

## 一、尺寸与颜色单位

1. 尺寸单位
2. 颜色单位

## 2. CSS 盒模型-做布局

1. 内容尺寸
2. 边框
  1. 边框实现
  2. 单边框设置
  3. 网页三角标制作
  4. 圆角边框
  5. 轮廓线
  6. 盒阴影
3. 内边距
4. 外边距
5. 元素最终尺寸的计算

# 一、尺寸与颜色单位

## 1. 尺寸单位

- px 像素单位
- % 百分比，参照父元素对应属性的值进行计算 一般是适配移动端。
- em 字体尺寸单位，参照父元素的字体大小计算，1em=16px 适用于移动端
- rem字体尺寸单位,参照根元素的字体大小计算，1rem=16px 适用于移动端

## 2. 颜色单位

- 英文单词：red，green，blue
- rgb(r,g,b) 使用三原色表示，每种颜色取值0~255
- rgba(r,g,b,alpha) 三原色每种取值0~255，alpha取值0（透明）~1（不透明）
- 十六进制表示：以#为前缀，分为长十六进制和短十六进制。
  - 长十六进制：每两位为一组，代表一种三原色；每位的取值范围0~9，a~f  
例：red rgb(255,0,0) #ff0000
  - 短十六进制：由3位组成，每一位代表一种三原色，浏览器会自动对每一位进行重复扩充，仍然按照长十六进制解析  
例：#000 #fff #f00

```
#d1{
    width:50%;
    height:50%;
    /* 短十六进制 */
    background-color:#0ff;
    /* 不透明度 */
    opacity:0.4;
    font-size:20px;
}
```

## 2. CSS 盒模型-做布局

### 1. 内容尺寸

- 一般情况下，为元素设置width/height，指定的是内容框的大小
- 内容溢出：内容超出元素的尺寸范围，称为溢出。默认情况下溢出部分仍然可见，可以使用overflow调整溢出部分的显示,取值如下：

取值	作用
visible	默认值，溢出部分可见
hidden	溢出部分隐藏
scroll	强制在水平和垂直方向添加滚动条
auto	自动在溢出方向添加可用滚动条

```
div{
  width:400px;
  height: 100px;
  background-color:red;
  /* overflow对溢出内容的显示效果，默认显示溢出内容 */
  overflow: auto;
}
```

## 2. 边框

### 1. 边框实现

语法：

```
border:width style color;
```

边框样式为必填项，分为：

```
div{
  width: 200px;
  height: 200px;
  /* 实线 */
  border:5px solid #000;
  /* 点线 */
  border:5px dotted #000;
  /* 虚线 */
  border:5px dashed #000;
  /* 双线 */
  border:5px double #000;
}
```

样式取值	含义
solid	实线边框
dotted	点线边框
dashed	虚线边框
double	双线边框

## 2. 单边框设置

分别设置某一方向的边框，取值：width style color;

```
div{
    width: 200px;
    height: 200px;
    border-top:5px solid #f00;
    border-top-left-radius: 15px;
    border-left:5px solid #f0f;
    border-right:5px solid #f00;
    border-bottom-color:transparent;
}
```

属性	作用
border-top	设置上边框
border-bottom	设置下边框
border-left	设置左边框
border-right	设置右边框

## 3. 网页三角标制作

1. 元素设置宽高为0
2. 统一设置四个方向透明边框
3. 调整某个方向边框可见色

```
div{
    width: 0;
    height: 0;
    border: 50px solid transparent;
    border-top-color: #000;
}
```

## b4. 圆角边框

1. 属性：border-radius 指定圆角半径
2. 取值：像素值或百分比
3. 取值规律：顺时针取值

```
<style>
  div{
    width:200px;
    height:200px;
    border:1px solid lightblue;
    border-radius:15px 60px ;
    background-color: yellow;
  }
</style>
```

一个值	表示统一设置上右下左
四个值	表示分别设置上右下左
两个值	表示分别设置上下 左右
三个值	表示分别设置上右下，左右保持一致

## 5. 轮廓线

1. 属性：outline
2. 取值：width style color
3. 区别：边框实际占位，轮廓不占位
4. 特殊：取**none**可以取消文本输入框默认轮廓线

## 6. 盒阴影

1. 属性：box-shadow
2. 取值：offsetX offsetY blur (spread) color;
3. 使用：  
不管是浏览器窗口还是元素自身都可以构建坐标系，统一以左上角为原点，向右向下为X轴和Y轴的正方向

offsetX	取像素值，阴影的水平偏移距离
offsetY	取像素值，阴影的垂直偏移距离
blur	取像素值，表示阴影的模糊程度，值越大越模糊
spread	选填，取像素值，阴影是否需要延伸
color	设置阴影颜色，默认为黑色

## 3. 内边距

1. 属性：padding
2. 作用：调整元素内容框与边框之间的距离
3. 取值：

20px;	一个值表示统一设置上右下左
20px 30px;	两个值表示分别设置(上下) (左右)
20px 30px 40px;	三个值表示分别设置上右下, 左右保持一致
20px 30px 40px 50px;	表示分别设置上右下左

```
div{
    width:200px;
    height:200px;
    border:1px solid red ;
    padding:10px 20px;
}
```

#### 4. 单方向内边距,只能取一个值 :

```
padding-top
padding-right
padding-bottom
padding-left
```

```
div{
    width:50px;
    height:60px;
    border:2px solid gray;
    /* 垂直居中 */
    line-height:60px;
    /* 水平居中 */
    text-align: center;
    font-size:16px;
}

div{
    width:32px;
    border:1px solid red;
    font-size:16px;
    /* 让元素水平垂直居中 */
    padding:22px 9px;
}
```

## 4. 外边距

1. 属性 : margin
2. 作用 : 调整元素与元素之间的距离
3. 特殊 :
  - 1 ) margin:0; 取消默认外边距
  - 2 ) margin:0 auto; 左右自动外边距, 实现元素在父元素范围内水平居中
  - 3 ) margin:-10px; 元素位置的微调
4. 单方向外边距 : 只取一个值

```
margin-top
margin-right
margin-bottom
margin-left
```

## 5. 外边距合并：

### 1) 垂直方向

#### 1. 子元素的margin-top作用于父元素上

解决：

为父元素添加顶部边框：

或为父元素设置padding-top:0.1px；

2. 元素之间同时设置垂直方向的外边距，最终取较大的值

### 2) 水平方向

块元素对盒模型相关属性（width,height,padding,border,margin）完全支持；

行内元素对盒模型相关属性不完全支持，不支持width/height,不支持上下边距

行内元素水平方向上的外边距会叠加显示

带有默认边距的元素：

```
body,h1,h2,h3,h4,h5,h6,p,u1,o1{
    margin:0;
    padding:0;
    list-style:none;
}
```

## 5. 元素最终尺寸的计算

盒模型相关的属性会影响元素在文档中的实际占位，进而影响布局

属性：box-sizing

取值：content-box/border-box

### 1) 标准盒模型计算：各个属性值累加得到最终尺寸

box-sizing:content-box; (默认设置)

元素设置width/height指定的是内容框的大小

最终尺寸 = width/height+padding+border+margin

### 2) 特殊盒模型计算（按钮元素）：

box-sizing:border-box;

元素设置width/height指定的是包含边框在内的区域大小

最终尺寸 = width/height+margin

作业：

在横向导航栏的基础上，调整导航项的边距

#### 1. 整体导航栏水平居中

#### 2. 导航项之间10px的外边距

如下这两种,效果一样.

```
div{
    width:200px;
    height:200px;
    border:5px solid red;
    background-color: orange;
    padding:10px;
    margin:10px;
}
div{
    width:180px;
    height:180px;
    /* border-box将内边框和边框一同计算在长宽中,自调整内容框的大小 */
    box-sizing:border-box;
    border:5px solid red;
    padding:10px;
```

```
margin:10px;
}
```

练习:

### 1. 做黑色下角标:

```
<!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>
    span{
      color:#666;
    }
    div{
      width:0px;
      width:0px;
      border:8px solid transparent;
      border-top-color:#000;
      display:inline-block;
      margin-bottom:-5px;
    }
  </style>
</head>
<body>
  <span>上海</span>
  <div></div>
</body>
</html>
```

### 2.PLUS会员

```
<!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>
    div{
      width:180px;
      height:62px;
      border:31px;
      border-radius: 31px;
      background-color: #666;
      color:rgb(228, 228, 113);
      font-size:30px;
      text-align:center;
    }
  </style>
</head>
<body>
  <div>PLUS会员</div>
</body>
</html>
```

```

        line-height: 62px;
        box-sizing: border-box;
    }
    div:hover{
        background-color: red;
        color: white;
        transition: all 0.25s;
        cursor: pointer;
    }
</style>
</head>
<body>
    <div>PLUS会员</div>
</body>
</html>

```

### 3. 难度: 全部 初级 中级 高级

```

<!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>
        div{
            width: 60px;
            font-size: 16px;
            text-align: center;
            display: inline-block;
            margin: 20px;
        }
        table{
            border: 1px solid rgb(207, 201, 201);
        }
        .all:hover{
            background-color: red;
            color: white;
            transition: all 0.3s;
            cursor: pointer;
        }
        #d1{
            color: rgb(99, 93, 93);
        }
    </style>
</head>
<body>
    <table>
        <tr>
            <td>
                <div id="d1">难度:</div>
            </td>
            <td>
                <div class="all">全部</div>
            </td>
            <td>

```



```
        <div class="all">初级</div>
    </td>
    <td>
        <div class="all">中级</div>
    </td>
    <td>
        <div class="all">高级</div>
    </td>
</table>

</body>
</html>
```