

# 第10章 使用Div布局网页

---

# 导读

---

网页布局是网页设计的核心工作，选择好的网页布局方式至关重要。采用传统的表格布局、框架布局方式，使得网页的制作、修改、维护和更新复杂，代码杂乱无章。基于**WEB** 标准的网页设计的核心理念在于内容与形式的分离。以**Div**标签为代表的区块标签的引入，凭借**CSS**强大的样式设置功能，布局的网页更加规范，浏览器加载更快，受到业内人士的欢迎。**HTML5+CSS3**标准的正式提出，使得网页设计迈进了崭新时代。

# 本章主要内容

---

- AP Div的基本操作；
- Div与AP Div的区别与联系；
- Div与Span标记的区别与联系；
- 盒子模型；
- CSS的定位属性；
- 常用布局版式及应用。

## 10.1 案例1 使用AP Div布局电影资讯网页

---

**学习目标** 在Dreamweaver CS6中，创建AP Div元素，在其内部插入文字和图像，并使用CSS设置网页元素的属性。

**知识要点** AP Div及其特点；AP Div面板的属性设置；在AP Div中插入文字和图像；创建嵌套AP Div并设置其排列次序；使用CSS设置网页元素属性。

## 10.1 案例1 使用AP Div布局电影资讯网页

---

### 10.1.1 AP Div概述

Dreamweaver 将具有绝对位置属性的div 标签视为AP Div，也称为AP元素。在Dreamweaver CS6环境下，可以使用AP Div灵活方便的定位网页中的元素。网页元素放在AP Div中，通过控制AP Div的位置、堆叠次序、显示或隐藏等属性，从而实现网页的精确布局。

## 10.1 案例1 使用AP Div布局电影资讯网页

---

### 10.1.1 AP Div概述

AP Div最主要的特性在于它是浮动在网页内容之上的，因而不受网页中其他元素的影响，可以实现对AP Div元素的精确定位。通过定义AP Div的堆叠次序，可以实现文字阴影等效果；通过设置AP Div元素的显示或隐藏属性，可以实现AP Div内容的动态交替显示等动态效果。

AP Div的属性可以通过“属性”面板或“AP元素”面板进行设置。

## 10.1 案例1 使用AP Div布局电影资讯网页

### 10.1.2 AP Div元素的“属性”面板

在Dreamweaver中，可以通过“插入”面板中的“布局”选项卡创建AP Div，也可以使用“插入→布局对象→AP Div（A）”创建。选中创建后的AP Div元素，可在“属性”面板设置其属性。



## 10.1 案例1 使用AP Div布局电影资讯网页

### 10.1.3 “AP元素”面板

在Dreamweaver中，可以通过“AP元素”面板对AP Div元素进行重命名、设置AP Div属性、设置AP Div元素叠放次序、修改AP Div元素的可见性。

在Dreamweaver中，选择“窗口→AP元素”菜单命令或按<F2>键，均可打开“AP元素”面板。



## 10.1 案例1 使用AP Div布局电影资讯网页

---

### 10.1.4 在AP Div中添加网页元素

首先在文档中创建四个AP Div，分别调整其位置，然后依次添加图像和文字。通过设置标题文字的属性及叠加次序，实现文字的阴影特效。



## 10.1 案例1 使用AP Div布局电影资讯网页

---

### 10.1.4 在AP Div中添加网页元素

✦ 案例小结 使用AP Div可以布局网页，由于AP Div位置可以随意控制，因此布局较为简单。本案例介绍在网页中插入AP Div元素并设置其属性的方法，重点是嵌套AP Div的创建步骤，以及通过设置AP Div的堆叠次序，实现文字阴影特效的方法。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

CSS 布局的基本元素是Div 标签，它是文本、图像或其它页面元素的容器。使用CSS 布局网页时，首先创建 Div标签，然后在标签中添加内容并设置其CSS样式。可以用绝对方式（指定 x 和 y 坐标）或相对方式（指定与其它页面元素的距离）来定位 Div 标签。还可通过指定浮动、填充和边距设置Div标签之间的位置关系。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

 **学习目标** 了解AP Div与Div以及span与Div的区别与联系，了解Div标签在网页布局中的重要作用，掌握网页布局的盒子模型，掌握网页元素的定位方式，掌握CSS网页布局技巧及常用布局类型。

 **知识要点** Div标签，盒子模型，与网页元素定位相关的CSS样式规则：float、position、z-index，网页布局技巧，常用CSS+DIV网页布局版式。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 10.2.1 AP Div标签与Div标签

在Dreamweaver 中可以通过“插入”面板创建Div元素。

在“插入”面板中，选择“布局”选项卡，可以展开“布局”工具栏。在“标准”模式下，和Div标签有关的是“插入Div标签”按钮 和“绘制AP Div”按钮 。

“插入Div标签”按钮：单击此按钮，在文档中插入一个Div标签。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 10.2.1 AP Div标签与Div标签

“绘制AP Div”按钮：单击此按钮，将鼠标指针移到文档窗口中，鼠标指针变为“+”字状，按下鼠标并拖曳出一个AP Div元素。

在Dreamweaver中分别插入Div标签和AP Div元素。

在设计视图中，Div标签由虚线框表示，而AP Div是实线框。用鼠标拖动标签，发现Div标签的位置不可移动，而AP Div可以任意移动位置。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 10.2.2 Div标签与span标签

在HTML中，Div与span 是常用标签。使用CSS控制其样式，可以实现各种布局效果。Div标签在HTML3.0时代就已出现，直到CSS引入后才逐渐发挥其优势。span标记直到HTML4.0时才被引入，是专门针对样式表而设计的标记。

Div是区块容器标记，可以容纳文字、图片、表格等各种网页元素。在使用时，可以将Div中的内容视为独立的对象，通过定义Div的CSS样式，可以控制内部元素的显示效果。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 10.2.2 Div标签与span标签

**span**标记作为容器被广泛应用于HTML语言。在**span**标签中同样可以容纳各种HTML元素。

二者区别在于，**Div**是块级元素，它可以实现自动换行；**span**是行内元素，不会自动换行。由此可见，**Div**标签用于网页布局，而**span**标签没有结构意义，是为应用样式而构造的标签。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 10.2.2 Div标签与span标签

在Dreamweaver中输入以下代码，并在拆分视图中观察显示效果。

```
<h2>生命存在的前提条件</h2>
```

```
<span>淡水</span>
```

```
<span>充足的空气</span>
```

```
<span>适宜的温度</span>
```

```
<hr>
```

```
<div>淡水</div>
```

```
<div>充足的空气</div>
```

```
<div>适宜的温度</div>
```

## 10.2 学习任务：使用CSS+Div布局网页基础

### 10.2.2 Div标签与span标签

```
4 <title>div与span</title>
5
6 </head>
7 <body>
8 <h2>生命存在的前提条件</h2>
9 <span>淡水</span>
10 <span>充足的空气</span>
11 <span>适宜的温度</span>
12
13 <hr>
14 <div>淡水</div>
15 <div>充足的空气</div>
16 <div>适宜的温度</div>
17 </body>
18 </html>
19
```

#### 生命存在的前提条件

淡水 充足的空气 适宜的温度

淡水  
充足的空气  
适宜的温度

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 10.2.2 Div标签与span标签

可以看出，**<span>**标记中定义的内容在同行显示，而**<div>**标记定义的内容会另起一行显示。块元素和行内元素可以相互转换，可以通过定义CSS的**display**属性值实现。**display**用于设置生成框的类型，通过设置不同属性，可以转换元素显示方式。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 10.2.2 Div标签与span标签

分别修改部分span标签和Div标签的display属性，并查看显示效果。如图10-14所示。

```
<span style="display:block;">充足的空气</span>
```

```
<span style="display:block;">适宜的温度</span>
```

```
...
```

```
<div style="display:inline;">充足的空气</div>
```

```
<div style="display:inline;">适宜的温度</div>
```

## 10.2 学习任务：使用CSS+Div布局网页基础

---

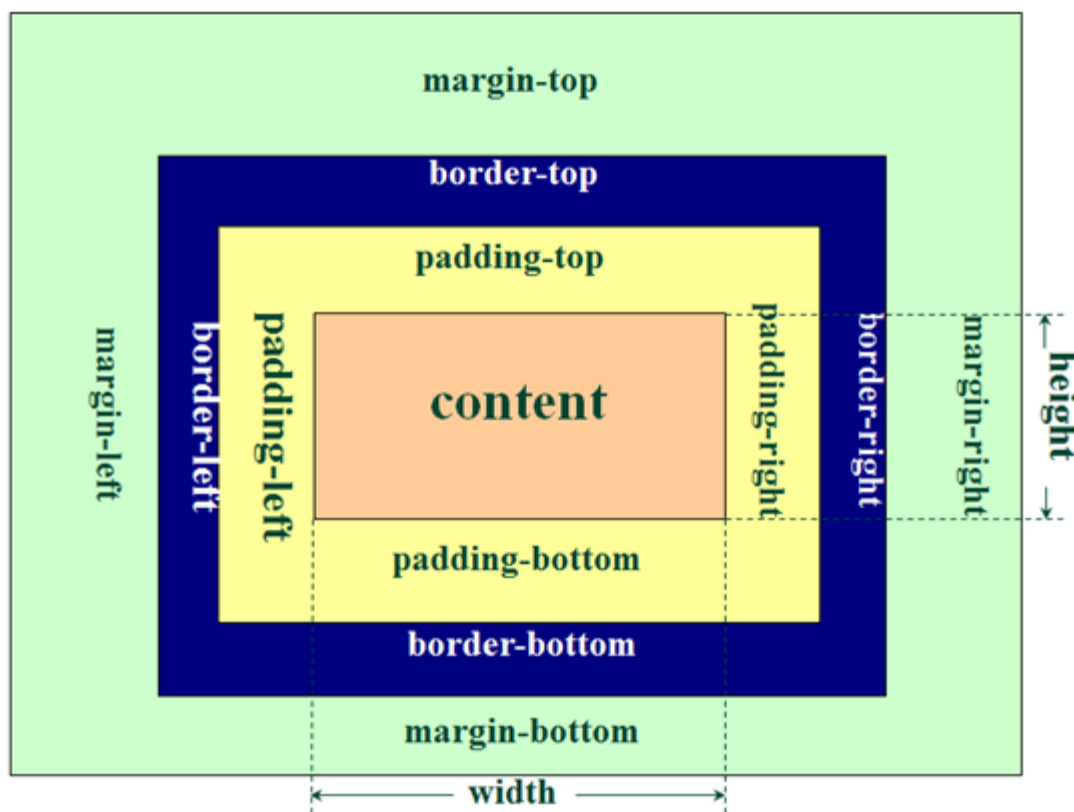
### 10.2.3 盒子模型

CSS+Div网页布局的精髓在于盒子模型。盒子模型（Box Model）用于描述一个为HTML元素形成的矩形盒子。盒子模型还涉及为各个元素调整外边距（margin）、边框（border）、内边距（padding）和内容（content）的具体操作。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 10.2.3 盒子模型



## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 10.2.3 盒子模型

对于盒子模型，可以借助日常生活中的盒子理解。

**content**是盒子里盛的物体，**padding**是盒子内壁和物体间的填充物，**border**是盒子本身，**margin**是盒子间的空隙。在网页中，**content**指文字、图片等元素，也可以是嵌套盒子。与现实生活中盒子不同的是，**CSS**的盒子是具有弹性的，内容可以大于盒子。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 10.2.3 盒子模型

Article 1:

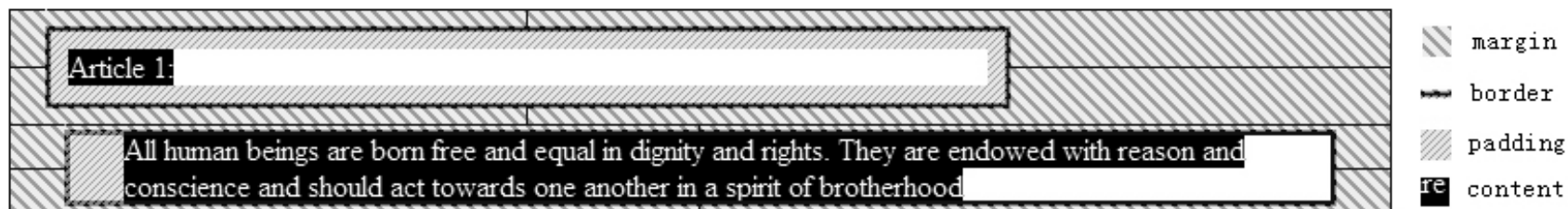
All human beings are born free and equal in dignity and rights. They are endowed with reason and conscience and should act towards one another in a spirit of brotherhood

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 10.2.3 盒子模型

本例包含了两个Div标签，分别设置其ID为title和content。这两个元素的盒子模型如图9-15所示。



## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 10.2.3 盒子模型

可以对margin、border、padding属性进行整体设置，依次为上、右、下、左顺时针方向，也可以单独设置某一侧的属性值，如margin-left。本实例关于盒子模型的部分CSS代码如下：

## 10.2 学习任务：使用CSS+Div布局网页基础

---

...

```
#title{  
    margin:10px 200px 10px 20px;  
    padding:10px;  
    border:1px;  
}
```

```
#content{  
    margin:4px 30px 4px 30px;  
    padding:0px 0px 0px 30px;  
    border:0px;  
}
```

...

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 10.2.3 盒子模型

提示：对于margin、border、padding属性，可以按照规定的顺序，给出2个、3个或者4个属性值，它们的含义将有所区别，具体含义如下：

- （1）如果给出2个属性值，前者表示上下边框的属性，后者表示左右边框的属性；
- （2）如果给出3个属性值，前者表示上边框的属性，中间的数值表示左右边框的属性，后者表示下边框的属性；
- （3）如果给出4个属性值，依次表示上、右、下、左边框的属性，即顺时针排序。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 10.2.4 元素的定位

网页的元素必须有合理的位置，从而构成有序的页面。网页元素的定位是通过**position**、**float**和**z-index**属性完成的。通过元素的定位，可以对网页中的块元素进行排版。

#### 1. float属性

**float**定位是CSS排版中的最重要的属性，用来定义元素向哪个方向浮动。**float**属性有三个值，**left**、**right**、**none**，当块元素设置了向左或向右浮动时，元素就会向其父元素左侧或右侧浮动。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 2. position属性

**position** 属性规定元素的定位类型。这个属性定义建立元素布局所用的定位机制。任何元素都可以定位，不过绝对或固定元素会生成一个块级框，而不论该元素本身是什么类型。相对定位元素会相对于它在正常流中的默认位置偏移。

## 10.2 学习任务：使用CSS+Div布局网页基础

| 值        | 描述                                                                                        |
|----------|-------------------------------------------------------------------------------------------|
| absolute | 生成绝对定位的元素，相对于 static 定位以外的第一个父元素进行定位。<br>元素的位置通过 “left”、“top”、“right” 以及 “bottom” 属性进行规定。 |
| fixed    | 生成绝对定位的元素，相对于浏览器窗口进行定位。<br>元素的位置通过 “left”、“top”、“right” 以及 “bottom” 属性进行规定。               |
| relative | 生成相对定位的元素，相对于其正常位置进行定位。<br>因此，“left:20” 会向元素的 left位置添加 20 像素。                             |
| static   | 默认值。没有定位，元素出现在正常的流中（忽略 top、bottom、left、right 或者 z-index 声明）。                              |
| inherit  | 规定应该从父元素继承 position 属性的值。                                                                 |

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 3. z-index属性

**z-index** 属性设置元素的堆叠顺序。拥有更高堆叠顺序的元素总是会处于堆叠顺序较低的元素的前面。该属性设置一个定位元素**z** 轴的位置，**z** 轴定义为垂直延伸到显示区的轴。通常网页元素的**z-index**值如果为正数，则离用户更近，为负数则表示离用户更远。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 10.2.5 常用CSS+Div布局版式

使用**CSS**布局网页，遵循内容与形式分离的原则。内容是网页的核心，形式是网页内容的直观表现，网页外观由内容和形式共同决定。在进行布局时，首先整体使用**Div**标签分块，然后设置各区块的**CSS**样式，最后为各个区块添加内容。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

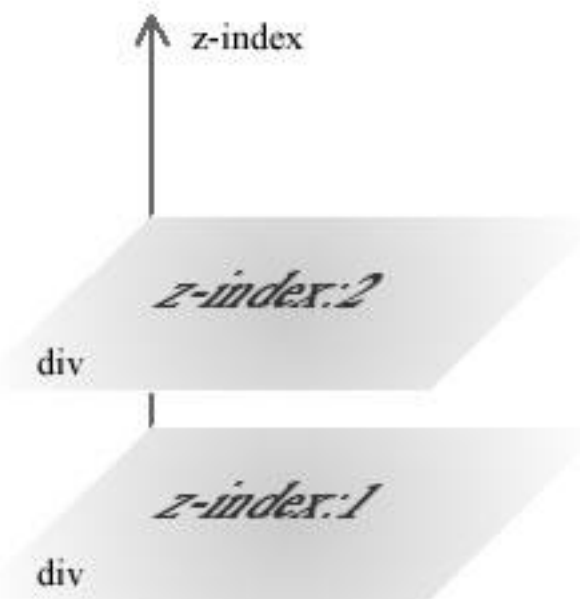
### 10.2.5 常用CSS+Div布局版式

使用**CSS**布局网页，真正实现网页内容与形式的分离，且排版灵活，更新容易，已经成为网页布局的主流技术。目前常见**CSS+Div**布局版式有网页内容居中布局、两列式布局、三列式布局等。两列式布局又可分为两列固定宽度居中布局、两列百分比布局、两列右列宽度自适应布局。下面以两列百分比布局、网页内容居中布局版式为例，介绍使用**CSS+Div**布局网页的步骤。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 10.2.5 常用CSS+Div布局版式



## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 1. 两列式宽度自适应布局版式

#### （1）使用Div划分页面模块

在进行网页布局时，首先对页面进行整体规划，包括模块划分，模块间的嵌套关系等。简单网页通常由标题模块（**banner**）、导航模块（**navigator**）、内容模块（**content**）和版权信息模块（**footer**）构成。各模块间的有机组合，形成不同的网页版式。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 1. 两列式宽度自适应布局版式

在网页布局时，通常引入用于辅助定位的Div标签“#container”，将所有Div放在父级元素container中，以实现网页元素的整体定位。对于每个子Div，可以在其内部加入Div标签或者其他元素。如图10-20所示，在左侧的划分中，在content标签中插入“#left”、“#right”两个Div。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 1. 两列式宽度自适应布局版式

#container

#banner

#navigator

#content

#footer

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### （2）使用CSS定位Div标签的位置

网页版块划分完毕，接着需要确定网页版块间的位置关系，即网页版式或网页布局类型，本节所涉及的版式是两列百分比布局版式。

在使用CSS设置样式时，页面整体定位效果由body标签和#container标签设置，采用宽度100%自适应布局方式。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### (2) 使用CSS定位Div标签的位置

CSS代码如下所示：

```
body {  
    margin:0px;  
}  
  
#container {  
    width:100%;  
}
```

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### （2）使用CSS定位Div标签的位置

设置body标签的margin为0px，#container标签的width为100%，使网页宽度和浏览器宽度自适应。接下来设置#banner、#navigator、#content和#footer模块的CSS样式。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### （2）使用CSS定位Div标签的位置

分别设置#left和#right的CSS样式，并利用float属性设置其浮动属性为“left”。设置#left的宽度为“20%”，#right宽度为“80%”。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### (2) 使用CSS定位Div标签的位置

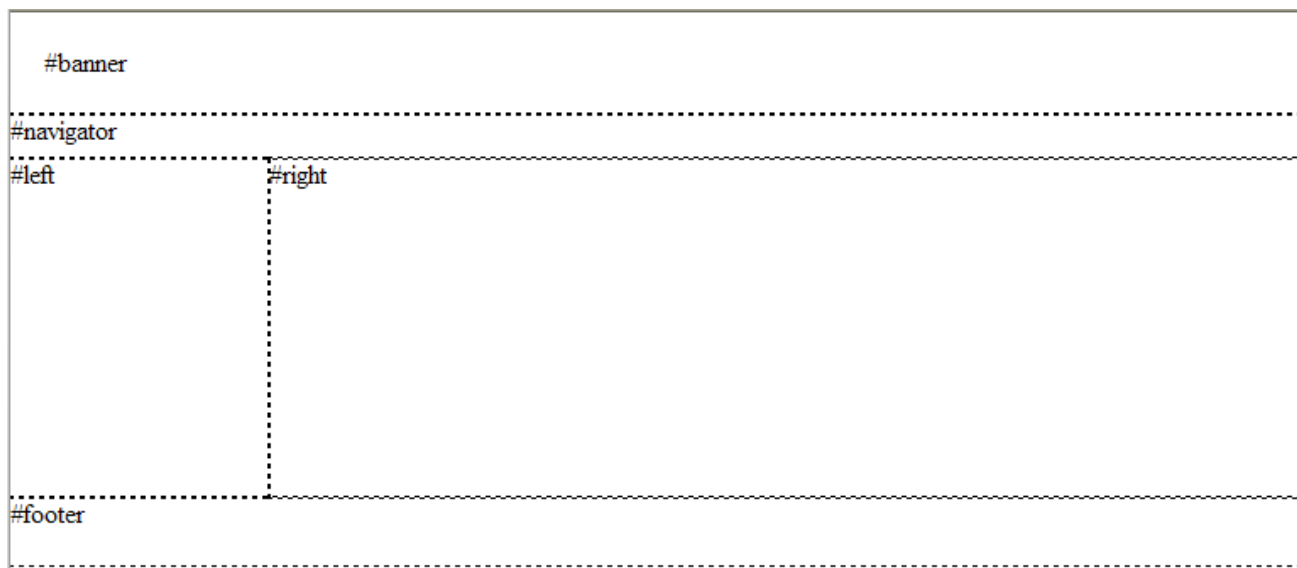
```
#left{  
    float:left;  
    width:20%;  
    height:200px; //假设高度为200px;  
}  
#right{  
    float:left;  
    width:80%;  
    height:200px; //假设高度为200px;  
}
```

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### (2) 使用CSS定位Div标签的位置

为防止#footer模块被#content中的元素覆盖，可以为其添加“clear:both;”属性，以清除float属性的影响。



## 10.2 学习任务：使用CSS+Div布局网页基础

---

### （2）使用CSS定位Div标签的位置

两列式布局模式还有一侧固定另一侧侧自适应版式。其中固定宽度只需将宽度值width设置为绝对像素值。对于左侧固定右侧自适应，只需设置左侧宽度值为绝对像素值，右侧Div的margin-left属性为200px即可。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 2. 网页内容居中布局版式

所谓内容居中，又称为固定宽度且居中，通常是指#container标签的宽度是固定的。网页宽度与显示分辨率有关。如果网页浏览者使用1024\*768像素分辨率的显示器，需设置#container的宽度为1002px，如果使用1280\*800像素分辨率的显示器，则设置#container的宽度值为1258px。即#container与显示器间的像素差值为22px。

实现固定宽度且居中的布局版式由多种方法。下面介绍两种常用方法。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### (1) 通过#container的margin属性实现

代码如下所示：

```
body{  
    margin:0px;  
    text-align:center;  
}  
#container{  
    width:1002px;  
    margin:0 auto;  
    text-align:left;  
}
```

---

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### (1) 通过#container的margin属性实现

其中，代码“margin:0px;”指定body标签的间距为0。“text-align:center;”将页面所有元素都设置为居中对齐。设置#container的“margin:0 auto;”，使模块上下边界距离为0，而左右自适应调整。最后设定“text-align:left;”，用来覆盖body标签设置的对齐方式，使#container中的元素左对齐。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### (2) margin属性的另类使用

代码如下所示：

```
body{  
    margin:0px;  
}  
#container{  
    width:1002px;  
    position:relative;  
    left:50%;  
    margin-left:-501px;  
}
```

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### (2) margin属性的另类使用

对于#container采取“position:relative”相对定位方法，并设置其left属性值为50%，即相对于body标签的距离为50%。然后用margin-left属性将块向左拉回501px，即实现整体居中效果。

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### 3. 两侧固定，中间宽度自适应式三列式布局

接两列百分比自适应式布局，在#left和#right中间添加“ID”属性为#main的Div标签。

(1) 调整“content” Div内的三个Div标签的次序

```
<div id="content">  
  <div id="left">#left</div>  
  <div id="right">#right</div>  
  <div id="center">#center</div>  
</div>
```

## 10.2 学习任务：使用CSS+Div布局网页基础

---

### （2）设置Div标签的CSS样式

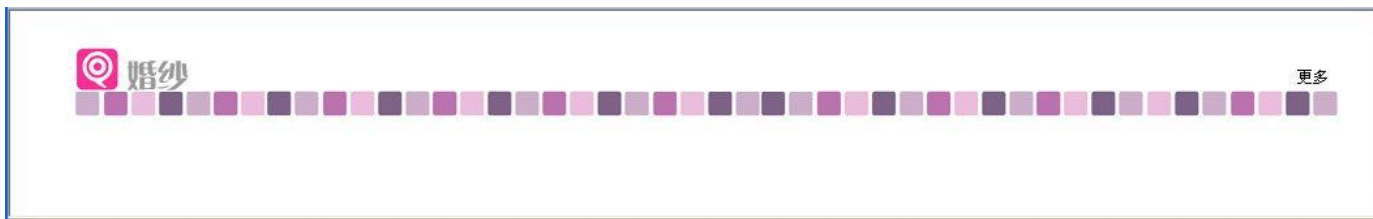
左右两侧Div宽度值为固定的200px，中间的Div设置margin属性为“0 200px”，表示上下间距为0，左右间距为两侧宽度200px。并通过设置left、right Div的“float”属性值分别为left、right，实现左右固定，中间宽度自适应的版式。

## 10.3 案例2 定位在网页布局中的应用

---

🕒 学习目标 综合应用Div标签的盒子模型、position属性、float属性、z-index属性进行网页布局。

🔊 知识要点 盒子模型，position、float、z-index等Div标签的定位属性。



请在Dreamweaver中完成案例的制作。

## 10.3 案例2 定位在网页布局中的应用

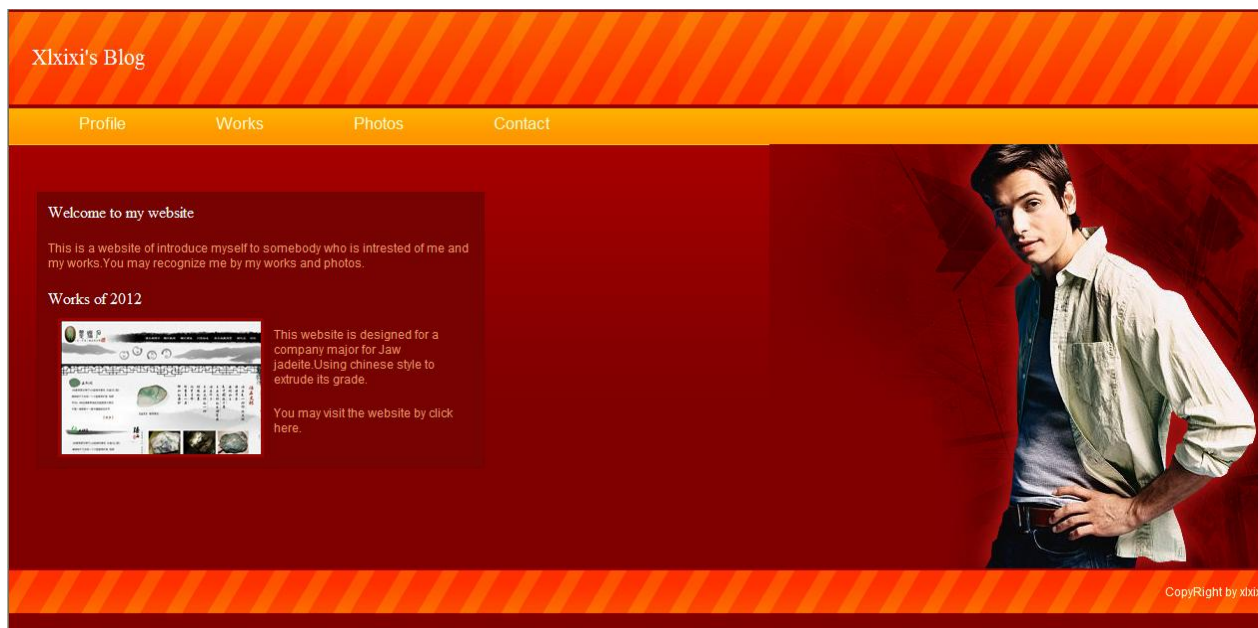
---

✦ 案例小结 通过网页布局实例，初次见证了CSS应用于网页布局的强大功能。使用盒子模型定义容器的外观，使用`position`和`float`定位容器的位置。由案例可见，不仅仅是本章所介绍的`<div>`是容器，`<a>`、`<img>`、`<h>`还有案例未涉及的`<p>`、`<ul>`、`<span>`等标签都可以被CSS作为容器使用。

## 10.4 案例3 使用CSS+Div布局个人Blog

🕒 学习目标 使用CSS+Div布局个人Blog。

🔊 知识要点 使用CSS对Div标签、列表标签、p标签、h标签等进行定位。



请在Dreamweaver中完成案例的制作。

## 10.4 案例3 使用CSS+Div布局个人Blog

---

 案例小结 本案例介绍使用CSS+Div进行网页布局的整体流程，主要包括使用列表标签设置导航条，使用float属性定位网页元素，元素背景图像的使用。

## 10.5 实训

---

本节重点练习**Div**标签在网页布局的使用，网页定位的方法，网页布局技巧。

## 10.5 实训

---

### 10.5.1 网页定位与布局

#### 1. 实训目的

掌握盒子模型在网页布局中的应用；

会使用position、float属性定位网页元素。

#### 2. 实训要求

继续案例2，在<h2>标签后添加<ul>标签，在列表项中加入图像和文字，并设置样式。插入图像“最新作品”，并设置其“position”属性为绝对定位。

## 10.5 实训

---

### 10.5.1 网页定位与布局



## 10.5 实训

---

### 10.5.2 使用Div+CSS布局页面

#### 1. 实训目的

掌握Div标签在网页定位中的操作步骤；

领会CSS布局技巧；

能够使用常用布局版式进行网页布局。

#### 2. 实训要求

使用Div+CSS布局学院教学成果申报网站，布局版式采用固定宽度且居中。网页使用Div标签划分为若干模块，并使用CSS设置样式。

# 10.5 实训

## 10.5.2 使用Div+CSS布局页面



---

# 谢谢！