

# CSS元素定位

王红元 coderwhy

# 目录

## content



**1** 标准流布局

**2** 相对定位

**3** 固定定位

**4** 绝对定位

**5** 粘性定位

**6** z-index

# 标准流 (Normal Flow)

■ 默认情况下，元素都是按照**normal flow**（标准流、常规流、正常流、文档流【document flow】）进行排布

□ 从左到右、从上到下按顺序摆放好

□ 默认情况下，互相之间不存在层叠现象

inline	inline	inline-block
block		
block		
inline-block	inline	
block		
inline	inline-block	inline

<body>

```
<span>span1</span>
```

```

```

```
<span style="display: inline-block">span2</span>
```

```
<div>div</div>
```

```
<p>p</p>
```

```
<span style="display: inline-block">span</span>
```

```
<strong>strong</strong>
```

```
<h1>h1</h1>
```

```
<span>span3</span>
```

```
<span style="display: inline-block">span4</span>
```

```
<span>span5</span>
```

</body>

# margin-padding位置调整

- 在标准流中，可以使用margin、padding对元素进行定位

- 其中margin还可以设置负数

- 比较明显的缺点是

- 设置一个元素的margin或者padding，通常会影响到标准流中其他元素的定位效果

- 不便于实现元素层叠的效果

- 如果我们希望一个元素可以跳出标准量,单独的对某个元素进行定位呢?

- 我们可以通过position属性来进行设置;

# 认识元素的定位

■ 定位允许您从**正常的文档流布局中取出元素**，并使它们具有不同的行为：

- 例如**放在另一个元素的上面**；
- 或者**始终保持在浏览器视窗内的同一位置**；

■ 定位在开发中非常常见：



【精彩集锦】清清马超神威天将勇  
不可挡 九尾不知火舞三连决胜



纵使黑夜吞噬了一切，  
太阳还可以重新回来

京东  
秒杀

特色  
优选

频道  
广场

为你  
推荐

客服

反馈

顶部

# 认识position属性

- 利用position可以对元素进行定位，常用取值有5个：

```
static | relative | absolute | sticky | fixed
```

- 默认值：

- static：默认值, 静态定位

- 使用下面的值, 可以让元素变成 定位元素(positioned element)

- relative：相对定位

- absolute：绝对定位

- fixed：固定定位

- sticky：粘性定位

# 静态定位 - static

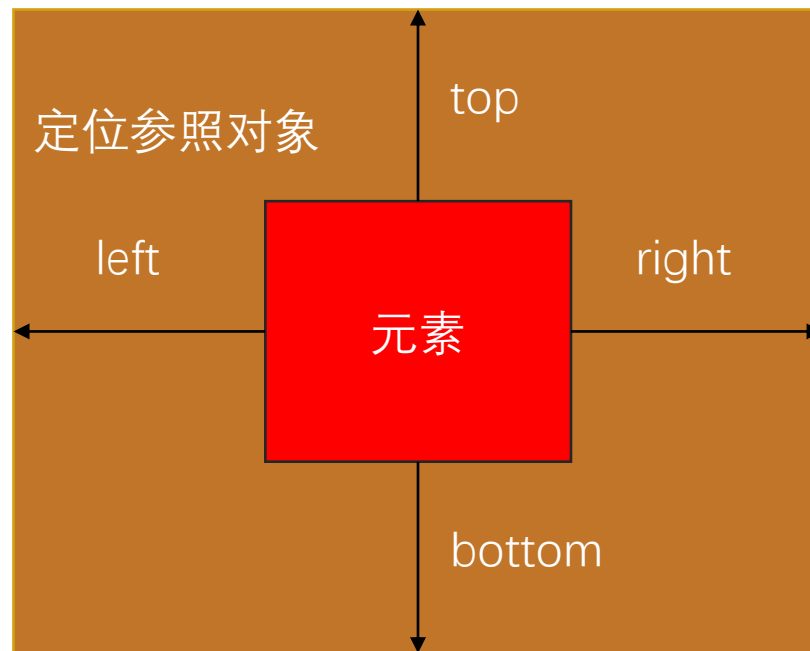
## ■ position属性的默认值

- 元素按照normal flow布局

- left、right、top、bottom没有任何作用

# 相对定位 - relative

- 元素按照**normal flow**布局
- 可以通过**left**、**right**、**top**、**bottom**进行定位
  - 定位**参照对象**是元素**自己原来的位置**
- left、right、top、bottom用来设置元素的具体位置，对元素的作用如下图所示



- 相对定位的应用场景
  - 在**不影响其他元素位置的前提下**，对当前元素位置进行微调

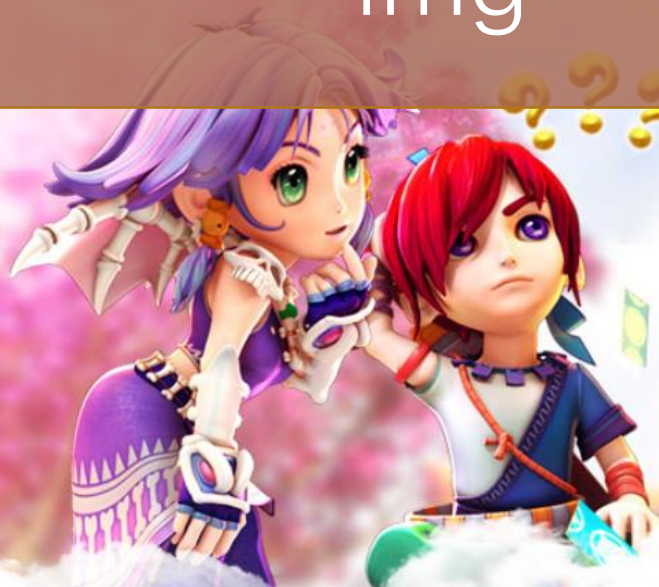


# 相对定位练习

div

img

官方玩法说明  
资料库



# 固定定位 - fixed

- 元素**脱离normal flow**（脱离标准流、脱标）
- 可以通过**left、right、top、bottom**进行定位
- 定位参照对象是视口（viewport）
- 当画布滚动时，固定不动

# 画布 和 视口

## ■ 视口 (Viewport)

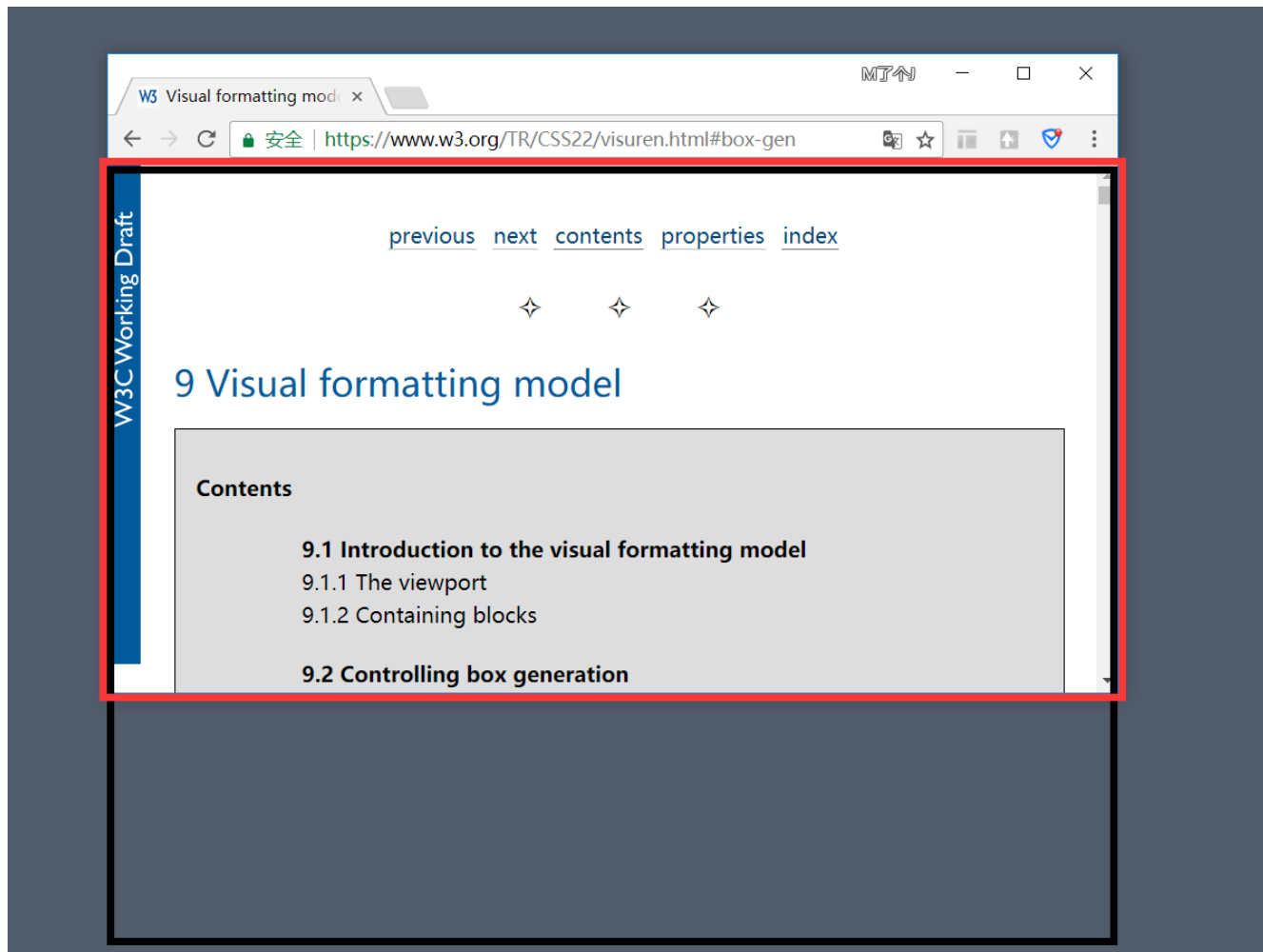
- 文档的可视区域
- 如右图红框所示

## ■ 画布 (Canvas)

- 用于渲染文档的区域
- 文档内容超出视口范围，可以通过滚动查看
- 如右图黑框所示

## ■ 宽高对比

- 画布  $\geq$  视口



# 定位元素的特点

- 可以随意设置宽高
- 宽高默认由内容决定
- 不再受标准流的约束
  - 不再严格按照从上到下、从左到右排布
  - 不再严格区分块级、行内级，块级、行内级的很多特性都会消失
- 不再给父元素汇报宽高数据
- 脱标元素内部默认还是按照标准流布局



# 固定定位练习



顶部

反馈

# 绝对定位 - absolute

- 元素脱离normal flow（脱离标准流、脱标）
- 可以通过left、right、top、bottom进行定位
  - 定位参照对象是最邻近的定位祖先元素
  - 如果找不到这样的祖先元素，参照对象是视口
- 定位元素（positioned element）
  - position值不为static的元素
  - 也就是position值为relative、absolute、fixed的元素

- 在绝大多数情况下，子元素的**绝对定位都是相对于父元素进行定位**
- 如果希望子元素相对于父元素进行定位，又不希望父元素脱标，常用解决方案是：
  - 父元素设置**position: relative**（让父元素成为定位元素，而且父元素不脱离标准流）
  - 子元素设置**position: absolute**
  - 简称为“**子绝父相**”

# 绝对定位练习

## ■ 绝对定位元素 (absolutely positioned element)

- position值为absolute或者fixed的元素

## ■ 对于绝对定位元素来说

- 定位参照对象的宽度 =  $\text{left} + \text{right} + \text{margin-left} + \text{margin-right} + \text{绝对定位元素的实际占用宽度}$
- 定位参照对象的高度 =  $\text{top} + \text{bottom} + \text{margin-top} + \text{margin-bottom} + \text{绝对定位元素的实际占用高度}$

## ■ 如果希望绝对定位元素的宽高和定位参照对象一样，可以给绝对定位元素设置以下属性

- `left: 0; right: 0; top: 0; bottom: 0; margin: 0;`

## ■ 如果希望绝对定位元素在定位参照对象中居中显示，可以给绝对定位元素设置以下属性

- `left: 0; right: 0; top: 0; bottom: 0; margin: auto;`
- 另外，还得设置具体的宽高值（宽高小于定位参照对象的宽高）



# 粘性定位 - sticky

■ 另外还有一个定位的值是 **position: sticky**, 比起其他定位值要新一些.

- sticky是一个大家期待已久的属性;
- 可以看做是**相对定位和固定定位的结合体**;
- 它允许被定位的元素**表现得像相对定位一样**, 直到它滚动到某个阈值点;
- 当**达到这个阈值点时**, 就会变成固定定位;

家具 家电 电脑 手机

- 列表10
- 列表11
- 列表12
- 列表13
- 列表14
- 列表15
- 列表16
- 列表17
- 列表18
- 列表19
- 列表20
- 列表21

■ sticky是相对于最近的滚动祖先包含视口的(the nearest ancestor scroll container' s scrollport )



# position值对比

	脱离标准流	定位元素	绝对定位元素	定位参照对象
static – 静态定位	×	×	×	×
relative – 相对定位	×	√	×	元素自己原来的位置
absolute – 绝对定位	√	√	√	最邻近的定位祖先元素 (如果找不到这样的祖先元素，参照对象是视口)
fixed – 固定定位	√	√	√	视口

# CSS属性 - z-index

## ■ z-index属性用来设置定位元素的层叠顺序（仅对定位元素有效）

□ 取值可以是正整数、负整数、0

## ■ 比较原则

□ 如果是兄弟关系

✓ z-index越大，层叠在越上面

✓ z-index相等，写在后面的那个元素层叠在上面

□ 如果不是兄弟关系

✓ 各自从元素自己以及祖先元素中，找出最邻近的2个定位元素进行比较

✓ 而且这2个定位元素必须有设置z-index的具体数值

