: border-box;
      margin: 100px auto;
      /* 可以通过border-xxx-xxx-radius的方式单独设置某一个角的值 */
      /* border-xxx-xxx-radius 接收两个参数，第一个表示水平方向，第二个表示
垂直方向 */
      /* border-xxx-xxx-radius 如果只传递了一个参数，那么垂直方向和水平方向
的值一样 */

      /*          水平方向 垂直方向 */
      border-top-left-radius: 100px 100px;
      /* 如果省略垂直方向，垂直方向和水平方向一样 */
      border-top-right-radius: 100px;
    }

    .two {
      width: 400px;
      height: 200px;
      border: 1px solid #000;
      box-sizing: border-box;
      margin: 300px auto;
      /* 绘制椭圆：
      设置水平方向为宽度的一半，垂直方向为高度的一半。 */
      border-top-left-radius: 200px 100px;
      border-top-right-radius: 200px 100px;
      border-bottom-left-radius: 200px 100px;
      border-bottom-right-radius: 200px 100px;
    }
  </style>
</head>
</html>

```

```
    }  
    </style>  
</head>  
<body>  
  <div class="one"></div>  
  <div class="two"></div>  
</body>  
</html>
```



05. 边框圆角练习1(理解)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>004-绘制太极八卦图</title>
  <style>
    * {
      margin: 0;
      padding: 0;
    }

    .all {
      width: 400px;
      height: 400px;
      border: 1px solid #000;
      box-sizing: border-box;
      margin: 200px auto;
      position: relative;
    }

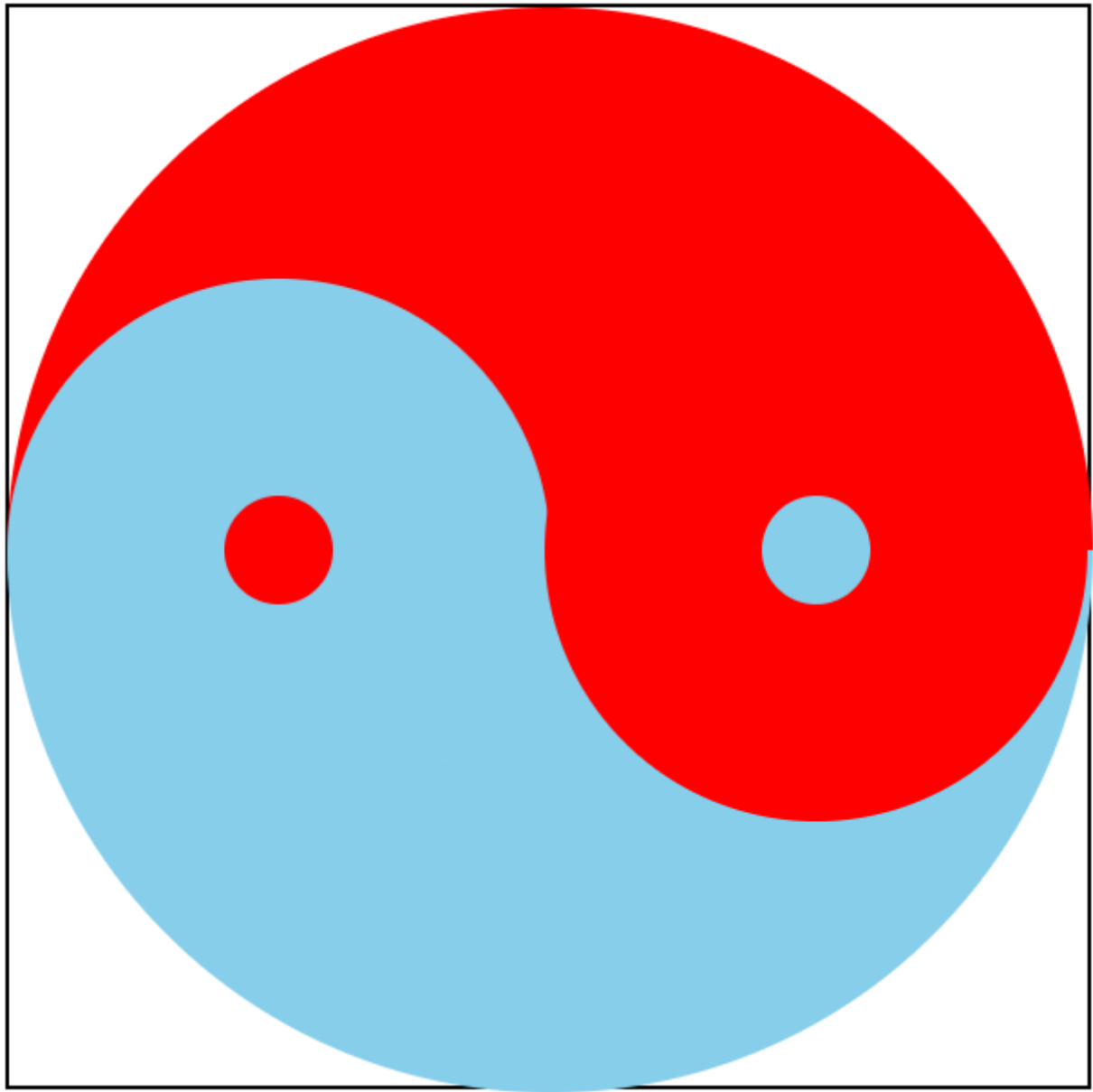
    .top {
      width: 400px;
      height: 200px;
      background-color: red;
      border-top-left-radius: 200px;
      border-top-right-radius: 200px;
    }

    .bottom {
      width: 400px;
      height: 200px;
      background-color: skyblue;
      border-bottom-left-radius: 200px;
      border-bottom-right-radius: 200px;
    }

    .left {
      width: 200px;
      height: 200px;
      background: red;
      border: 80px solid skyblue;
      box-sizing: border-box;
      border-radius: 50%;
    }
```

```
        position: absolute;
        left: 0;
        top: 100px;
    }

    .right {
        width: 200px;
        height: 200px;
        background: skyblue;
        border: 80px solid red;
        box-sizing: border-box;
        border-radius: 50%;
        position: absolute;
        right: 0;
        top: 100px;
    }
</style>
</head>
<body>
<div class="all">
    <!-- 上面的半圆 -->
    <div class="top"></div>
    <!-- 下面的半圆 -->
    <div class="bottom"></div>
    <!-- 左边的小圆 -->
    <div class="left"></div>
    <!-- 右边的小圆 -->
    <div class="right"></div>
</div>
</body>
</html>
```

06. 边框圆角练习2(理解)

一个圆盖住另一个圆，所以用定位，而不是用浮动。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>005-绘制跳动的心</title>
  <style>
    * {
      margin: 0;
      padding: 0;
```

```
}

.all {
    width: 450px;
    height: 400px;
    /* border: 1px solid black; */
    box-sizing: border-box;
    margin: 200px auto;
    position: relative;
}

.left, .right {
    width: 250px;
    height: 250px;
    background: red;
    border-radius: 50%;
    position: absolute;
    top: 0;
    box-shadow: 0 0 40px red;
}

.left {
    left: 0;
}

.right {
    right: 0;
}

.bottom {
    width: 250px;
    height: 250px;
    background-color: red;
    position: absolute;
    bottom: 50px;
    left: 50%;
    /* margin-left: -125px; */
    /* 技巧：在企业开发中推荐使用如下的方式让元素居中
        水平居中： left: 50%; transform: translateX(-50%);
        垂直居中： top: 50%; transform: translateY(-50%);
    */
    transform: translateX(-50%) rotateZ(45deg);
    box-shadow: 0 0 40px red;
}

.all {
    animation: sport 1s linear 0s infinite normal;
}

@keyframes sport {
    0% {
        transform: scale(1);
    }
}
```

```
    }
    50% {
        transform: scale(1.1);
    }
    100% {
        transform: scale(1);
    }
}
</style>
</head>
<body>
    <div class="all">
        <!-- 左边的小圆 -->
        <div class="left"></div>
        <!-- 右边的小圆 -->
        <div class="right"></div>
        <!-- 底部的尖尖 -->
        <div class="bottom"></div>
    </div>
</body>
</html>
```



07. 边框图片

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>006-边框图片</title>
  <style>
    * {
      margin: 0;
      padding: 0;
    }

    div {
      width: 300px;
      height: 300px;
      border: 70px solid #000;
      box-sizing: border-box;
      margin: 200px auto;
      /* 告诉浏览器，让哪一张图片成为边框 */
      /* 注意点：
        如果只通过 source 指定了哪一张图片作为边框的图片，默认情况下会将图
        片放到边框的四个顶点。
        如果设置了边框图片，那么就不会显示边框颜色，边框图片的优先级高于边框
        颜色
      */
      border-image-source: url("images/border.jpg");
      /* background: skyblue; */
      /* 告诉浏览器，如何对指定的边框图片进行切割 */
      /* 注意：没有单位！ */
      border-image-slice: 70 70 70 70;
      /* 告诉浏览器，边框图片显示的宽度，并不是指定边框的宽度 */
      /* 注意点：如果通过 border-image-width 指定了边框图片的宽度，那么默认
        的边框宽度就会失效 */
      /* 其实也不是失效，只不过会按照 border-image-width 指定的宽度显示 */
      border-image-width: 10px;
      /* 告诉浏览器，除了边框图片四个角以外的图片，如何填充，默认是拉伸 */
      border-image-repeat: stretch;
      /* 除了四个顶点，重复边框图片 */
      border-image-repeat: repeat;
      /* 除了四个顶点，重复边框图片，中间的图片重复整数次 */
      border-image-repeat: round;
    }
  </style>
</head>
```

```
<body>
  <div></div>
</body>
</html>
```

08. 边框图片补充(掌握)

- 边框可以向外移动

```
/* 告诉浏览器，边框浏览器需要向外移动多少 */
/*           上   右   下   左 */
border-image-outset: 10px 30px 50px 70px;
```

- 在切割方式中，设置填充边框图片中间的图片

```
border-image-slice: 70 70 70 70 fill;
```

- 连写形式

```
/*
  border-image: 资源地址 切割方式 填充模式
*/
border-image: url("images/border.jpg") 70 fill stretch;
```

09. 滑动门(掌握)

滑动门第一版：使用了两张图片

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>滑动门</title>
  <style>
    * {
      margin: 0;
      padding: 0;
    }
  </style>
</head>
<body>
```

```

/* div {
    width: 200px;
    height: 60px;
    background: red url("images/border/nav-all.jpg") no-repeat;
    margin: 200px auto;
    background-size: 200px 60px;
} */

.main {
    width: 100px;
    height: 60px;
    background: red url("images/border/nav-left2.jpg") no-repeat;
    margin: 200px auto;
}

.right {
    width: 20px;
    height: 60px;
    background: url("images/border/nav-right.jpg") no-repeat;
    float: right;
}
</style>
</head>
<body>
    <div class="main">
        <div class="right"></div>
    </div>
</body>
</html>

```

滑动门第二版：使用了三张图片

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>滑动门2</title>
    <style>
        .all {
            width: 100px;
            height: 60px;
            background: red url("images/border/nav-center.jpg") repeat-x;
            margin: 200px auto;
        }

        .left {
            width: 100%;
            height: 100%;

```

```

        background:url("images/border/nav-left.jpg") no-repeat;
    }

    .right {
        width: 100%;
        height: 100%;
        background: url("images/border/nav-right.jpg") center right n
o-repeat;
    }
</style>
</head>
<body>
    <div class="all">
        <div class="left">
            <div class="right"></div>
        </div>
    </div>
</body>
</html>

```

10. 边框图片按钮(掌握)

上右下左切四刀，分九份

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>010-边框图片按钮</title>
    <style>
        * {
            margin: 0;
            padding: 0;
        }

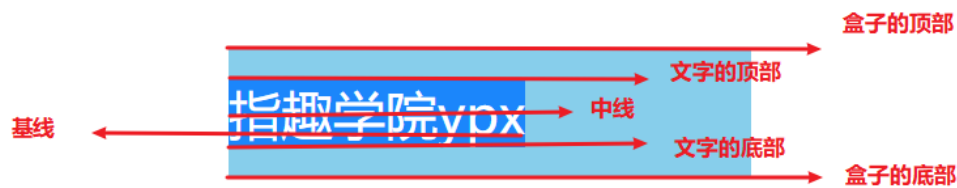
        div {
            width: 100px;
            height: 20px;
            border: 20px solid #000;
            border-image: url("images/border/nav-all.jpg") 20 fill repea
t;

            margin: 200px auto;
        }
    </style>

```

```
</head>
<body>
  <div></div>
</body>
</html>
```

11. vertical-align(掌握)



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>vertical-align</title>
  <style>
    * {
      margin: 0;
      padding: 0;
    }

    div {
      width: 500px;
      background: skyblue;
      font-size: 40px;
      margin: 200px auto;
      line-height: 100px;
    }

    img {
      /* 默认情况下和一行文字的基线对齐 */
      /* 基线就是一行文字中最短文字的底部 */
    }
```



```

        vertical-align: baseline;
        vertical-align: top;
        vertical-align: bottom;
        vertical-align: text-top;
        vertical-align: text-bottom;
        /* 中线不是文字正中的地方，而是基线往上走大概四分之一的位置 */
        vertical-align: middle;
    }
</style>
</head>
<body>
<!--
    1. 什么是 vertical-align?
    设置元素的垂直对齐方式。

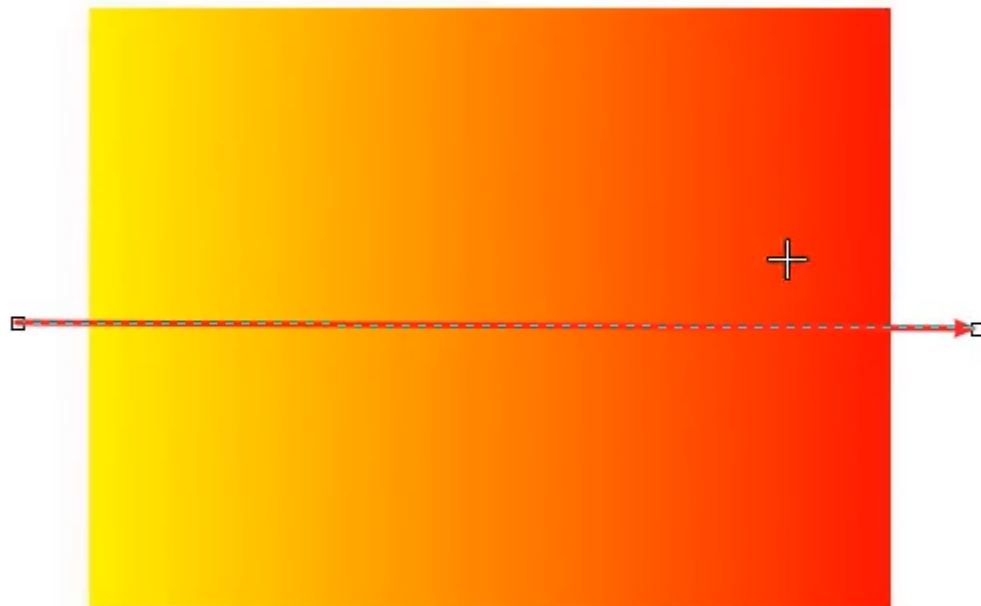
    2. vertical-align 注意点:
        text-align 是设置给需要对齐元素的父元素
        vertical-align 是设置给需要对齐的那个元素本身
        vertical-align 只针对行内元素有效
-->

<div>
    <!-- 默认情况下，图片是和这一行的基线对齐的 -->
    斯塔克工业ypx
</div>
</body>
</html>

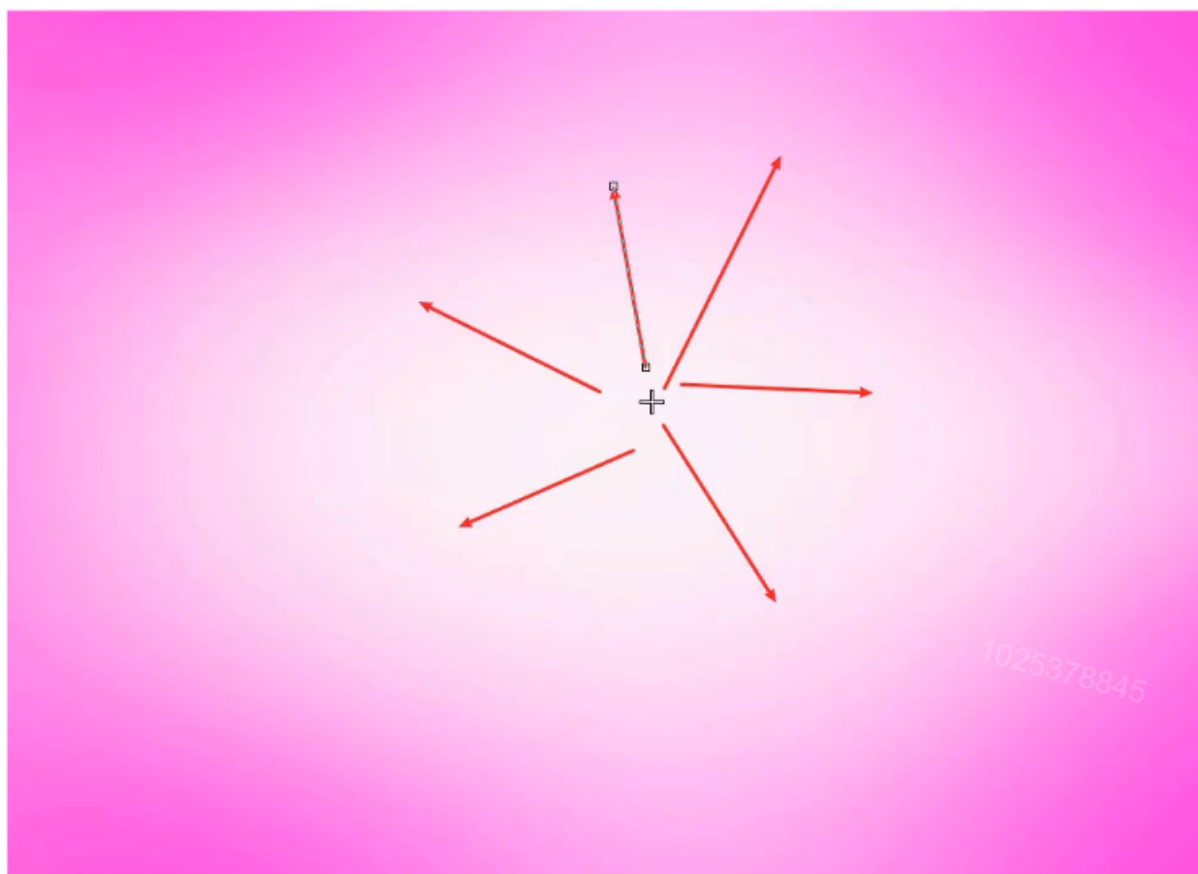
```

12. 线性渐变上(掌握)

线性渐变：从一个方向往另外一个方向渐变。



径向渐变：从指定的位置往四周进行渐变。



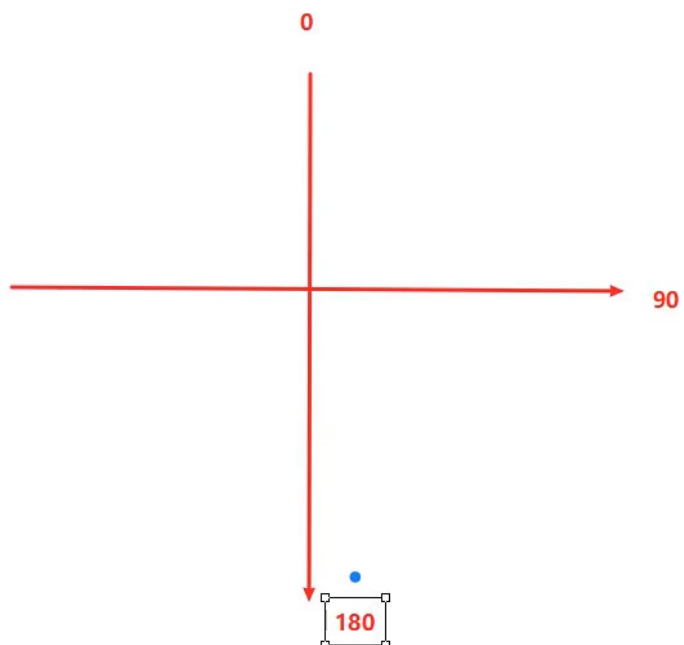
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>012-线性渐变</title>
  <style>
```

```

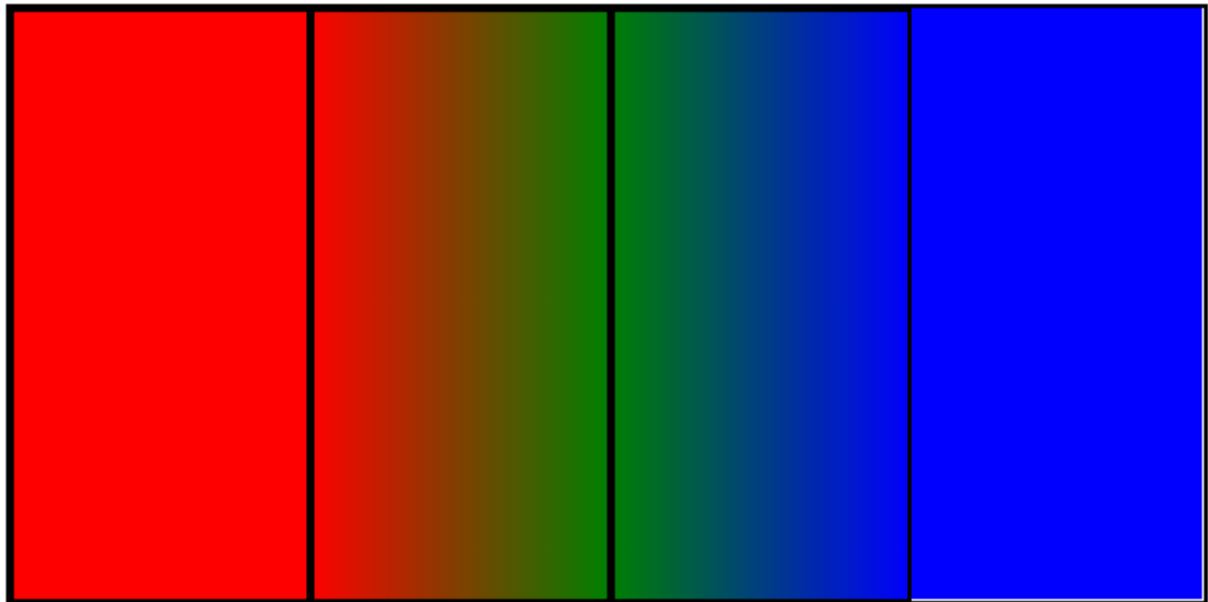
* {
  margin: 0;
  padding: 0;
}

div {
  width: 300px;
  height: 200px;
  border: 1px solid #000;
  box-sizing: border-box;
  margin: 200px auto;
  /* 默认情况下从上至下渐变 */
  background: linear-gradient(red, green);
  background: linear-gradient(to top, red, green);
  background: linear-gradient(to right, red, green);
  background: linear-gradient(to left, red, green);
  background: linear-gradient(to top right, red, green);
  background: linear-gradient(90deg, red, green);
}
</style>
</head>
<body>
  <div></div>
</body>
</html>

```



13. 线性渐变注意点(掌握)



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>012-线性渐变</title>
  <style>
    * {
      margin: 0;
      padding: 0;
    }

    div.main {
      width: 400px;
      height: 200px;
      border: 1px solid #000;
      box-sizing: border-box;
      margin: 200px auto;
      /* 注意点:
        至少需要传递两个颜色，至多没有上限
      */
      background: linear-gradient(to right, red, green, blue, yellow);

      /* 注意点
        默认情况下自动计算纯色和渐变色的范围，但是我们可以手动指定。
        手动指定的格式：颜色 范围
        只有第一个颜色后面的范围是指定纯色的范围，后面的都是指定渐变的范围。
      */
      background: linear-gradient(to right, red 100px, green 200px, blu
e 300px);
    }

    .one, .two, .three {
```

```

        width: 100px;
        height: 100%;
        border: 1px solid #000;
        box-sizing: border-box;
        float: left;
    }

</style>
</head>
<body>
    <div class="main">
        <div class="one"></div>
        <div class="two"></div>
        <div class="three"></div>
    </div>
</body>
</html>

```

14. 线性渐变练习(理解)



```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>014-线性渐变练习</title>
    <style>
        * {
            margin: 0;
            padding: 0;
        }

        body {

```

```

        background: #000;
    }

    div {
        width: 600px;
        height: 200px;
        line-height: 200px;
        font-size: 150px;
        font-weight: bold;
        border: 1px solid red;
        margin: 200px auto;
        /* color: red; */
        background: linear-gradient(-45deg, transparent 100px, white
150px, white 180px, transparent 220px);
        /* 渐变就相当于背景图片, 可以定位 */
        background-image: linear-gradient(-45deg, transparent 100px,
white 150px, white 180px, transparent 220px);
        background-position: -450px 0;
        background-repeat: no-repeat;

        /* background: skyblue; */
        color: rgba(255, 255, 255, 0.3);
        /*
        -webkit-background-clip: text;
        将背景裁剪为文字的形狀
        注意点: -webkit-background-clip 是一个私有属性, 并不是所有的浏览
器都能识别

        这种方式在业界上统称: 识别码、前缀
        -ms 代表ie内核识别码
        -moz 代表火狐firefox内核识别码
        -webkit 代表谷歌chrome、苹果safari内核识别码
        -o代表欧朋opera内核识别码
        */
        -webkit-background-clip: text;
        transition: all 5s;
    }

    div:hover {
        background-position: 450px 0;
    }
</style>
</head>
<body>
    <div>塔克工业</div>
</body>
</html>

```

15. 卡拉OK效果(理解)

伤过的心，就像玻璃碎片

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>卡拉OK</title>
  <style>
    * {
      margin: 0;
      padding: 0;
    }

    div {
      width: 900px;
      height: 78px;
      line-height: 78px;
      font-size: 75px;
      font-weight: bold;
      border: 1px solid #000;
      margin: 200px auto;

      background-color: red;
      color: transparent;
      -webkit-background-clip: text;

      background-image: url("images/text_bg.jpg");
      background-repeat: no-repeat;
      background-position: -800px 0;

      animation: loop 5s linear 0s infinite normal;
    }

    @keyframes loop {
      from {
        background-position: -800px 0;
      }
      to {
        background-position: 0 0;
      }
    }
  </style>
</head>
<body>
  <div>伤过的心，就像玻璃碎片</div>
```

```
</body>
</html>
```

16. 径向渐变(掌握)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>径向渐变</title>
  <style>
    * {
      margin: 0;
      padding: 0;
    }

    div {
      width: 400px;
      height: 400px;
      border: 1px solid #000;
      margin: 200px auto;
      /*
        线性渐变默认从上至下渐变
        径向渐变默认从中心点向四周
      */
      background: radial-gradient(red, green);

      /*
        线性渐变可以通过 to 关键字的方式修改渐变方向
        径向渐变可以通过 at 关键字的方式修改开始渐变的位置
      */
      background: radial-gradient(at top left, red, green);

      /*
        线性渐变可以通过 to deg 的方式修改渐变的方向
        径向渐变可以通过 at 位置 的方式修改开始渐变的位置
      */
      background: radial-gradient(at 200px 100px, red, green);

      /*
        线性渐变可以指定纯色和渐变的范围
        径向渐变也可以指定扩散的范围
      */
      background: radial-gradient(100px, red, green);
      background: radial-gradient(100px at 200px 100px, red, green);
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```



```
        /*
        注意点：如果需要同时指定扩散的位置和扩散的范围，那么范围必须写到 at
        的前面。
        */
    }
</style>
</head>
<body>
    <div></div>
</body>
</html>
```

完成于 2019.8.16