

CSS布局

浮动



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌

学习目标

Learning Objectives

- ◆ 能够使用 结构伪类选择器 在HTML中定位元素
- ◆ 能够说出 标准流 元素的布局特点
- ◆ 能够说出 浮动的布局特点
- ◆ 能够使用浮动完成 小米模块布局案例
- ◆ 能够说出 清除浮动 的目的，并能够使用清除浮动的方法



目录

Contents

- ◆ 结构伪类选择器
- ◆ 伪元素
- ◆ 标准流
- ◆ 浮动
- ◆ 清除浮动



目录

Contents

- ◆ 结构伪类选择器
- ◆ 伪元素
- ◆ 标准流
- ◆ 浮动
- ◆ 清除浮动

一、结构伪类选择器

目标：能够使用 **结构伪类选择器** 在HTML中定位元素

1. 作用与优势：

1. 作用：根据元素在文档中所在位置，通过结构关系查找元素
2. 优势：减少对于HTML中类的依赖，有利于保持代码整洁
3. 场景：常用于查找某父级选择器中的子元素

2. 选择器

选择器	说明
<code>E:first-child {}</code>	匹配父元素中第一个子元素，并且是E元素
<code>E:last-child {}</code>	匹配父元素中最后一个子元素，并且是E元素
<code>E:nth-child(n) {}</code>	匹配父元素中第n个子元素，并且是E元素
<code>E:nth-last-child(n) {}</code>	匹配父元素中倒数第n个子元素，并且是E元素

一、结构伪类选择器

注意点

➤ n 的注意点：

1. n 为：0、1、2、3、4、5、6、.....
2. 通过 n 可以组成常见公式

功能	公式
偶数	$2n$ 、even
奇数	$2n+1$ 、 $2n-1$ 、odd
找到前5个	$-n+5$
找到从第5个往后	$n+5$

一、结构伪类选择器

小结

➤ 结构伪类选择器有哪些常见的？

选择器	说明
<code>E:first-child {}</code>	匹配父元素中第一个子元素，并且是E元素
<code>E:last-child {}</code>	匹配父元素中最后一个子元素，并且是E元素
<code>E:nth-child(n) {}</code>	匹配父元素中第n个子元素，并且是E元素
<code>E:nth-last-child(n) {}</code>	匹配父元素中倒数第n个子元素，并且是E元素

一、结构伪类选择器

(拓展补充) 结构伪类选择器的易错点

- 问题：在下列案例中，如果需要找到第一个a标签，如何去查找？

```
<ul>
  <li><a href="#">我是第1个a标签</a></li>
  <li><a href="#">我是第2个a标签</a></li>
  <li><a href="#">我是第3个a标签</a></li>
  <li><a href="#">我是第4个a标签</a></li>
  <li><a href="#">我是第5个a标签</a></li>
  <li><a href="#">我是第6个a标签</a></li>
  <li><a href="#">我是第7个a标签</a></li>
  <li><a href="#">我是第8个a标签</a></li>
</ul>
```


一、结构伪类选择器

(拓展补充) nth-of-type

➤ 选择器：

选择器	说明
<code>E:nth-of-type(n) { }</code>	只在父元素的同类型(E)子元素范围中，匹配第n个

➤ 区别：

- `:nth-child` → 直接在所有孩子中数个数
- `:nth-of-type` → 先通过该 **类型** 找到符合的一堆子元素，然后在这一堆子元素中数个数



目录

Contents

- ◆ 结构伪类选择器
- ◆ 伪元素
- ◆ 标准流
- ◆ 浮动
- ◆ 清除浮动

二、伪元素

目标：能够使用 **伪元素** 在网页中创建内容

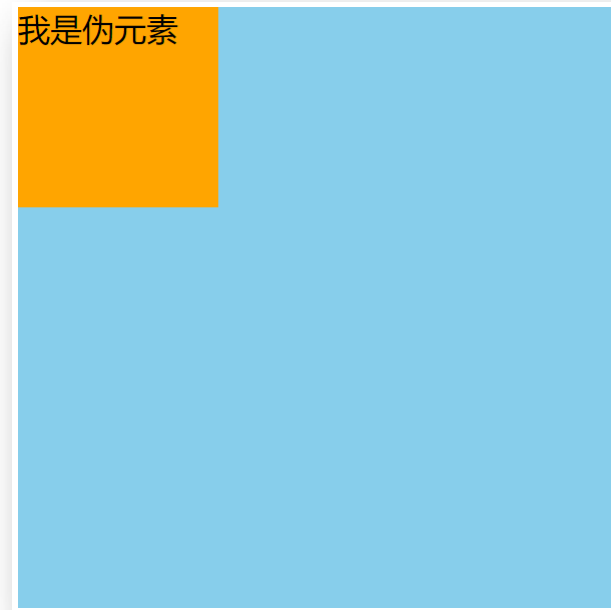
- 伪元素：一般页面中的非主体内容可以使用伪元素
- 区别：
 1. 元素：HTML设置的标签
 2. 伪元素：由css模拟出的标签效果

➤ 种类：

伪元素	作用
<code>::before</code>	在父元素内容的最前添加一个伪元素
<code>::after</code>	在父元素内容的最后添加一个伪元素

➤ 注意点：

1. 必须设置content属性才能生效
2. 伪元素默认是行内元素



二、伪元素

小结

- 伪元素的必加属性是？
 - content属性
- 伪元素创建出来后默认的显示模式是？
 - 行内元素



目录

Contents

- ◆ 结构伪类选择器
- ◆ 伪元素
- ◆ 标准流
- ◆ 浮动
- ◆ 清除浮动

目标：能够认识 **标准流** 的默认排布方式及其特点

- 标准流：又称**文档流**，是浏览器在渲染显示网页内容时默认采用的一套排版规则，规定了应该以何种方式排列元素
- 常见标准流排版规则：
 1. 块级元素：从上往下，**垂直布局**，独占一行
 2. 行内元素 或 行内块元素：从左往右，**水平布局**，空间不够自动折行

小结

- 标准流中块级元素的排版规则是？
 - 从上往下、**垂直布局**、独占一行
- 标准流中行内元素或行内块元素的排版规则是？
 - 从左往右、**水平布局**、空间不够自动折行



目录

Contents

- ◆ 结构伪类选择器
- ◆ 伪元素
- ◆ 标准流
- ◆ 浮动
- ◆ 清除浮动

四、浮动

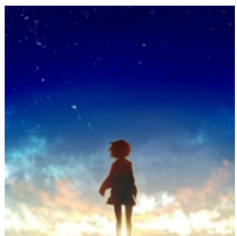
目标：能够认识使用 **浮动的作用**，了解 **浮动的特点**

学习路径：

1. **浮动的作用**
2. 浮动的代码
3. 浮动的特点
4. 浮动的案例

1.1 浮动的作用

➤ 早期的作用：图文环绕



前端开发是创建Web页面或app等前端界面呈现给用户的过程，通过HTML，CSS及JavaScript以及衍生出来的各种技术、框架、解决方案，来实现互联网产品的用户界面交互。它从网页制作演变而来，名称上有很明显的时代特征。在互联网的演化进程中，网页制作是Web1.0时代的产物，早期网站主要内容都是静态，以图片和文字为主，用户使用网站的行为也以浏览为主。随着互联网技术的发展和HTML5、CSS3的应用，现代网页更加美观，交互效果显著，功能更加强大。前端开发是创建Web页面或app等前端界面呈现给用户的过程，通过HTML，CSS及JavaScript以及衍生出来的各种技术、框架、解决方案，来实现互联网产品的用户界面交互。它从网页制作演变而来，名称上有很明显的时代特征。在互联网的演化进程中，网页制作是Web1.0时代的产物，早期网站主要内容都是静态，以图片和文字为主，用户使用网站的行为也以浏览为主。随着互联网技术的发展和HTML5、CSS3的应用，现代网页更加美观，交互效果显著，功能更加强大。

➤ 现在的作用：网页布局

- 场景：让垂直布局的盒子变成水平布局，如：一个在左，一个在右

左护法div

右护法div

2.1 浮动的代码

- 属性名：float
- 属性值：

属性名	效果
left	左浮动
right	右浮动

小结

- 浮动的作用是什么？
 1. 早期作用：图文环绕
 2. 现在作用：用于布局，让垂直布局的盒子变成水平布局，如：一个在左，一个在右
- 左浮动的属性是？右浮动的属性是？
 1. 左浮动：`float : left`
 2. 右浮动：`float : right`

四、浮动

目标：能够认识使用 **浮动的作用**，了解 **浮动的特点**

学习路径：

1. 浮动的作用
2. 浮动的代码
3. **浮动的特点**
4. 浮动的案例

3.1 浮动的特点

1. 浮动元素会脱离标准流（简称：脱标），在标准流中不占位置
 - 相当于从地面飘到了空中
 2. 浮动元素比标准流高半个级别，可以覆盖标准流中的元素
 3. 浮动找浮动，下一个浮动元素会在上一个浮动元素后面左右浮动
 4. 浮动元素会受到上面元素边界的影响
 5. 浮动元素有特殊的显示效果
 - 一行可以显示多个
 - 可以设置宽高
- 注意点：
- 浮动的元素不能通过text-align:center或者margin:0 auto，让浮动元素本身水平居中



小结

➤ 浮动元素的特点有哪些？

1. 浮动元素会脱标，在标准流中不占位置
2. 浮动元素比标准流高出半个级别，可以覆盖标准流中的元素
3. 浮动找浮动，下一个浮动元素会在上一个浮动元素后面左右浮动
4. 浮动元素会受到上面元素边界的影响
5. 浮动元素有特殊的显示效果：① 一行可以显示多个 ② 可以设置宽高

注意点：

1. 浮动的元素 一定要搭配一个 **标准流** 的父亲
2. 标准流负责上下排列大盒子， 里面左右排列的需要浮动
3. 父亲里面如果有一个孩子浮动了， 尽量所有的孩子都浮动 一浮全浮

四、浮动

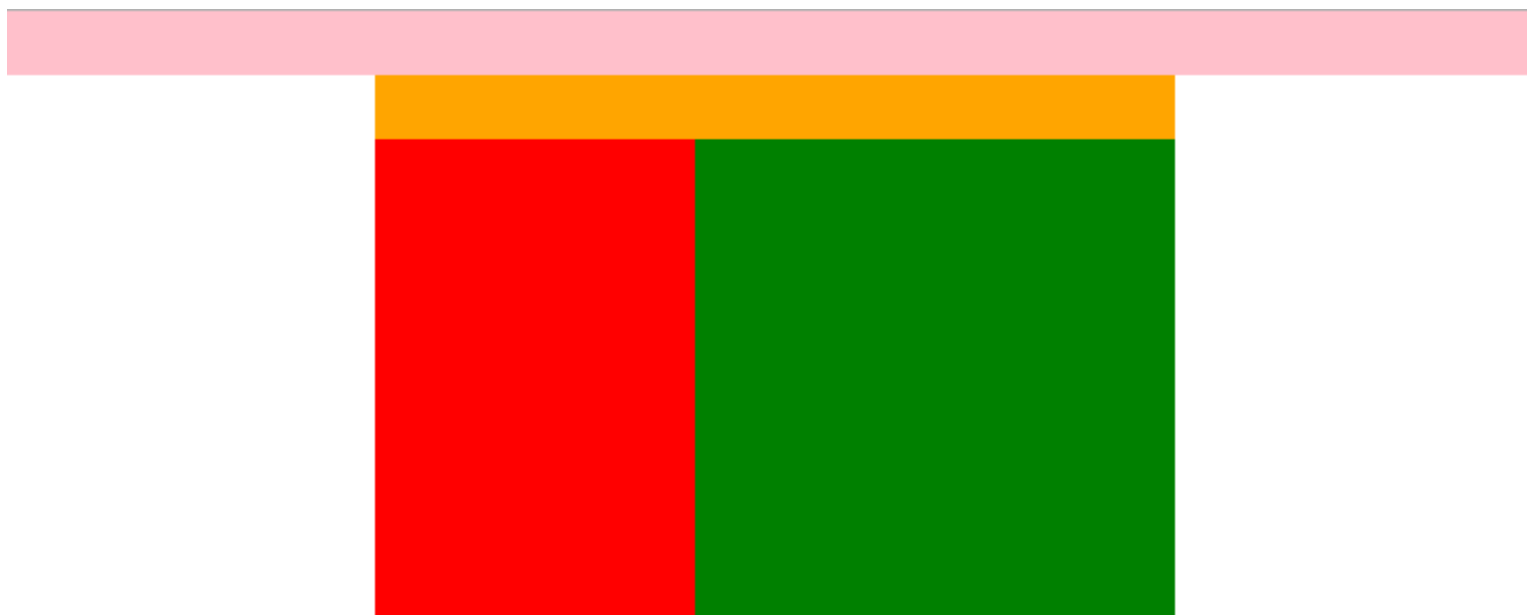
目标：能够认识使用 **浮动的作用**，了解 **浮动的特点**

学习路径：

1. 浮动的作用
2. 浮动的代码
3. 浮动的特点
4. **浮动的案例**

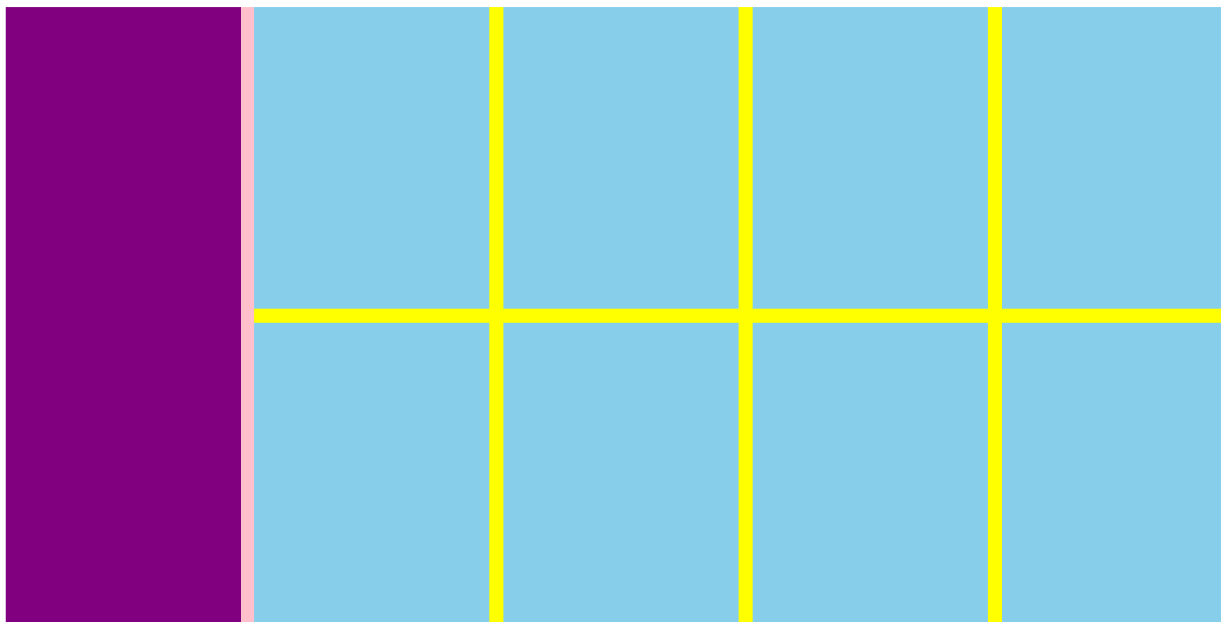
4.1 网页布局案例

- 需求：使用浮动，完成设计图中布局效果



4.2 小米模块案例

- 需求：使用浮动，完成设计图中布局效果



4.3 网页导航案例

- 需求：使用浮动，完成设计图中布局效果



- 导航栏开发常识：
 - 1. 比较重要的导航栏 `.nav>ul>li>a`
 - 2. 为了seo优化



目录

Contents

- ◆ 结构伪类选择器
- ◆ 伪元素
- ◆ 标准流
- ◆ 浮动
- ◆ 清除浮动

五、清除浮动

目标：能够认识 **清除浮动的目的**，并且能够使用**清除浮动的方法**

学习路径：

1. **清除浮动的介绍**
2. 清除浮动的方法

1.1 清除浮动的介绍

- 含义：清除浮动带来的影响
 - 影响：如果子元素浮动后，此时子元素不能撑开标准流的块级父元素
- 原因：子元素浮动后脱标 → 不占位置
- 目的：需要父元素有高度，从而不影响其他网页元素的布局

小结

- 清除浮动的含义是什么？
 - 清除浮动带来的影响
 - 影响：如果子元素浮动了，此时子元素不能撑开父元素
- 清除浮动的目的是什么？
 - 需要父元素有高度，从而不影响其他网页元素的布局

五、清除浮动

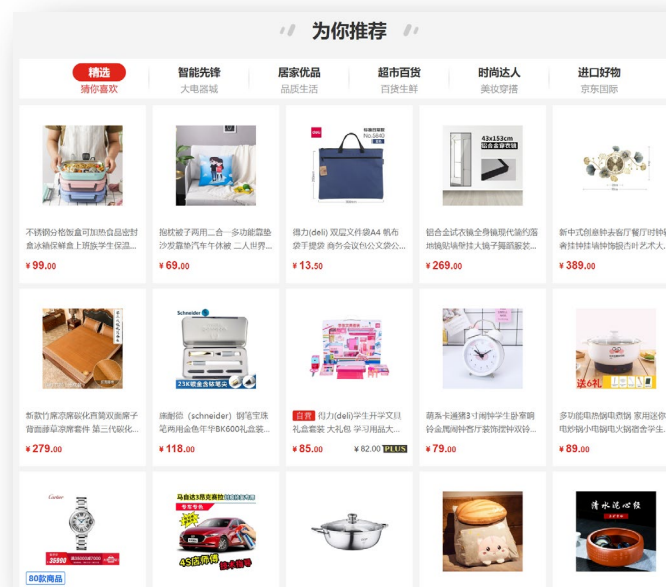
目标：能够认识 **清除浮动的目的**，并且能够使用**清除浮动的方法**

学习路径：

1. 清除浮动的介绍
2. **清除浮动的方法**

➤ 特点:

- 优点：简单粗暴，方便
- 缺点：有些布局中不能固定父元素高度。如：新闻列表、京东推荐模块



2.2 清除浮动的方法 — ② 额外标签法

➤ 操作：

1. 在最后一个浮动的元素后添加一个块级元素
2. 给添加的块级元素设置 `clear:both`

➤ 特点：

- 缺点：会在页面中添加额外的标签，会让页面的HTML结构变得复杂

2.3 清除浮动的方法 — ③ 单伪元素清除法

➤ 操作：用伪元素替代了额外标签

①：基本写法

```
.clearfix::after {  
    content: '';  
    display: block;  
    clear: both;  
}
```

②：补充写法

```
.clearfix::after {  
    content: '';  
    display: block;  
    clear: both;  
    /* 补充代码：在网页中看不到伪元素 */  
    height: 0;  
    visibility: hidden;  
}
```

➤ 特点：

- 优点：项目中使用，直接给标签加类即可清除浮动

2.4 清除浮动的方法 — ④ 双伪元素清除法

➤ 操作：

```
.clearfix::before,  
.clearfix::after {  
  content: '';  
  display: table;  
}  
.clearfix::after {  
  clear: both;  
}
```

➤ 特点：

- 优点：项目中使用，直接给标签加类即可清除浮动

2.5 清除浮动的方法 — ⑤ 给父元素设置 `overflow : hidden`

➤ 操作：

1. 直接给父元素设置 `overflow : hidden`

➤ 特点：

- 优点：方便

小结

➤ 清除浮动的方法有哪些？

1. 直接设置父元素高度
2. 额外标签法
3. 单伪元素清除法
4. 双伪元素清除法
5. 给父元素设置overflow : hidden

```
.clearfix::after {  
    content: '';  
    display: block;  
    clear: both;  
}
```

```
.clearfix::before,  
.clearfix::after {  
    content: '';  
    display: table;  
}  
.clearfix::after {  
    clear: both;  
}
```

(拓展补充) BFC的介绍

- 块格式化上下文（Block Formatting Context）： BFC
 - 是Web页面的可视CSS渲染的一部分，是块盒子的布局过程发生的区域，也是浮动元素与其他元素交互的区域。
- 创建BFC方法：
 1. html标签是BFC盒子
 2. 浮动元素是BFC盒子
 3. overflow属性取值不为visible。如：overflow : hidden
 4.
- BFC盒子常见特点：
 1. BFC盒子会默认包裹住内部子元素（标准流、浮动）→ 应用：清除浮动
 2. BFC盒子与子元素之间不存在margin的塌陷现象 → 应用：解决margin的塌陷
 3.

过渡 (transition)是CSS3中具有颠覆性的特征之一，我们可以在不使用 Flash 动画或 JavaScript 的情况下，当元素从一种样式变换为另一种样式时为元素添加效果。

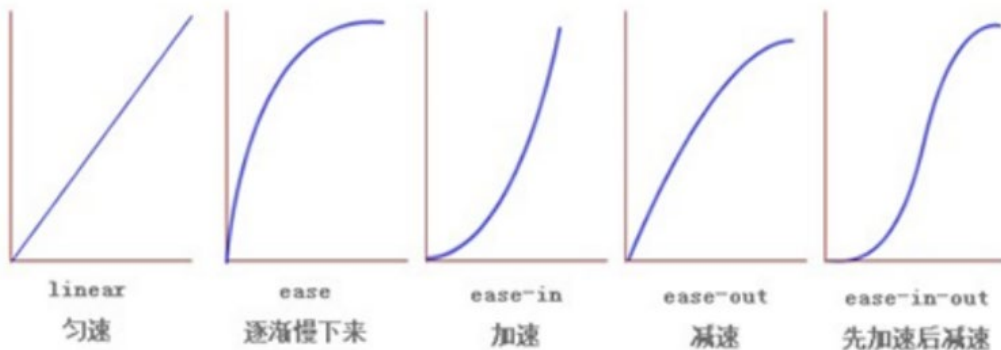
过渡动画： 是从一个状态 渐渐的过渡到另外一个状态

可以让我们页面更好看，更动感十足，虽然 低版本浏览器不支持（ie9以下版本） 但是不会影响页面布局。

我们现在经常和 :hover 一起 搭配使用。

transition: 要过渡的属性 花费时间 运动曲线 何时开始;

- 1.属性：** 想要变化的 css 属性， 宽度高度 背景颜色 内外边距都可以 。如果想要所有的属性都变化过渡， 写一个all 就可以。
- 2. 花费时间：** 单位是 秒（必须写单位） 比如 0.5s
- 3. 运动曲线：** 默认是 ease （可以省略）
- 4.何时开始：** 单位是 秒（必须写单位） 可以设置延迟触发时间 默认是 0s （可以省略）



记住过渡的使用口诀：谁做过渡给谁加

七. 圆角边框

在 CSS3 中，新增了圆角边框样式，这样我们的盒子就可以变圆角了。

新人福利

`border-radius` 属性用于设置元素的外边框圆角。

语法：

```
border-radius: length;
```

- 参数值可以为数值或百分比的形式
- 如果是正方形，想要设置为一个圆，把数值修改为高度或者宽度的一半即可，或者直接写为 50%
- 该属性是一个简写属性，可以跟四个值，分别代表左上角、右上角、右下角、左下角
- 分开写：`border-top-left-radius`、`border-top-right-radius`、`border-bottom-right-radius` 和 `border-bottom-left-radius`
- 兼容性 ie9+ 浏览器支持, 但是不会影响页面布局,可以放心使用.

CSS3 中新增了盒子阴影，我们可以使用 `box-shadow` 属性为盒子添加阴影。

新人福利

语法：

```
box-shadow: h-shadow v-shadow blur spread color inset;
```

值	描述
<i>h-shadow</i>	必需。水平阴影的位置。允许负值。
<i>v-shadow</i>	必需。垂直阴影的位置。允许负值。
<i>blur</i>	可选。模糊距离。
<i>spread</i>	可选。阴影的尺寸。
<i>color</i>	可选。阴影的颜色。请参阅 CSS 颜色值。
<i>inset</i>	可选。将外部阴影 (outset) 改为内部阴影。

注意：

1. 默认的是外阴影(outset), 但是不可以写这个单词, 否则造成阴影无效
2. 盒子阴影不占用空间, 不会影响其他盒子排列。



总结

- ◆ 能够使用 结构伪类选择器 在HTML中定位元素
- ◆ 能够说出 标准流 元素的布局特点
- ◆ 能够说出 浮动的布局特点
- ◆ 能够使用浮动完成 小米模块布局案例
- ◆ 能够说出 清除浮动 的目的，并能够使用清除浮动的方法



传智教育旗下高端IT教育品牌