

第五天

display

作用：块级元素、行内元素以及行内块级元素之间的相互转行

display:block; 块级元素

特点：

- 独占一行(从上到下排列)
- 可以设置css盒子模型的所有属性
(width/height/border/padding/margin)
- 可以嵌套其他元素
 - p,dt不能嵌套块级元素
- 在不设置宽高时，宽是继承父级的宽度，高是由内容决定的
- 永远在父级元素的左上角排布的

display:inline; 行内

特点：

- 在一行显示(从左到右排布)
- 不能设置宽高(css盒子模型的属性)

- 在不设置宽高时，宽高是由内容决定的
- 在编辑代码时，行内元素出现回车或者换行时，会默认有间隙
 - 解决方法：body{font-size:0;}
- 不可以嵌套块级元素
- 基线对齐

display:inline-block; 行内块

特点：

- 在一行显示(从左到右排布)
- 可以设置宽高(css盒子模型的属性)
- 在不设置宽高时，宽高是由内容决定的
- 在编辑代码时，行内元素出现回车或者换行时，会默认有间隙
 - 解决方法：body{font-size:0;}
- 不可以嵌套块级元素
- 基线对齐

display:none; 隐藏、消失

如果设置了display:none;，这个css属性，会将这个元素在页面中完全消失。

如果想再让这个消失的元素显示在页面中，给这个元素设置display:block;就会重新显示在页面中

vertical-align 改变行内元素和行内块级元素的对齐方式

值	描述
top	所有同级元素，找高度(最高)的顶部进行对齐
middle	所有同级元素，找高度(最高)的中部进行对齐
bottom	所有同级元素，找高度(最高)的底部进行对齐
px像素、%百分比	

改变元素的对齐方式，要给所有的元素设置这个CSS属性(vertical-align)
vertical-align常用在**背景图片(雪碧图)和文字**结合的时候

用在以下结构中：

```
<div>  
    <i></i>/**背景图片、雪碧图**/  
    文字  
</div>
```

盒子模型

在html页面中，每一个元素相当于一个盒子，盒子和盒子之间有距离(margin外边距)，盒子本身有厚度(border边框)，盒子里面的内容(width/height),盒子与内容之间有个白色泡沫(padding内边距)

width 宽度

height 高度

border 边框

padding 内边距

margin 外边距

margin 外边距

margin:1px 2px 3px 4px; 上 右 下 左

margin:1px 2px 3px; 上 左右 下

margin:1px 2px; 上下 左右

margin:1px;设置了四个方向的值

margin-top:10px; 上外边距

margin-right:10px; 右外边距

margin-bottom:10px; 下外边距

margin-left:10px; 左外边距

padding 内边距

padding:1px 2px 3px 4px; 上 右 下 左

padding:1px 2px 3px; 上 左右 下

padding:1px 2px;上下 左右

padding:1px;设置四个方向的值

padding-top:1px; 上内边距

padding-right:1px; 右内边距

padding-bottom:1px; 下内边距

padding-left:1px; 左内边距

盒子和盒子之间的距离、块级元素与块级元素之间的距离就用margin

盒子内的距离、行内元素与行内元素之间的距离就用padding

border 边框

border:1px solid red; **常用**

border-top:1px solid red; 上边框 **常用**

border-bottom:1px solid red; 下边框 **常用**

border-width:1px; 边框的宽度

border-width:1px 2px 3px 4px; 上 右 下 左

border-width:1px 2px 3px;上 左右 下

border-width:1px 2px; 上下 左右

border-width:1px;设置了四个方向的值

border-style:solid; 边框的样式

border-style:solid dashed dashed dashed; 上右下左

border-color:red; 边框的颜色

border-color:red yellow blue green; 上右下左

border制作实心三角形

border-width是控制三角形的大小

给这个元素设置width:0;

```
<style>
.div1{
    width:0;
    border-width: 20px;
    border-style: solid;
    border-color: red transparent trans
parent transparent;
}
</style>
<div class="div1"></div>
```

CSS盒子模型的计算公式

AllWidth = width + 左右padding + 左右border

240 = 200px 10+10 10+10

ALLHeight = height+上下padding+上下border

margin兼容问题

1. 一个元素设置margin-bottom, 另一个元素设置margin-top, 这两个值不会相加, 会取最大值(这个元素是平级、同级关系的时候)

```
<style>
div{
    width:100px;
    height: 100px;
}
.div1{
    margin-bottom: 30px;
    background-color: red;
}
.div2{
    margin-top:50px;
    background-color: yellow;
}
</style>
<div class="div1"></div>
<div class="div2"></div>
```

2. margin-top值传递问题(父子包含关系)

当父级元素没有设置padding-top和border-top值时, 给子元素设置了margin-top值时, 这个值就会传递给父级元素

如何解决:

- 给父级元素设置一个CSS属性-overflow: hidden; 超出隐藏
- 不要给子级元素设置margin-top, 给父级元素设置padding-top **常用**

注意：给父级元素设置padding-top值时，父级元素会在原有高度之上增加，那时，一定要减去增加的那个高度

overflow 内容溢出的显示方式

- `overflow: hidden;` 超出元素高度就隐藏掉—**常用**
- `overflow: auto;` 浏览器自动判断内容是否超出元素的高度，如果超出就会出现滚动条，没有超出，则不显示滚动条
- `overflow: scroll;` 不管元素的宽高是否超出，都出滚动条

CSS属性之background 背景

- `background-color:red;` 背景颜色
- `background-image: url("1.jpg")` 设置背景图片
- `background-repeat:no-repeat;` 改变背景图片的平铺方式
 - 默认的值：repeat 平铺
 - **no-repeat** 不平铺
 - repeat-x x轴平铺

- repeat-y y 轴平铺
- `background-position:20px 30px;` 设置背景图片的位置
 - 设置单位：px、%、英文字母
 - 有两个值：第一个值代表x轴或者水平方向、第二个值代表y轴或者垂直方向
 - 如何让背景图片与盒子的中心点重合：
`background-position:center center;`
- `background-size:100% 100%;` 设置背景图片的大小
 - 设置值的单位：px、%
 - 有两个值：第一个值代表x轴、第二个值代表y轴
 - 当两个值都设置为100%时，根据元素的盒子有多大，背景图片就有多大

```
background: pink url("favicon.ico") no-repeat 20px 30px;
```

`background`：颜色 图片路径 背景图片是否平铺 改变背景图片位置

如果想要改变背景图片的大小，只能重新另起一行设置这个元素的样式

要想插入背景图片，元素必须有宽高