

Markdown语法学习

作者：胡萝卜

时间：2020-7-11

内容：

1. 学习markdown基本语法以及基本使用，并用markdown编写一篇文章；
2. 学习flex、绝对定义（absolute）、浮动(flow)布局，并用markdown来展示代码和演示截图。

Part1 Flex（弹性布局）

一、定义

Flex是Flexible Box的缩写，意为“弹性布局”，用来为盒状模型提供最大的灵活性。

二、基本概念

1. 采用Flex布局的元素，称为Flex容器（flex container），简称“容器”。它的所有子元素自动成为容器成员，称为Flex项目（flex item），简称“项目”。
2. 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main start，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cross end。
3. 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。

三、容器的属性

- flex-direction
- flex-wrap
- flex-flow
- justify-content
- align-items
- align-content

四、项目的属性

- order
- flex-grow
- flex-shrink
- flex-basis
- flex
- align-self

五、例子

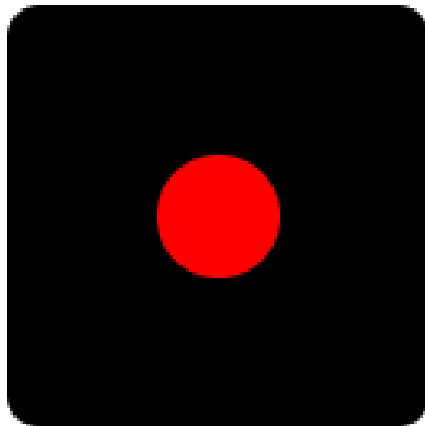
例1

```
<!DOCTYPE html>
```

```

<html>
  <head>
    <meta charset="utf-8" />
    <title>Flex Demo</title>
    <style>
      .box{
        display: flex;
        justify-content: center;
        align-items: center;
        width: 150px;
        height: 150px;
        background-color: black ;
        border-radius: 10px;
      }
      .item{
        width: 20px;
        height: 20px;
        border-radius: 50%;
        background-color: red;
      }
    </style>
  </head>
  <body>
    <div class="box">
      <span class="item"></span>
    </div>
  </body>
</html>

```



例2

```

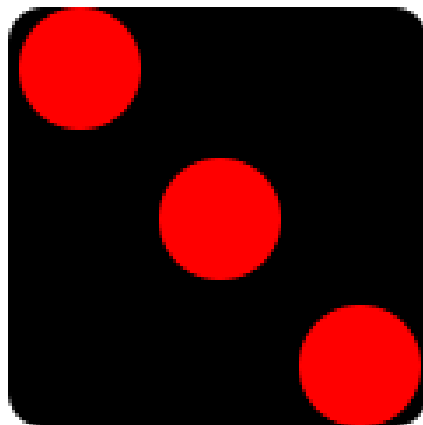
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Flex Demo</title>
    <style>
      .box{
        display: flex;
        justify-content: space-around;
        width: 120px;
        height: 120px;
        background-color: black ;
      }
    </style>
  </head>
  <body>
    <div class="box">
      <div class="item"></div>
      <div class="item"></div>
      <div class="item"></div>
    </div>
  </body>
</html>

```

```

        border-radius: 10px;
    }
    .item1{
        width: 35px;
        height: 35px;
        border-radius:50%;
        background-color: red;
    }
    .item2{
        width: 35px;
        height: 35px;
        border-radius:50%;
        background-color: red;
        align-self: center;
    }
    .item3{
        width: 35px;
        height: 35px;
        border-radius:50%;
        background-color: red;
        align-self: flex-end;
    }
}
</style>
</head>
<body>
    <div class="box">
        <span class="item1"></span>
        <span class="item2"></span>
        <span class="item3"></span>
    </div>
</body>
</html>

```



例3

```

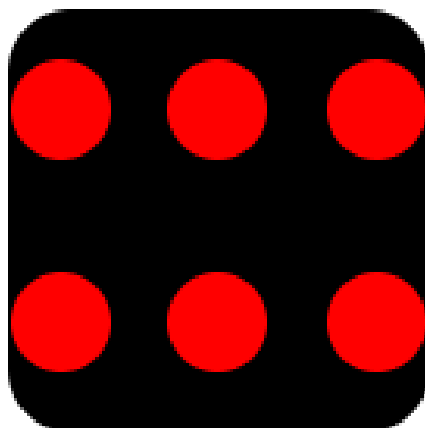
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8" />
        <title>Flex Demo</title>
        <style>
            .box{
                display: flex;

```

```

        flex-wrap: wrap;
        justify-content: space-around;
        align-content: space-around;
        width: 135px;
        height: 135px;
        background-color: black ;
        border-radius: 20px;
    }
    .item{
        width: 32px;
        height: 32px;
        border-radius: 50%;
        background-color: red;
    }
    .space{
        width: 15px;
        height: 35px;
        background-color: black;
    }
}
</style>
</head>
<body>
    <div class="box">
        <span class="item"></span>
        <span class="space"></span>
        <span class="item"></span>
        <span class="space"></span>
        <span class="item"></span>
        <span class="item"></span>
        <span class="space"></span>
        <span class="item"></span>
        <span class="space"></span>
        <span class="item"></span>
    </div>
</body>
</html>

```



例4

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">

```


$$\begin{array}{c} 1 \\ + \\ 2 \\ = \\ 3 \end{array}$$

Part2 Absolute (绝对布局)

一、定义

absolute意为绝对布局，即直接通过X,Y坐标来控制组件的位置！。

二、特点

1. 在没有设置left,top,bottom,right时，它将在没有定位之前的位置上脱离文档流，不占空间，后面的元素将会替代它的位置，并且有层级的概念。它总是在文档流的上面，除非手动设置z-index属性。
2. 如果设置了left,top,bottom,right时，它将按照设置进行定位，如果它不是在一个定位的父元素里面(除static外)，那么它的起点将在窗口的左上角，反之，它的起点将在父元素的位置上进行定位。

三、属性

- left: 距左边框距离
- right: 距右边框距离
- top: 距上边框距离
- bottom: 距下边框距离


```
<span class="item1"></span>
</div>
</body>
</html>
```



Part3 Float（浮动布局）

一、定义

float意为浮动布局，float设计之初主要用于实现文字环绕效果，后面被在页面布局方面用的比较多。

二、特点

1. 对自身影响：

- float 元素可以形成块，如 span 元素。可以让行内元素也拥有宽和高，因为块级元素具有宽高
- 浮动元素的位置尽量靠上
- 尽量靠左（float:left）或右（float:right），如果那一行满足不了浮动元素的宽度要求，则元素会往下掉

2. 对其他元素影响：

- 不影响其他块级元素的位置
- 影响其他块级元素的文本
- 上面贴非 float 元素
- 旁边贴 float 元素或者边框

3. 对父级元素影响：

- 从布局上“消失”
- 高度塌陷

三、属性

- float: left

- float: right

四、例子

例1

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>float</title>
    <style>
      .container{
        width: 304px;
        height: 150px;
      }
      .left{
        background: url("img/5.jpg") no-repeat;
        background-size: cover;
        float: left;
        height: 100%;
        width: 150px;
        border-style: solid;
        border-width: thin;
      }
      .right{
        background: url("img/6.jpg");
        background-size: cover;
        float: right;
        height: 100%;
        width: 150px;
        border-style: solid;
        border-width: thin;
      }
    </style>
  </head>
  <body>
    <div class="container">
      <span class="left"></span>
      <span class="right"></span>
    </div>
  </body>
</html>
```



例2

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>float</title>
    <style>
      .container{
        width: 600px;
        height: 150px;
        border-radius: 30px;
        border-color: cornflowerblue;
        border-style: solid;
        border-width: thin;
        background-color: pink;
      }
      .left{
        background: url("img/5.jpg") no-repeat;
        background-size: cover;
        float: left;
        height: 100%;
        width: 150px;
      }
      .right{
        background: url("img/6.jpg");
        background-size: cover;
        float: right;
        height: 100%;
        width: 150px;
      }
      .middle{
        background: url("img/7.jpg");
        background-size: cover;
        margin-left: 150px;
        margin-right: 150px;
        height: 100%;
      }
    </style>
  </head>
  <body>
```

```
<div class="container">
  <div class="left"></div>
  <div class="right"></div>
  <div class="middle"></div>
</div>
</body>
</html>
```



参考资料

[float布局](#)

[flex布局](#)

[flex布局](#)

(完)