# CSS基础入门

第6天课堂笔记（本课程共8天）

# 目录

[CSS基础入门 1](#_Toc30318)

[目录 2](#_Toc2990)

[一、 复习 3](#_Toc1749)

[二、 定位 3](#_Toc28629)

[1、 相对定位 4](#_Toc17383)

[2、 绝对定位 5](#_Toc32557)

[①不针对祖先元素的定位参考点 6](#_Toc14551)

[②祖先元素有定位的参考点 7](#_Toc21395)

[③绝对定位的顶点。 8](#_Toc19876)

[④利用绝对定位来做压盖效果。 8](#_Toc23640)

[⑤绝对定位元素水平居中 8](#_Toc26231)

[3、固定定位 9](#_Toc24212)

[4、 z-index属性 10](#_Toc18239)

[默认压盖顺序 10](#_Toc7403)

[自定义压盖顺序 10](#_Toc9424)

# 复习

1、a标签

根据用户的行为，a有四个状态：  
 a:link

a:visited

a:hover

a:active

:link,都叫做伪类。

顺序必须遵循爱恨准则。

四个状态可以不全写，还可以合写。

默认：把盒子的属性写在a标签选择器里，文字属性有两个不能从父级继承，color、text-decoration。

实际工作中：清除默认样式的时候，a的样式也要重置。重置的样式一般找a出现最多的样式。

|  |
| --- |
| 1. body{ 2. color: #666; 3. font-size: 20px; 4. font-family: "Microsoft Yahei"; 5. } 6. a{ 7. font-size: 12px; 8. color: #999; 9. text-decoration: none; 10. } 11. a:hover{ 12. color: #0ff; 13. } |

1. background背景

五个单独的属性：

|  |
| --- |
| 1. background-color: lightblue; 2. background-image: url(images/1.jpg); 3. background-repeat: no-repeat; 4. background-position: center top; 5. background-attachment: fixed; |

背景属性合写：

|  |
| --- |
| 1. background: lightblue url(images/1.jpg) no-repeat center top fixed; |

# 定位

css布局：盒模型、浮动、定位这三个是最终要的属性。

辅助：文字、颜色、a、bg，都是可以辅助让我们的网站更加美观。

定位：相对定位、绝对定位、固定定位。

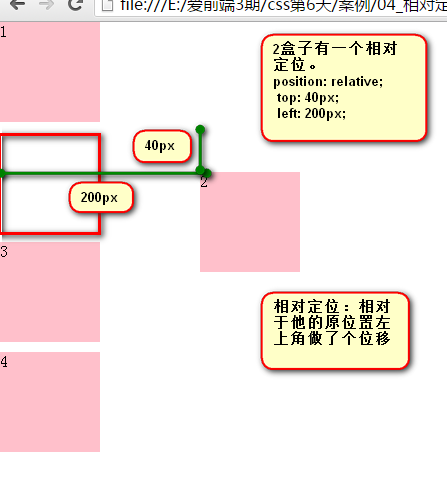
脱标方法：浮动、绝对定位、固定定位。

## 相对定位

相对于元素本身的位置进行调整。

position：定位。

|  |
| --- |
| 1. .box{ 2. position: relative; 3. top: 40px; 4. left: 200px; 5. } |



相对定位可以有四个方向的位置调整。

|  |
| --- |
| 1. top: 30px; 正数，向下移30像素 2. bottom: 40px; 正数，向上移动40像素 3. left: 10px; 正数，向右移动10像素 4. right: 20px; 正数，向左移动10像素 |

四个值可以随机组合。左右选一个，上下选一个。

|  |
| --- |
| 1. .box1{ //参考点是本身盒子的左上顶点 2. top: 20px; 3. left: 20px; 4. } 5. .box2{ //参考点是本身盒子的右上顶点 6. top: 20px; 7. right: 20px; 8. } 9. .box3{ //参考点是本身盒子的左下顶点 10. bottom: 20px; 11. left: 20px; 12. } 13. .box4{ //参考点是本身盒子的右下顶点 14. bottom: 20px; 15. right: 20px; 16. } |

负值：是往同方向动。

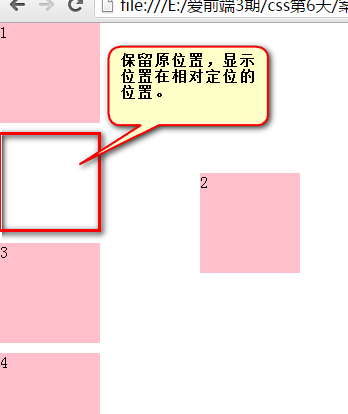
|  |
| --- |
| 1. .box1{ //参考点是本身盒子的右下顶点 2. bottom: -20px; 3. right: -20px; 4. } |

相当于：

|  |
| --- |
| 1. .box1{ //参考点是本身盒子的左上顶点 2. top: 20px; 3. left: 20px; 4. } |

相对定位不会让元素脱离标准流。原位置保留，显示位置在新的相对定位位置。

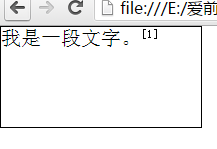
形影分离。



相对定位对我们的布局没有太大的作用，用途：

①微调。

|  |
| --- |
| 1. div p span{ 2. font-size: 12px; 3. position:relative; 4. left: 0; 5. top: -7px; 6. } |



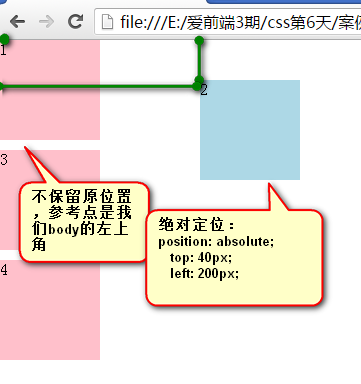
②相对定位的盒子会作为绝对定位的参考盒子。“子绝父相”。

应用：



## 绝对定位

|  |
| --- |
| position：absolute； |



|  |
| --- |
| 1. position: absolute; 2. top: 40px; 3. left: 200px; |

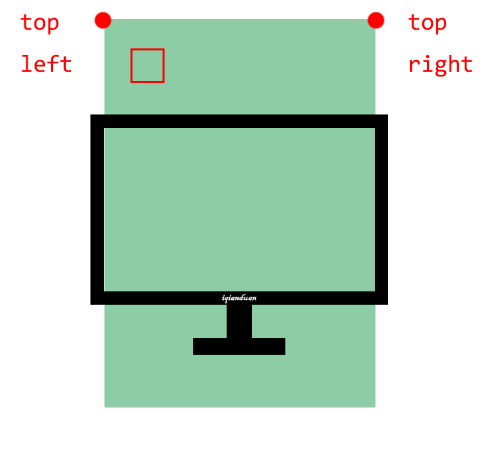
参考点是我们的body的左上角。

绝对定位元素脱离标准流，把标准流的位置让给其他的元素。

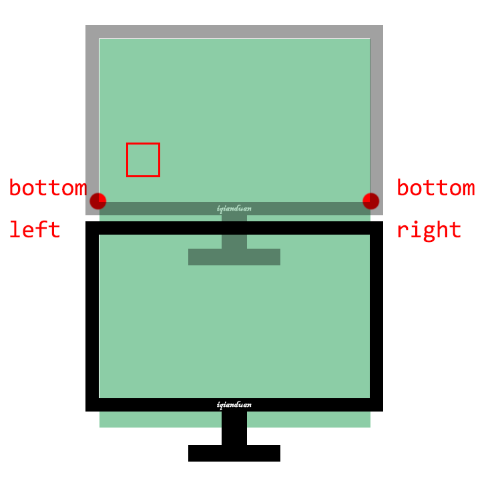
绝对定位也可以有四个方向的位移：left、right、top、bottom。

### ①不针对祖先元素的定位参考点

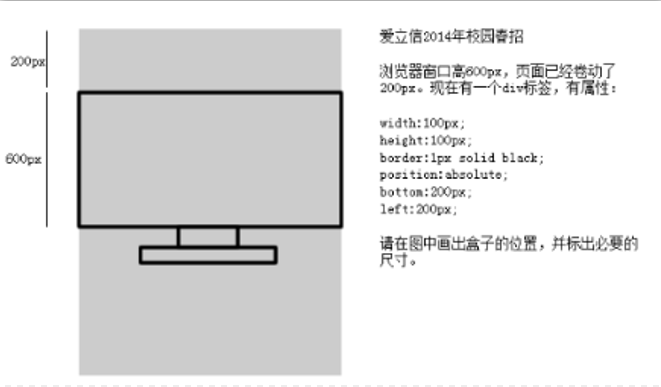
有top参与的绝对定位，参考点是我们页面（body）的左上角或右上角。



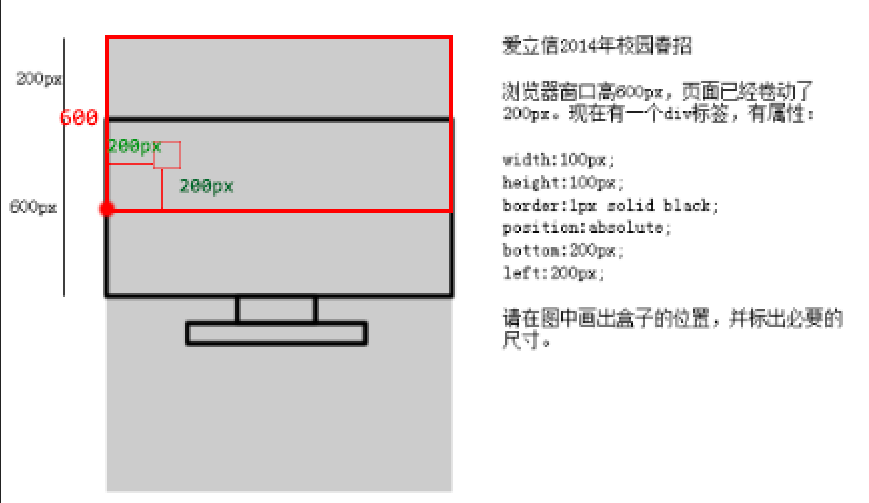
有bottom参与的绝对定位，参考点是我们页面首屏的右下角或左下角。



考题：



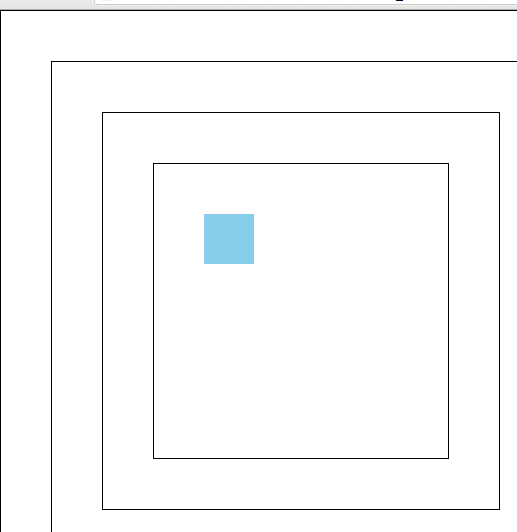
答案：



### ②祖先元素有定位的参考点

定位参考盒子：参考的是祖先元素中距离最近的有定位的盒子。

|  |
| --- |
| 1. <div class="box1"> //没有定位 2. <div class="box2"> //有相对定位 3. <div class="box3"> //没有定位 4. <div class="box4"> //有相对定位，距离绝对定位子元素最近 5. <p></p> 6. </div> 7. </div> 8. </div> 9. </div> |



结果：相对于box4盒子进行了绝对定位。

子绝父相：相对定位的元素，它本身不脱离标准流，结构比较固定。是我们常用的绝对定位的方式。

父级元素只要是任何一种定位，都会被当作参考盒子。

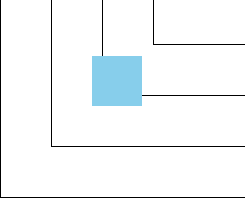
如：子绝父绝

|  |
| --- |
| 1. <div class="box1"> 2. <div class="box2"> //有相对定位 3. <div class="box3"> 4. <div class="box4"> //有绝对定位 5. <p></p> //相对于box4定位，因为距离最近的定位 6. </div> 7. </div> 8. </div> 9. </div> |

子绝父固。

### ③绝对定位的顶点。

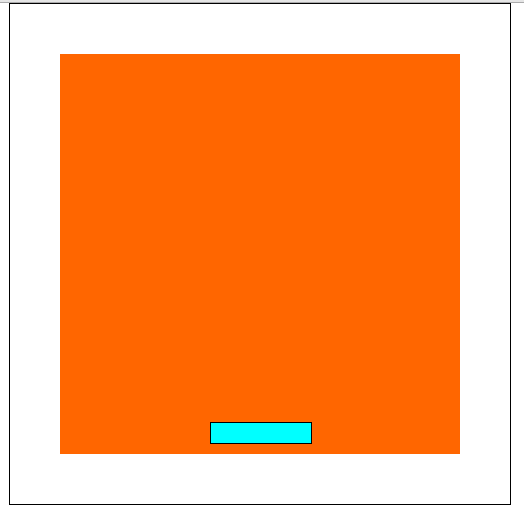
绝对定位的元素会忽视祖先元素的padding。定位的点是在border的内部。



有背景的四个顶点就是我们绝对定位的四个参考点。

### ④利用绝对定位来做压盖效果。

|  |
| --- |
| 1. <div class="box1"> //父级元素，相对定位 2. <div class="box2"> //橙色区域被压盖住 3. </div> 4. <div class="box3"></div> //青色区域，绝对定位，压盖住box2 5. </div> |



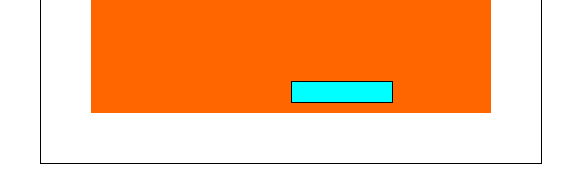
### ⑤绝对定位元素水平居中

不能再使用margin：0 auto；标准流里块级元素的居中。

我们用一种特殊的方法做绝对定位元素在祖先元素内的居中。

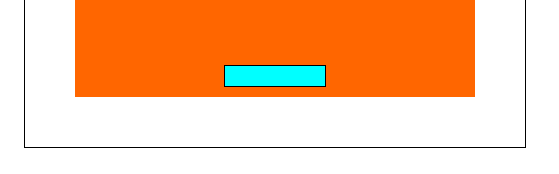
第一步：left值为50%。父盒子的padding+width的50%。左边缘是居中位置：

|  |
| --- |
| 1. left: 50%; |



第二步：将绝对定位的元素往左拽回他本身宽度的一般。利用margin-left为负值。

|  |
| --- |
| 1. margin-left: -51px; |



实际应用：两种轮播图。

干蹦轮播图：



|  |
| --- |
|  |

代码制作圆：

|  |
| --- |
| 1. border-radius: 50%; |

滚动轮播图：

解决图片3像素边距的问题：

|  |
| --- |
| 1. display：block； |

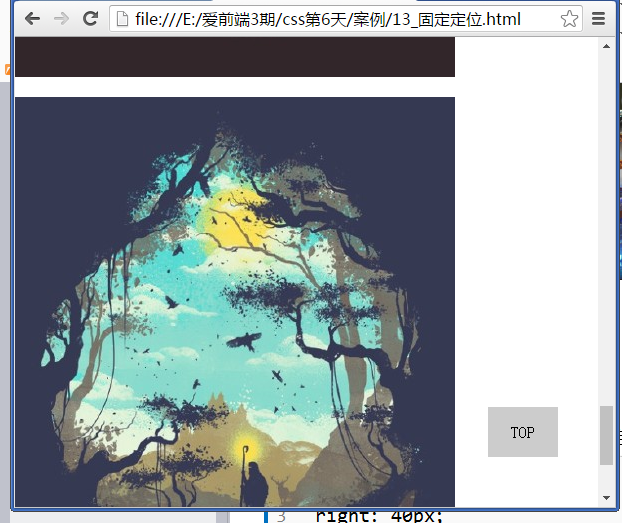


## 3、固定定位

position：fixed；

固定定位针对浏览器的四个角进行定位，始终出现在我们浏览器窗口内。

|  |
| --- |
| 1. position: fixed; 2. bottom: 50px; 3. right: 40px; |



参考点：浏览器窗口的四个点。

四个方向的值：left、right、top、bottom。

固定定位脱离标准流。

## z-index属性

### 默认压盖顺序

①定位的压盖没有定位的。

②有定位的元素，html结构顺序在后面的压盖在前面的。

### 自定义压盖顺序

需要用z-index属性。

属性值：数字，没有单位，数字越大，压盖顺序越靠上。

①没有定位的元素不能加z-index属性。浮动也不能加。

|  |
| --- |
| 1. z-index：5； |

②定位的元素，值越大，压盖顺序越靠上。

③如果定位的元素z-index值一样，html结构顺序在后面的压盖在前面的。

④不管子级的z-index多大，只要父级z-index小，压盖不住其他的。

父亲怂了，儿子再厉害也没用。

|  |
| --- |
| 1. <div class="big"> //5 2. <div class="big\_son"></div> //4444 3. </div> 4. <div class="small"> //6 5. <div class="small\_son"></div> //333 6. </div> |