

# 目录

# Contents

◆ 定位

**目标：能够说出四种定位的应用场景，能够使用子绝父相，实现淘宝轮播图（焦点图）案例**

学习路径

1. 定位能解决的问题
2. 定位的组成
3. 四种定位模式
4. 定位案例（土豆）



## 4.1 定位能解决的问题

### 目标：知道并明确定位的应用场景 — 装饰

1. 能够把装饰性小盒子叠加在一个大盒子上，并且随意摆放位置，同时不干扰大盒子的内容（BFC）



2. 能够固定在屏幕的某一个位置



**目标：能够说出四种定位的应用场景，能够使用子绝父相，实现淘宝轮播图（焦点图）案例**

学习路径

1. 定位能解决的问题
2. 定位的组成
3. 四种定位模式
4. 定位案例（土豆）

## 4.2. 定位的组成

### 目标：知道定位的两个组成部分

1. 要使用定位需要**两**类属性
  1. 定位模式属性：如何定位？
  2. 边偏移属性：定位在哪里？
2. **定位模式**属性 **position**（位置）+ **边偏移**属性

序号	值	语义
1	static	静态定位
2	relative ☆	相对定位
3	absolute ☆	绝对定位
4	fixed ☆	固定定位
5	sticky	粘性定位

边偏移属性	示例	描述
top	top: 80px	顶端偏移量，定义元素相对于其父元素上边线的距离。
bottom	bottom: 80px	底部偏移量，定义元素相对于其父元素下边线的距离。
left	left: 80px	左侧偏移量，定义元素相对于其父元素左边线的距离。
right	right: 80px	右侧偏移量，定义元素相对于其父元素右边线的距离。

提示：标准流和浮动不能使用边偏移属性

静态定位就是不定位

**目标：能够说出四种定位的应用场景，能够使用子绝父相，实现淘宝轮播图（焦点图）案例**

学习路径

1. 定位能解决的问题
2. 定位的组成
3. 四种定位模式
4. 定位案例（土豆）



## 4.3.1 相对定位

**目标：理解相对定位的特点：不脱标 + 相对自身原有位置定位 + 层次升级**

1. 属性值：relative
2. 特点：
  1. 不脱标
  2. 相对自身位置定位
  3. 提升显示层次

提示：相对定位大多与绝对定位联用，负责占地方

注意：不要同时指定 左 + 右 或者 上 + 下



## 4.3.2 绝对定位

目标：理解绝对定位的特点：**脱标** + **相对有定位的父亲**左上角位置定位

1. 属性值：**absolute**
2. 特点：
  - 脱标
  - 相对**最近有定位属性**的父亲定位
3. 布局方案（**子绝父相**）
  1. 父盒子采用相对定位，看起来和标准流没有区别，在标准流中占位置，按照标准流方式布局
  2. 子盒子采用绝对定位，在父盒子中随意摆放点缀盒子





## 4.3.3 固定定位

目标：理解固定定位的应用场景及特点：**脱标** + 固定在屏幕上的固定位置

1. 属性值：**fixed**
2. 特点：
  - 脱标
  - 固定在屏幕上的固定位置
  - 至少要指定一个**上**或者**下**的边偏移属性



## 4.3.4 粘性定位（了解）

目标：理解粘性定位的应用场景及特点：**保留原有位置** + 根据情况固定在屏幕上

1. 属性值：**sticky**
2. 特点：
  - 不脱标，按照标准流布局
  - 当屏幕滚动到某一个位置后，会固定在屏幕上
  - 至少要指定一个边偏移属性
  - 极少使用（兼容性不好）

提示：粘性定位的兼容性不好，开发时通常使用 JavaScript 实现粘性定位的效果

## 4.3 定位模式小结

目标：能够说出四种定位的应用场景

序号	值	语义	脱标	移动位置	常用	场景
1	static	静态定位	否	不移动	几乎不用	不定位
2	relative ☆	相对定位	否	相对自身	常用	子绝父相
3	absolute ☆	绝对定位	是	带定位的父级	常用	子绝父相
4	fixed ☆	固定定位	是	浏览器可视区	常用	浏览器可视区
5	sticky	粘性定位	否	浏览器可视区	不常用	浏览器可视区

**目标：能够说出四种定位的应用场景，能够使用子绝父相，实现淘宝轮播图（焦点图）案例**

学习路径

1. 定位能解决的问题
2. 定位的组成
3. 四种定位模式
4. 定位案例（土豆）

一个完整的网页，是由标准流、浮动、定位一起完成布局的，每个都有自己的专门用法。

1. 标准流
  1. 块级元素上下排列 / 行内元素或行内块元素左右排列
  2. 垂直的块级盒子显示就用标准流布局
2. 浮动
  1. 让多个块级元素一行显示或者左右对齐盒子
  2. 多个块级盒子水平显示就用浮动布局
3. 定位
  1. 如果元素自由在某个盒子内摆放就用定位布局
  2. 定位最大的特点是有层叠的概念，就是可以让多个盒子前后叠压来显示



## 一. 定位扩展+练习



黑马程序员  
www.itheima.com

传智播客旗下高端IT教育品牌

**目标：加深对定位布局应用的理解，能够完成淘宝轮播图（焦点图）案例**

学习路径

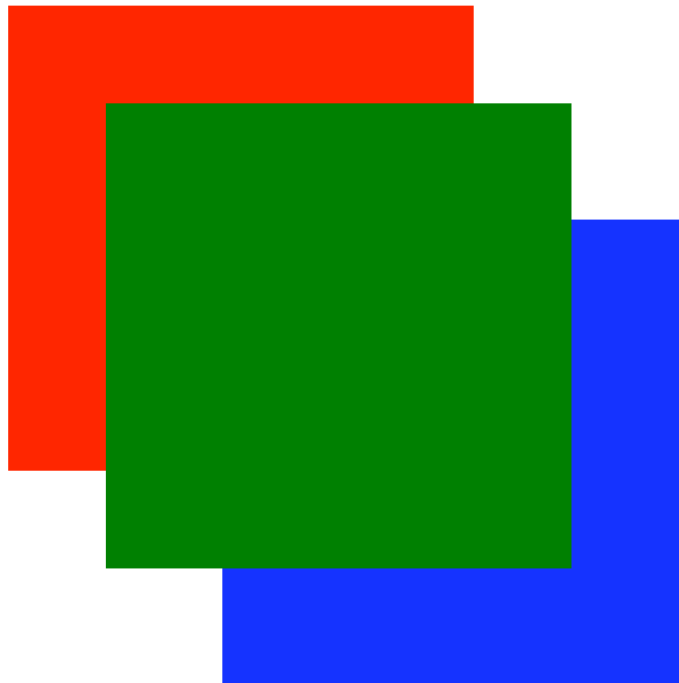
1. 叠放顺序
2. 位置计算
3. 综合案例



## 1.1 叠放顺序

**目标：能够使用 `z-index` 控制定位元素的层级**

1. 应用场景：在使用定位时，如果出现盒子重叠的情况，调整盒子的叠放顺序
2. 属性：`z-index`
3. 属性值：数字（数值越大，盒子越靠前）





## 一. 定位扩展+练习



黑马程序员  
www.itheima.com

传智播客旗下高端IT教育品牌

**目标：加深对定位布局应用的理解，能够完成淘宝轮播图（焦点图）案例**

学习路径

1. 叠放顺序
2. 位置计算
3. 综合案例



## 1.2 位置计算 – 绝对定位的盒子水平垂直居中

目标：能够说出定位的盒子水平垂直居中对齐的算法；

1. 应用场景：让绝对定位的盒子水平垂直居中，用户登录窗口常用

2. 实现代码：

```
position: absolute;
top: 0;
right: 0;
bottom: 0;
left: 0;
margin: auto;
```

提示：固定定位的盒子是脱标的，周围没有兄弟，直接使用 margin 值因为没有边界，无法居中。



## 1.2.1 位置计算 – 绝对定位的盒子水平居中

目标：理解靠绝对定位水平居中的计算方法并能够用代码实现

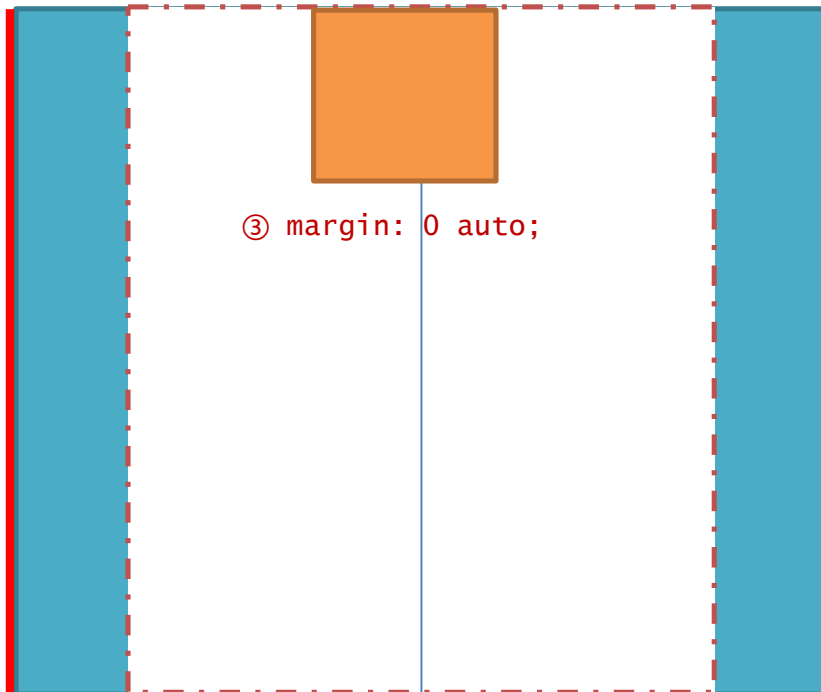
1. 应用场景：让绝对定位的盒子水平居中

2. 实现思路：

① `left: 0;`

② `right: 0;`

③ `margin: 0 auto;`



提示：固定定位的盒子是脱标的，周围没有兄弟，设置 margin 值会把自己推走

注意：指定了哪个方向的边偏移，哪个方向才有参考的边

## 1.2.2 位置计算 – 绝对定位的盒子水平垂直居中

目标：能够说出定位的盒子水平垂直居中对齐的算法；

1. 应用场景：让绝对定位的盒子水平垂直居中，用户登录窗口常用

2. 实现代码：

```
position: absolute;
top: 0;
right: 0;
bottom: 0;
left: 0;
margin: auto;
```

提示：固定定位的盒子是脱标的，周围没有兄弟，直接使用 margin 值因为没有边界，无法居中。

注意：指定了哪个方向的边偏移，哪个方向才有参考的边

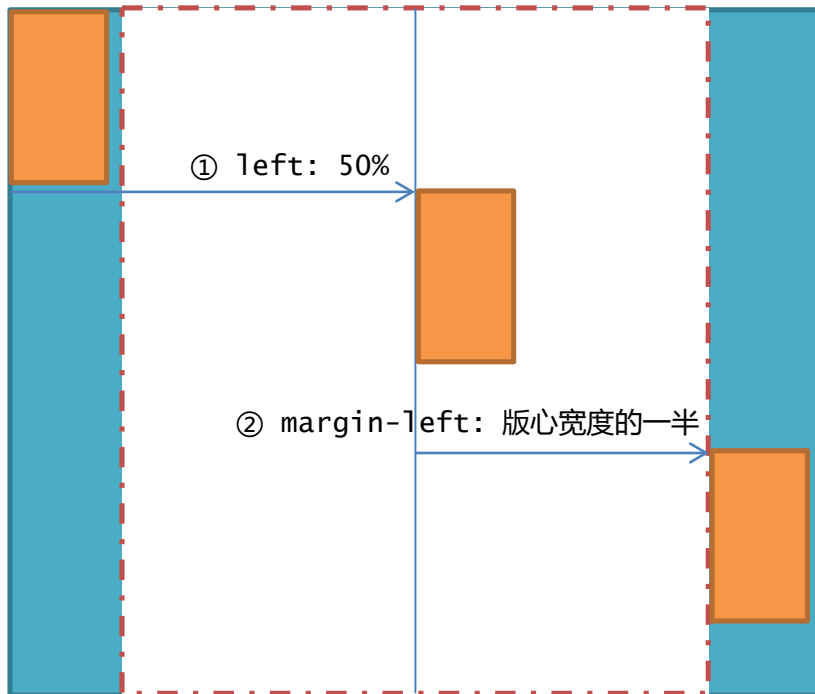


## 1.2.3 位置计算 – 固定定位靠版心右侧

目标：理解靠版心右侧的计算方法并能够用代码实现

1. 应用场景：使用固定定位，让盒子参照版心定位，不受浏览器的宽高变化影响

2. 实现思路：



提示：在网页中，通常把水平居中的可视区域称为**版心**

提示：固定定位的盒子是脱标的，周围没有兄弟，指定了哪个方向的边偏移，哪个方向才有参考的边



## 一. 定位扩展+练习



黑马程序员  
www.itheima.com

传智播客旗下高端IT教育品牌

**目标：加深对定位布局应用的理解，能够完成淘宝轮播图（焦点图）案例**

学习路径

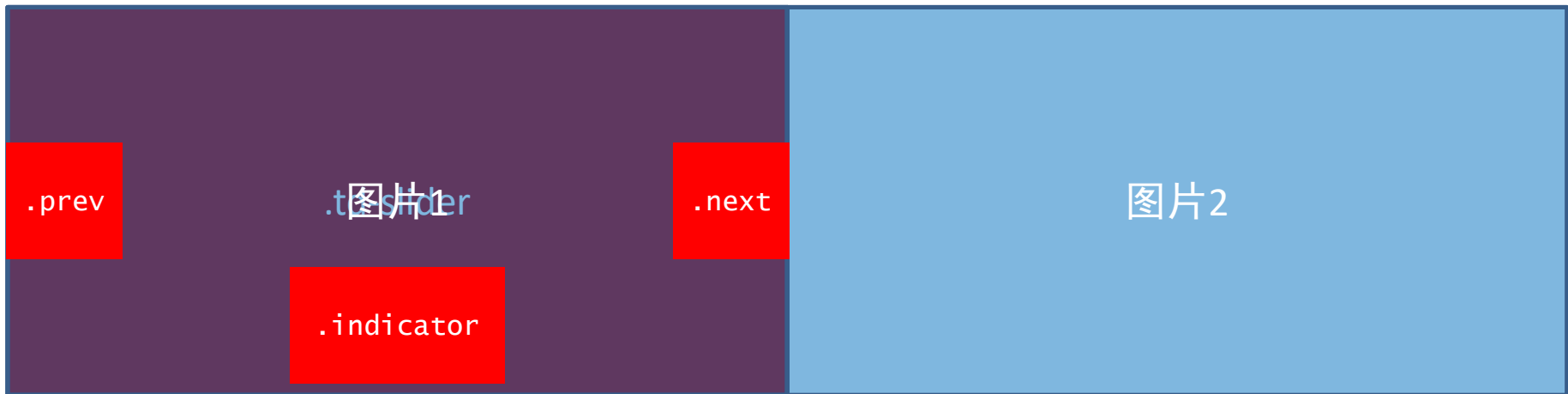
1. 叠放顺序
2. 位置计算
3. 综合案例 - 淘宝轮播图



## 1.3 淘宝轮播图 – 分析

目标：加深对子绝父相应用的理解，能够完成淘宝轮播图（焦点图）案例

1. 案例效果
2. 结构分析（标准流 → 浮动 → 定位）
  1. 标准流：`.tb-slider>ul>li*2`
  2. 浮动：`li>div.pic>img`
  3. 定位：`.prev+.next+ul.indicator>li*5`



3. 代码实现
  1. 先结构，后样式，最后填内容
  2. 从外向内，从上向下，从左向右



## 1.3 淘宝轮播图 – 实现

目标：加深对子绝父相应用的理解，能够完成淘宝轮播图（焦点图）案例

1. 结构分析（标准流 → 浮动 → 定位）
  1. 标准流：`.tb-slider>ul.list>li*2`
  2. 浮动：`li>div.pic>img`
  3. 定位：`.prev+.next+ul.indicator>li*5`
2. 参数信息
  1. `tb-slider` 大小：520px \* 280px
  2. `.prev .next` 大小：20px \* 30px / 背景色：`rgba(0, 0, 0, 0.3)` / 鼠标指针：小手
    1. `.prev` 左侧垂直居中 / 背景位置：5px -890px
    2. `.next` 右侧垂直居中 / 背景位置：9px -940px
    3. 默认隐藏，鼠标悬停在轮播图上才显示
  3. `.indicator` 大小：70px \* 13px / 背景色：`rgba(255, 255, 255, 0.3)` / 圆角边框
    1. 水平居中，距离底部 15px
    2. 默认隐藏，鼠标悬停在轮播图上才显示
  4. `.indicator li` 大小：8px \* 8px / 左浮动 / `margin 3px;` / 圆形 / 背景色：`#fff` | `#f50`
3. 代码实现
  1. 先结构，后样式，最后填内容
  2. 从外向内，从上向下，从左向右
  3. 标准流 → 浮动 → 定位（每一个盒子内部重复上述 1~2 步）

## 目标：能够知道鼠标样式的作用，并至少记住三种常见的鼠标样式

1. 场景：
  1. 当鼠标经过某一个元素时，改变形状，吸引用户的注意力
  2. 经常和 **:hover** 一起搭配使用，体现页面的制作细节
2. 属性：**cursor**
3. 属性值：

属性	描述
<code>default</code>	小白，默认
<code>pointer</code> ☆	小手，提示用户可以点击
<code>zoom-in</code>	放大，提示用户可以点击放大
<code>zoom-out</code>	缩小，提示用户可以点击缩小
<code>move</code>	移动，提示用户可以拖拽
<code>text</code>	文本，提示用户可以输入内容
<code>not-allowed</code>	禁止，提示用户，例如购物车数量不能小于等于 0



## 目标：能够取消表单轮廓线，能写出防止文本域拖拽的样式

### 1. 场景：

表单点击的时候，会有默认轮廓样式，为了美观一般需要取消

### 2. 属性：outline

### 3. 属性值： none 或者 0

### 1. 场景：

文本域默认可以拉伸，这样会影响页面布局，一般直接取消文本域的拖拽效果

### 1. 属性：resize

### 2. 属性值： none

### 1. 场景：

文本框点击之后，边框会有颜色变化，可以选择获得焦点的表单，并设置其样式属性：

### 2. 选择器： input:focus { 样式 }

## 目标：能够知道如何利用边框原理绘制三角形（高频面试题）

### 1. 原理：

1. 利用边框之间的衔接处是斜线连接
2. 先将四个边都设置为 **transparent** 色
3. 需要哪个方向的三角形就设置哪个方向的 **border** 颜色

### 2. 效果：



## 目标：能够知道显示隐藏的**两种方式**和应用场景

1. 场景：希望某一个元素在屏幕上不可见，例如鼠标 **:hover** 显示隐藏效果
2. 显示隐藏的两种方法：
  1. **display: none** 和 **display: block**
  2. **visibility: hidden**
3. 区别：
  1. **display: none** 会隐藏元素，同时**空间也不会保留**（重点，常与 JavaScript 联用实现动画效果）
  2. **visibility: hidden** 会隐藏元素，但是**空间会保留**（极少用）

提示：显示隐藏是一次性完成，不能加过渡效果

## 目标：能够知道溢出隐藏的方式

1. 场景：子盒子内容超过父盒子不可见，可以使用 overflow

### 可能的值

值	描述
visible	默认值。内容不会被修剪，会呈现在元素框之外。
hidden	内容会被修剪，并且其余内容是不可见的。
scroll	内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容。
auto	如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容。
inherit	规定应该从父元素继承 overflow 属性的值。

提示：溢出隐藏是非常重要的地位

CSS 的 **vertical-align** 属性使用场景：经常用于设置图片或者表单(行内块元素) 和文字垂直对齐。

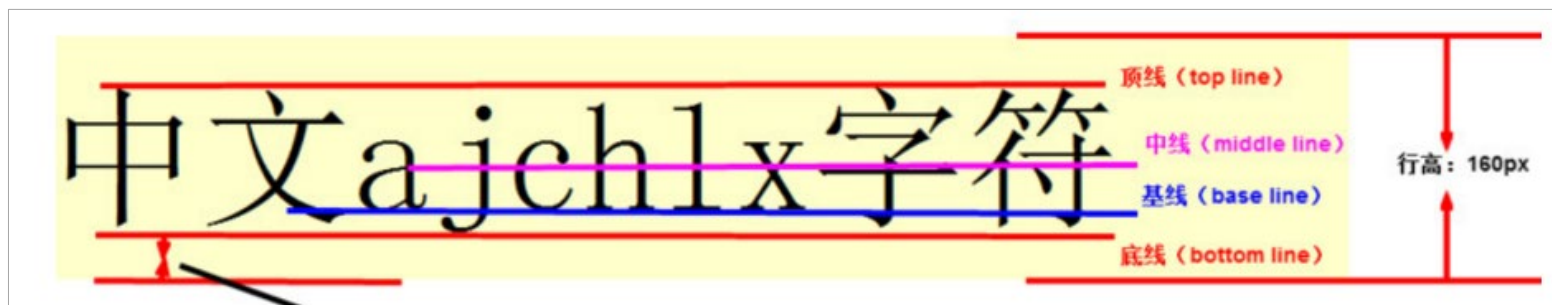
官方解释：用于设置一个元素的**垂直对齐方式**，但是它只针对于行内元素或者行内块元素有效。

语法：

```
vertical-align : baseline | top | middle | bottom
```

值	描述
baseline	默认。元素放置在父元素的基线上。
top	把元素的顶端与行中最高元素的顶端对齐
middle	把此元素放置在父元素的中部。
bottom	把元素的顶端与行中最低的元素顶端对齐。

```
vertical-align : baseline | top | middle | bottom
```



## 1 图片、表单和文字对齐

图片、表单都属于行内块元素，默认的 vertical-align 是基线对齐。



此时可以给图片、表单这些行内块元素的 vertical-align 属性设置为 middle 就可以让文字和图片垂直居中对齐了。

## 2 解决图片底部默认空白缝隙问题

bug: 图片底侧会有一个空白缝隙, 原因是行内块元素会和文字的基线对齐。

主要解决方法有两种:

1. 给图片添加 `vertical-align:middle | top| bottom` 等。 (提倡使用的)
2. 把图片转换为块级元素 `display: block;`

```
view-source:www.1 x
view-source:www.163.com
21 table,td,tr,th{font-size:12px;}
22 ol,ul {list-style:none;}
23 li{list-style-type:none;}
24 img{vertical-align:top;border:0;}
25 h1,h2,h3,h4,h5,h6{font-size:inherit; font-weight:normal;}
26 address,cite,code,em,th,i{font-weight:normal; font-style:no
27 .hx a,.hx em,.fb{font-weight:bold;}
28 .clearfix{*zoom:1;}
29 .clearfix:after{display:block; overflow:hidden; clear:both;
```

```
www.sina.com.cn/t x
www.sina.com.cn/temp/css/79/index2016/v0510/index.css
@charset "UTF-8";
/* 30,79,24 2016-10-24 11:39:04 */
html,body,ul,li,ol,dl,dd,dt,p,h1,h2,h3,h4,h5,h6,form,fieldset,legend,in
{margin:0;padding:0}fieldset,img{border:0}img{display:block}address,car
ion,cite,code,dfn,th,var{font-style:normal;font-
weight:normal}ul,ol{list-style:none}input{padding-top:0;padding-
bottom:0;font-family:"SimSun","宋体"}input::-moz-focus-
inner{border:0;padding:0}select,input{vertical-
align:middle}select,input,textarea{font-
size:12px;margin:0}input[type="text"],input[type="password"],textarea{c
tline-style:none;-webkit-
appearance:none}textarea{resize:none}table{border-
```





传智播客旗下高端IT教育品牌