

css第二节课

css第二节课

- 一、定位 position
- 二、flex布局
 - 1.前言
 - 2.何为Flex布局
 - 3.使用flex
- 移入效果（拓展内容）

一、定位 position

位置：

- top: 距离上面的位置
- bottom: 距离下面的位置
- left: 距离左边的位置
- right: 距离右边的位置

1. 相对定位 relative

当前位置也是自身位置 进行移动(参照物)

css代码

```
<style>
  #box {
    width: 500px;
    height: 500px;
    background-color: darkmagenta;
    /* 改变紫色盒子的位置 */
    margin-top: 100px;
    margin-left: 100px;
  }

  #box1 {
    width: 100px;
    height: 100px;
    background-color: chartreuse;
    /* 设置这个绿色的盒子为相对定位 */
    position: relative;
    /* position定位要结合位置描述属性去进行使用才会产生效果 */
    top: 50px;
    left: 50px;
  }
</style>
```

html代码

```

<!-- 参照物div -- box -->
<div id="box">
    <!-- 用于演示相对定位的 box1 -->
    <div id="box1">
        相对定位盒子
    </div>
</div>

```

2. 绝对定位 absolute

相对于浏览器的位置 而且不占网页位置

css代码

```

<style>
    #box {
        width: 500px;
        height: 500px;
        background-color: darkmagenta;
        /* 改变紫色盒子的位置 */
        margin-top: 100px;
        margin-left: 100px;
    }

    #box2{
        width: 100px;
        height: 100px;
        background-color: pink;
        /* 设置这个粉色的盒子为绝对定位 */
        position: absolute;
        /* position定位要结合位置描述属性去进行使用才会产生效果 */
        top: 150px;
        left: 250px;
    }
</style>

```

html代码

```

<!-- 参照物div -- box -->
<div id="box">
    <!-- 用于演示绝对定位的 box2 -->
    <div id="box2">
        绝对定位盒子
    </div>
</div>

```

相对定位与绝对定位对比 - 案例完整代码:

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">

```

```

<title>相对定位和绝对定位</title>
<style>
    /* css代码书写位置 */
    body {
        margin: 0px;
        padding: 0px;
    }

    #box {
        width: 500px;
        height: 500px;
        background-color: darkmagenta;
        /* 改变紫色盒子的位置 */
        margin-top: 100px;
        margin-left: 100px;
    }

    #box1 {
        width: 100px;
        height: 100px;
        background-color: chartreuse;
        /* 设置这个绿色的盒子为相对定位 */
        position: relative;
        /* position定位要结合位置描述属性去进行使用才会产生效果 */
        top: 50px;
        left: 50px;
    }

    #box2{
        width: 100px;
        height: 100px;
        background-color: pink;
        /* 设置这个粉色的盒子为绝对定位 */
        position: absolute;
        /* position定位要结合位置描述属性去进行使用才会产生效果 */
        top: 150px;
        left: 250px;
    }
</style>
</head>

<body>
    <!-- 相对定位和绝对定位
        相对定位是相对于之前盒子出生点去进行位置描述，该定位会受到周围环境影响。
        绝对定位是相对于整个浏览器窗口进行定位，它是脱离文档流的，不受周围环境影响。
    -->
    <!-- 参照物div -- box -->
    <div id="box">
        <!-- 用于演示相对定位的 box1 -->
        <div id="box1">
            相对定位盒子
        </div>
        <!-- 用于演示绝对定位的 box2 -->
        <div id="box2">
            绝对定位盒子

```

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

</html>
```

3. 固定定位 fixed

固定在一个位置

css代码

```
<style>
    #box0{
        width: 100%;
        height: 54px;
        background-color: brown;
        /* 设置这个砖红色盒子为固定定位 */
        position: fixed;
        top: 0px;
        left: 0px;
    }
    #box1{
        width: 100%;
        height: 2000px;
        background-color: darkorchid;
    }
</style>
```

html代码

```
<!-- box0 盒子是用于模拟固定不动栏的 -->
<div id="box0"></div>

<!-- box1 盒子是用于模拟网页内容的 -->
<div id="box1"></div>
```

详细案例：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>固定定位</title>
    <style>
        body{
            margin: 0px;
            padding: 0px;
        }
        #box0{
            width: 100%;
            height: 54px;
            background-color: brown;
```

```

        /* 设置这个砖红色盒子为固定定位 */
        position: fixed;
        top: 0px;
        left: 0px;
    }
    #box1{
        width: 100%;
        height: 2000px;
        background-color: darkorchid;
    }
</style>
</head>
<body>
    <!-- box0 盒子是用于模拟固定不动栏的 -->
    <div id="box0"></div>

    <!-- box1 盒子是用于模拟网页内容的 -->
    <div id="box1"></div>
</body>
</html>

```

二、flex布局

1.前言

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。目前，它已经得到了所有浏览器的支持，这意味着，现在就能很安全地使用这项功能。

2.何为Flex布局

“弹性布局”，用来为盒状模型提供最大的灵活性，任何一个容器都可以指定为Flex布局。

3.使用flex

一、**创建Flex容器**：使用Flex布局，首先需要创建一个Flex容器，即一个元素的父级元素。在父级元素上应用 `display: flex` 样式，以将其设置为Flex容器。例如：

```

<style>
    .box1{
        display: flex;
    }
</style>

<div class="box1">
    <div class="box1_01">Flex项目1</div>
    <div class="box1_01">Flex项目2</div>
</div>

```

二、**定义Flex项目**：将要放置在Flex容器内的元素称为Flex项目。这些项目将根据Flex容器的规则进行布局。默认情况下，Flex项目会水平排列。可以通过在Flex容器上应用其他CSS属性来更改Flex项目的布局方式，如 `flex-direction`、`justify-content` 和 `align-items` 等。例如：

案例css代码

```

<style>

```

```

/*
flex布局 - 容器的一些属性：
    flex-direction（默认值：row）用于定义 Flex 项目的排列方向。
    可选值包括：
        row：水平方向（默认值），从左到右排列。
        row-reverse：水平方向，从右到左排列。
        column：垂直方向，从上到下排列。
        column-reverse：垂直方向，从下到上排列。

justify-content（默认值：flex-start）用于定义 Flex 项目在主轴上的对齐方式。
    可选值包括：
        flex-start：在主轴起始位置对齐。
        flex-end：在主轴末尾位置对齐。
        center：在主轴中心位置对齐。
        space-between：在主轴上均匀分布，首个项目放在起始位置，末尾项目放在末尾位置。

        space-around：在主轴上均匀分布，项目之间和两侧均有空间。
        space-evenly：在主轴上均匀分布，包括首个项目和末尾项目两侧的空间。

align-items（默认值：stretch）用于定义 Flex 项目在侧轴上的对齐方式。
    可选值包括：
        stretch：默认值，如果项目未设置固定的侧轴尺寸，则会拉伸以填满容器。
        flex-start：在侧轴起始位置对齐。
        flex-end：在侧轴末尾位置对齐。
        center：在侧轴中心位置对齐。
        baseline：以基线对齐，适用于文本等。
*/
.box1{
    width: 100%;
    height: 100vh;
    display: flex;
    flex-direction: row; /* 水平排列项目（默认值） */
    justify-content: center; /* 水平居中对齐项目 */
    align-items: center; /* 垂直居中对齐项目 */
}
.box1_01{
    padding: 20px;
    border: 1px solid #000;
}
</style>

```

案例html代码

```

<!-- 先创建一个盒子，作为flex容器 -->
<div class="box1">
    <div class="box1_01">Flex项目1</div>
    <div class="box1_01">Flex项目2</div>
</div>

```

移入效果（拓展内容）

hover

给当前属性加上hover

鼠标移入.box2的内容 背景颜色 和文字颜色 都会改

css代码

```
.box2:hover{  
    background-color: rgb(85, 219, 61);  
    color: #fff;  
}
```

html代码

```
<div class="box2">1</div>
```