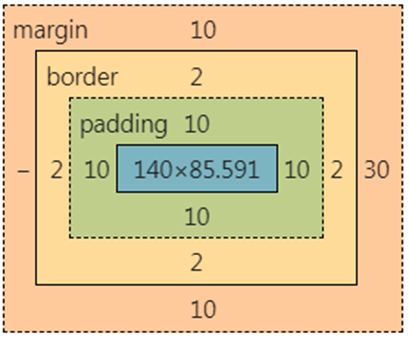
# DAY06 CSS

# 1 CSS盒子模型(Box Model)

### 默认的盒子模型（内容盒子）

一个HTML元素，需要占用页面的空间。

内容、边框、内容与边框之间的距离、不同元素直接的距离都需要占用页面空间

所以我们需要了解具体的计算方式

|  |
| --- |
| content: 内容区域  padding: 内间距—元素内容与边框之间的距离  border: 元素的边框  margin: 外间距—元素与元素之间的距离 |
| 涉及到四个方向:left左 right右 top上 bottom下 |
| **一个元素在水平方向上占用的总空间：**  左外间距**+**左边框**+**左内间距**+**内容宽度**+**右内间距**+**右边框**+**右外间距  margin-left**+**border-left**+**padding-left**+**width**+**padding-right**+**border-right**+**margin-right |
| **一个元素在垂直方向上占用的总空间：**  上外间距**+**上边框**+**上内间距**+**内容高度**+**下内间距**+**下边框**+**下外间距  margin-top**+**border-top**+**padding-top**+**height**+**padding-bottom**+**border-bottom**+**margin-bottom |

### 边框盒子模型(怪异盒子)

|  |
| --- |
| 之前我们学习的是内容盒子box-sizing:content-box;  内容盒子会把元素本身的大小 内间距 边框 外间距 都会增大元素在页面布局中所占据的总空间  如果想增加距离且不增加元素占据的页面布局空间，推荐使用边框盒子：  box-sizing:border-box;  注意：边框盒子需要指定width |
| **一个元素在水平方向上占用的总空间：**  左外间距 **+** 宽度(内容+内间距+边框) **+**右外间距  margin-left **+**width(content+padding+border) **+**margin-right |
| **一个元素在垂直方向上占用的总空间：**  上外间距 **+** 宽度(内容+内间距+边框) **+**下外间距  margin-top **+**width(content+padding+border) **+**margin-bottom |

# 2 边框

### 1. 边框的不同写法

|  |
| --- |
| **对边框进行综合设置(最常用)**  border:1px solid #000; 粗细 线型 颜色 |
| **设置某一个方向的边框：**  border-top: 1px solid #000; 设置上边框的粗细 线型 颜色  border-top-color:#0f0; 设置上边框的颜色 |
| **设置所有边框的单独一个属性:**  颜色: border-color：red;  宽度: border-width: 20px;  线型: border-style：solid;  (solid单实线 dashed虚线 dotted点状线 double双实线) |
| **如果有的元素不需要设置边框，可以去掉**  border:none; **或** border:0; |

### 2. 边框的巧妙用法

|  |
| --- |
| 1. 需要一个没有内容，且宽高为0的div  2. 给此div设置一个大大的四条边都为透明色的边框  3. 给需要显示三角形的边设置颜色，作为要显示的三角形  .c1 {  width: 0; /\* 因为块级元素都是默认宽度父元素的100% 所以要给0\*/  border: 50px solid transparent; /\* 四个边都设置为透明，边框宽度决定三角的大小 \*/  border-bottom-color: red; /\* 单独设置需要的边角有颜色即可显示 \*/  }  <h4>向上三角</h4>  <div class="c1"></div> |

# 3 内间距

元素内容与边框之间的距离，叫做内间距(padding)

### 四个方向的内间距

|  |
| --- |
| 上内间距padding-top: 10px;  右内间距padding-right: 10px;  下内间距padding-bottom: 10px;  左内间距padding-left: 10px; |

### 内间距的简写形式

|  |
| --- |
| padding:10px; 四个方向的内间距都是10  padding:10px 20px; 上下10 左右20  padding:10px 20px 30px; 上10 左右20 下30  padding:10px 20px 30px 40px; 上10 右20 下30 左40  记忆诀窍：永远都是上右下左4个值，有值就顺时针赋值，没值就找对门 |

### 不同显示模式下的内间距

|  |
| --- |
| 1. 块级与行内块元素的四个方向的内间距均生效 2. 行内元素的左右内间距正常生效，但上下内间距只会“视觉生效”并不占用布局空间，还可能会产生布局的堆叠，所以行内元素尽量不要使用上下内间距 3. img是特殊的行内元素，它的四个方向的内间距均生效 |

### 使用场景

|  |
| --- |
| 让元素内部距离外层父级元素的边有点距离，显得好看，主要是为了美化  比如：输入框的文字不要贴边显示；文本卡片内容不要贴边显示 |

# 4 外间距

元素边框外部与另外的元素之间的距离称为外间距(margin)

外间距主要用于控制当前元素在四个方向上与其他元素的距离

外间距不属于元素本身的大小，我们还可以利用外间距让元素发生移动

### 1. 四个方向的外间距

|  |
| --- |
| margin-top:10px; 上外间距 元素顶部与其他元素的距离  margin-right:10px; 右外间距 元素右侧与其他元素的距离  margin-bottom:10px; 下外间距 元素底部与其他元素的距离  margin-left:10px; 左外间距，元素左侧与其他元素的距离 |

### 2. 外间距的简写形式

|  |
| --- |
| margin:10px; 上右下左四个方向都是10  margin:10px 20px; 上下:10 左右:20  margin:10px 20px 30px; 上:10 左右:20 下:30  margin:10px 20px 30px 40px; 上:10 右:20 下:30 左:40  **记忆诀窍：**  永远是上右下左4个值，有值就顺时针赋值，没值就找对门 |

### 外间距的特点

|  |
| --- |
| 1. 外间距产生的四个方向上的空间并不在元素内部，而是在元素外部，不带任何背景颜色 2. 块级与行内块元素四个方向上的外间距均生效 3. 行内元素的左右外间距可以正常生效，上下外间距不生效 |

### 外间距的取值

|  |
| --- |
| 正值：增加距离  0值: 去掉间距  负值：减少距离  auto:自动计算距离，流式布局中块级元素左右方向的外间距计算成等值，实现水平居中效果  注意：margin:0 auto;只针对块级元素中的块级子元素水平居中，文字和行内元素不生效 |

### 外间距的重叠【重难点】

|  |
| --- |
| 在文档流布局中：   1. 块级元素在垂直方向上相邻的外间距会发生合并现象，二者取最大值显示 2. 父元素中的首/尾子元素都有和父元素“重叠的边”，首/尾子元素的上/下margin会超出父元素的范围,解决方案：   方案1：给父元素加边框  方案2：给父元素加内间距 |

# 5背景颜色和背景图片

### 1. 背景颜色

|  |
| --- |
| background-color:#fff; 背景颜色会在元素的最底层  颜色可以使用任何的色值(十六进制 rgb rgba 单词等) |

### 2. 背景图片的插入

|  |
| --- |
| background-image:url(路径);  **注意背景图和img标签的图片插入是有区别的！**  1. img标签插入的图片在默认情况下直接就是图片本身的大小  2. background-image:url(路径);方式并不能直接显示图片的，必须给当前元素设置宽度和高度才能显示出背景图片 |

### 3. 背景图的重复(平铺)

|  |
| --- |
| **图片被平铺到元素中会出现两种情况：**  1. 图片范围大于元素范围，显示不全  2. 图片范围小于元素范围，会出现重复，哪个方向有空余空间，就向哪个方向重复 |
| **属性与值：**  background-repeat: repeat; 默认值，X-Y轴都重复  background-repeat: repeat-x; 仅在X轴(水平方向)重复  background-repeat: repeat-y; 仅在Y轴(垂直方向)重复  background-repeat: no-repeat; 背景图仅显示一次，不重复 |

### 4. 背景图定位

|  |
| --- |
| background-position:left top; 左上，默认值，先写左右，再写上下  X轴(水平方向)的值：left左 center中 right右  Y轴(竖直方向)的值：top上 center中 bottom下  center center表示水平与竖直方向均居中，两个center可以简写成一个center  还可以写像素值，原点(0px 0px)就是左上角，正值向右/向下移动；负值向左/向上移动 |

### 5. 精灵图

|  |
| --- |
| CSS Sprite 直译为CSS精灵图，网页图片的一种处理方式  UI会把多个小图标整合到一张图片中，再利用背景图定位，选取到需要的小图标位置展示出来。  优点：减少了向服务器的请求次数，进行了CSS优化 |
| 阿里图标库：<https://iconfont.cn/>  里面有很多的矢量图，还可以自定义尺寸，免费使用  第一次 使用需要注册，手机号就可以注册 |

### 6. 背景渐变

|  |
| --- |
| 背景颜色可以设置渐变的效果，使用最多的就是线性渐变  **1. 颜色设定（默认颜色从上向下排列）**  background-image: linear-gradient(颜色1，颜色2，颜色3…)  **2. 角度设定（角度单位deg）**  background-image: linear-gradient(角度值，颜色1，颜色2…)  **3. 渐变色范围设定：**  background-image: linear-gradient(角度，颜色1 起始位置 结束位置，颜色2 起始位置 结束位置，…)  范围的设定需要写在每个颜色色值的后面，值可以使用长度单位px或者百分比% |

# 拓展

### 面试题：如果两张图片中间想要留一点空白，可以有哪些实现方案？哪种好？为什么？

|  |
| --- |
| 1. 添加透明边框——不适合于图片本身需要边框的情形 2. 添加外间距——可以但是不好，因为margin会增大元素所占据的页面布局空间 3. 添加内间距——要配合边框盒子使用，是最佳方案！ |

# C:\Users\TEACHER\Desktop\WEB2212\DAY06\PIC\20230103114038.png作业

详见作业文件夹