

CSS属性-定位

@M了个J

<https://github.com/CoderMJLee>

<https://space.bilibili.com/325538782>



实力IT教育 www.520it.com



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】）进行定位
- 从上到下、从左到右按顺序摆放好
- 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>

span1

span2

<div>div</div>

<p>p</p>

span

strong

<h1>h1</h1>

span3

span4

span5

</body>

margin、padding定位

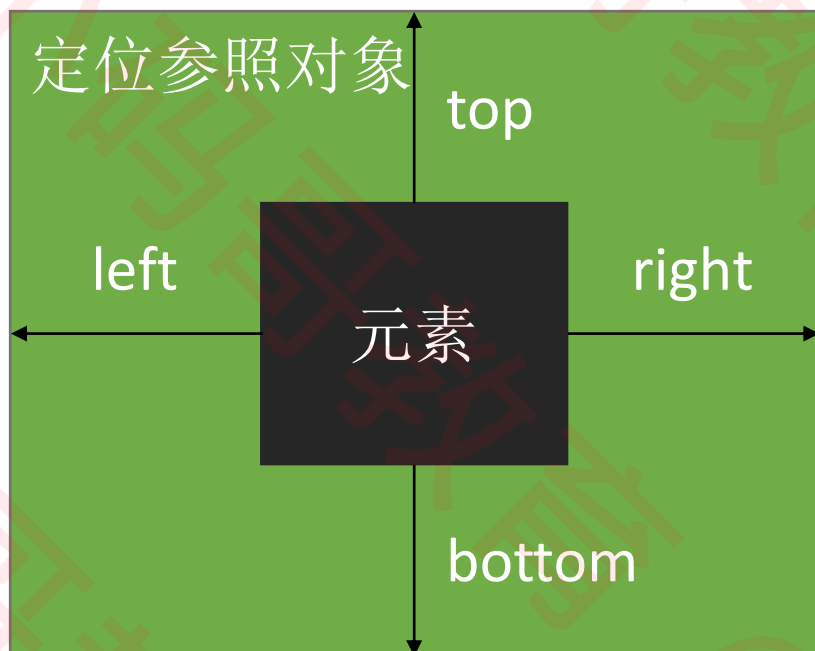
- 在标准流中，可以使用margin、padding对元素进行定位
- 其中margin还可以设置负数
- 比较明显的缺点是
 - 设置一个元素的margin或者padding，通常会影响到标准流中其他元素的定位效果
 - 不便于实现元素层叠的效果

CSS属性 - position

- 利用position可以对元素进行定位，常用取值有4个
 - static：静态定位
 - relative：相对定位
 - absolute：绝对定位
 - fixed：固定定位

relative - 相对定位

- 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

练习

- 已知：sub、sup元素分别用于设置上标和下标
- 尝试使用相对定位对sub、sup进行微调

请计算一下 $n_1 + n_2 + n^2$ 的值

练习

希望图片向左移动的距离：
= 图片宽度*0.5 - div宽度*0.5
= 960px - div宽度*0.5

img

官方玩法说明
资料库



static - 静态定位

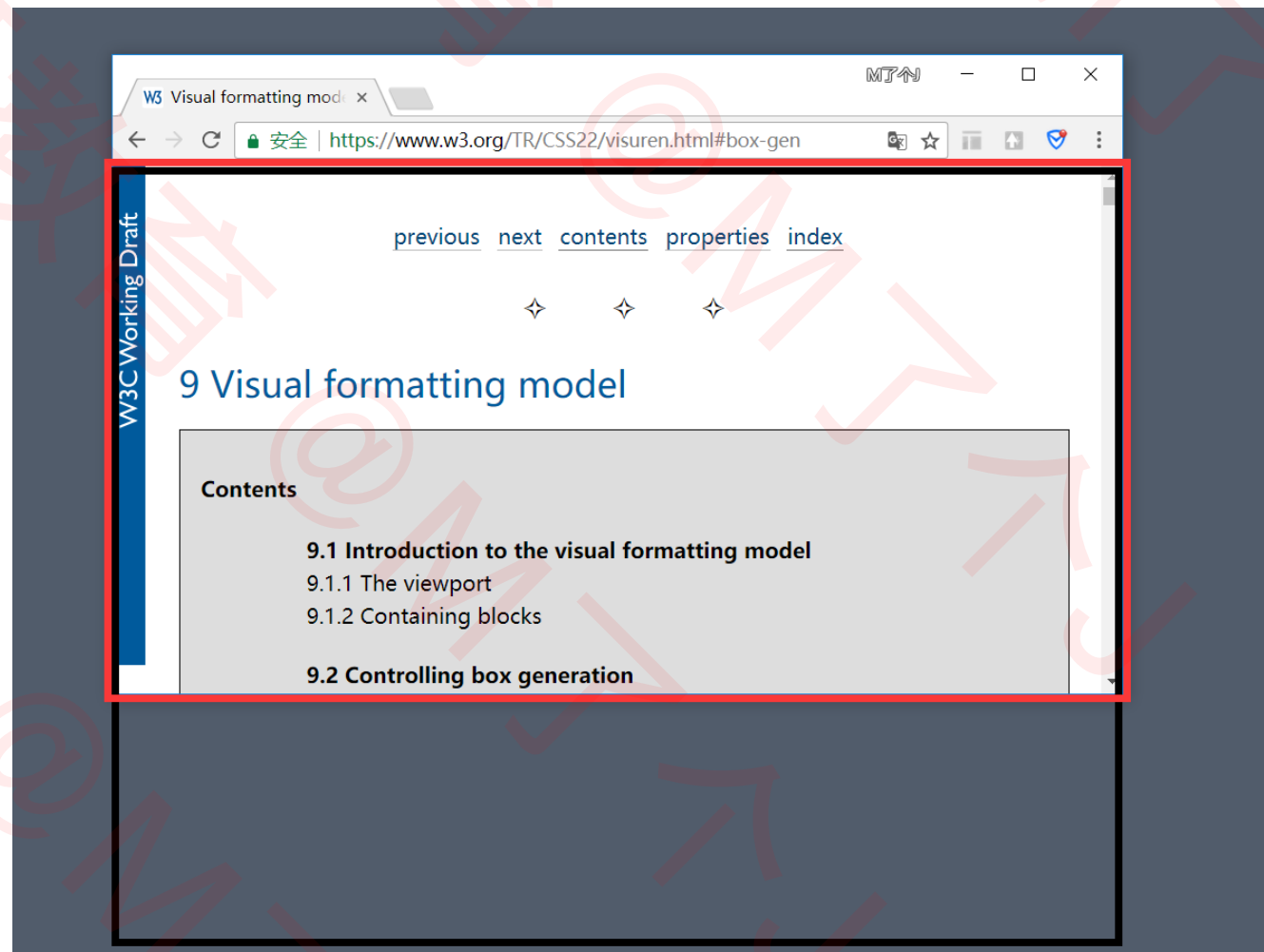
- position属性的默认值
- 元素按照normal flow布局
- left、right、top、bottom没有任何作用

fixed - 固定定位

- 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



脱标元素的特点

- 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



absolute - 绝对定位

- 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

子绝父相

- 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”



下载

下载APP



下载领1000元新人礼包

绝对定位技巧

- 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）

position总结

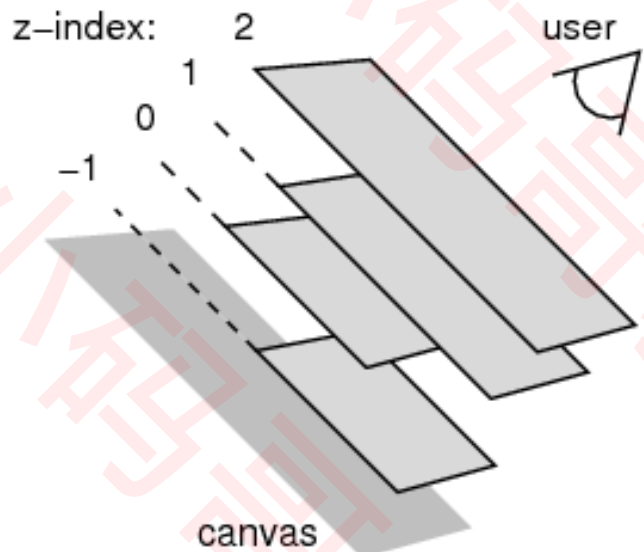
	脱离标准流	定位元素	绝对定位元素	定位参照对象
static – 静态定位	×	×	×	×
relative – 相对定位	×	√	×	元素自己原来的位置
absolute – 绝对定位	√	√	√	最邻近的定位祖先元素 (如果找不到这样的祖先元素, 参照对象是视口)
fixed – 固定定位	√	√	√	视口

元素的层叠

父子关系	子元素会层叠在父元素上	
非父子关系	都是非定位元素	在标准流中一般不存在层叠现象
	1个是定位元素 1个是非定位元素	定位元素会层叠在非定位元素上面
	都是定位元素	使用CSS属性z-index来控制层叠顺序

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