

# Research on Web Page Layout Method Based on CSS + Div

Kui Gao, Weiyan Li, Xiaocui Fu

School of Information Science and Engineering, Shandong Agricultural University, Taian Shandong  
Email: gaokui@sdau.edu.cn, liweiyan@sdau.edu.cn, fxckey@sdau.edu.cn

Received: Feb. 16<sup>th</sup>, 2019; accepted: Mar. 4<sup>th</sup>, 2019; published: Mar. 11<sup>th</sup>, 2019

---

## Abstract

The CSS + Div layout method has gained more and more applications in the actual website design and production, and can be used through <div> Labels and CSS property settings implement web page layout. Through the example of the CSS + Div layout method and the analysis and explanation, this paper gives the usage method and attention to use the CSS + Div layout method to achieve the purpose of mastering.

## Keywords

CSS + Div, Margin, Padding, Border, Position, Float, Clear

---

# 基于CSS + Div的网页布局方法研究

高 葵, 李蔚妍, 付晓翠

山东农业大学, 信息科学与工程学院, 山东 泰安  
Email: gaokui@sdau.edu.cn, liweiyan@sdau.edu.cn, fxckey@sdau.edu.cn

收稿日期: 2019年2月16日; 录用日期: 2019年3月4日; 发布日期: 2019年3月11日

---

## 摘 要

CSS + Div布局方法在实际网站设计制作中得到了越来越多的应用, 可以通过<div>标签和CSS属性设置实现网页布局。本文通过CSS + Div布局方法举例并进行分析说明, 给出使用CSS + Div布局方法进行布局的使用方法和注意事项, 达到掌握的目的。

## 关键词

CSS + Div, 外边距, 内边距, 边框, Position, Float, Clear

---



## 1. 引言

在网页设计中, CSS 即层叠样式表, 是用来控制网页样式并允许样式与内容分离的一种标记性语言。Div 标签用来定义网页内容中的逻辑区域, 是一个块级元素, 是一个容器, 可以容纳任何网页元素。使用 CSS + Div 技术可以实现网页布局。与使用表格布局方法相比, CSS + Div 布局方法具有结构简洁、定位灵活、代码效率高等优点, 因此该技术在网站设计制作中得到了越来越多的应用, 同时也成为网站制作者的必备技术。但在实际应用中由于对相关概念和属性掌握不透彻, 导致在网页布局中出现异常, 本文通过 CSS + Div 布局方法举例并进行分析说明, 给出使用 CSS + Div 布局方法进行布局的使用方法和注意事项, 达到掌握的目的。

## 2. 使用举例

### 2.1. 相关概念

#### 1) 盒子模型

盒子模型是 CSS + Div 网页布局的重要概念, 网页元素和它周围空间所形成的矩形区域称为盒子(box) [1], 根据需要将网页中的盒子进行排列和分布, 就形成了网页布局。

盒子模型将页面中的每个元素看作一个矩形框, 这个框由元素的内容、内边距(padding)、边框(border)和外边距(margin)组成。按 top、bottom、left、right 的四个方向, 分别进行定义和设置, 描述盒子属性。具体设置如下图 1 所示。

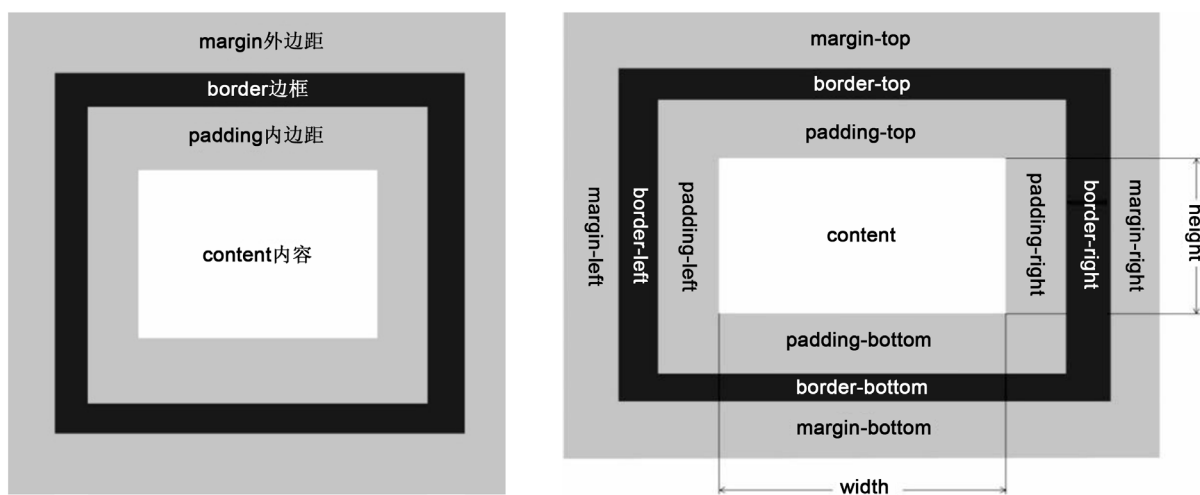


Figure 1. Box model structure and properties

图 1. 盒子模型结构和属性

#### 2) <div>标签

<div>标签是盒子模型的主要载体[2], 具有分割网页的功能。<div>标签是一个块状容器类标签, 在<div>和</div>之间可以容纳各种 HTML 元素, 构成一个独立的矩形区域。在网页中插入若干个<div>标签, 可以将网页分隔成若干个区域, 当还需要对该区域进一步分隔时, 可以在其中再插入<div>标签。无

论在页面中使用多少个标签，<div>标签之间仅存在并列关系和嵌套关系。

### 3) position 定位属性

在 CSS 样式中，position 定位属性定义元素区域的相对空间位置，可以相对于其上级元素，或相对于另一个元素，或相对于浏览器窗口，包括了四种属性值：static、relative、absolute 和 fixed，它们决定了元素区域的布局方式。

static 静态定位为默认值，网页元素遵循 HTML 的标准定位规则，即网页各种元素按照“前后相继”的顺序进行排列和分布[3]。

relative 相对定位，网页元素也遵循 HTML 的标准定位规则，但需要为网页元素相对于原始的标准位置设置一定的偏移距离，可以使用 CSS 的属性如 left、top、right、bottom 来设置位置。

absolute 绝对定位，位置将依据浏览器左上角的 0 点开始计算，使用 CSS 属性 left、top、right、bottom 来设置位置。

fixed 固定定位与绝对定位类似，是绝对定位的子类别，相对于浏览器窗口进行定位，当拖动浏览器窗口滚动条时，该元素位置始终保持不变，使用 CSS 属性 left、top、right、bottom 来设置位置。

### 4) 浮动方式

float 属性定义了元素浮动方向，它使元素进行左右浮动，可以产生多个网页元素并行排列的效果。float 属性有三个属性值：left 控制网页元素向左浮动，right 控制网页元素向右浮动，None 没有浮动。当设定<div>标签的 float 属性为 left 时，只要前驱<div>标签没有占满一行，其相继的<div>标签就向左浮动占据该行的空白位置[4]。

clear 属性与 float 属性配合使用，清除各种浮动。clear 属性有三个属性值：left 清除向左浮动，right 清除向右浮动，both 清除向左或向右浮动，None 不清除。当设定<div>标签的 clear 属性为 left 时，该标签清除向左浮动，重新回到前后相继的排列顺序中。

## 2.2. 布局方法

### 2.2.1. “上中下”布局

在“上中下”布局中，<div>标签按照前后相继的顺序排列，分割网页空间，不需要使用浮动方式，其大小和外观由 CSS 样式控制。

例题 1：完成页面，效果如下图 2。



Figure 2. Web layout 1

图 2. 网页布局 1

对应的主要代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN""http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Div+CSS 网页布局 1</title>
<style type="text/css">
#container {                               /*页面顶部容器的 CSS 规则*/
width: 400px;                               /*设置元素宽度*/
margin: 0 auto;                             /*设置元素自动居中对齐*/
}
#head {                                     /*页面 head 区域的 CSS 规则*/
background-color: #0F0;                     /*设置元素背景色*/
height: 20px;                               /*设置元素高度*/
}
#main {                                     /*页面 main 区域的 CSS 规则*/
background-color: #FF0;                     /*设置元素背景色*/
margin-top: 5px;                            /*设置元素上外边距*/
margin-bottom: 5px;                         /*设置元素下外边距*/
height: 80px;                               /*设置元素高度*/
}
#main2 {
background-color:#0FF;
margin-top: 5px;
margin-bottom: 5px;
height: 100px;
}
#foot {
background-color: #F0F;
height: 40px;
}
</style>
</head>
<body>
<div id="container">
<div id="head">此处显示 id "head" 的内容</div>
<div id="main">此处显示 id "main" 的内容</div>
<div id="main2">此处显示 id "main2" 的内容</div>
<div id="foot">此处显示 id "foot" 的内容</div>
</div>
</body>
</html>
```

分析：从上可以看到

1) 在 CSS + Div 布局中，一般地将所有<div>标签都嵌入到 ID 名称为 container 的<div>标签中，Height 属性为空，表示 container 标签高度可变。

2) 设置 container 标签的 margin 属性左右为 auto [5]，表示 container 标签及其嵌入的<div>标签在网页中居中对齐。

3) 若<div>标签没有设置宽度，边框和内外边距，则其宽度就与包含它的<div>标签 container 的宽度一致，如 ID 名称为 head 的<div>标签中 Width 属性为空，表示宽度为 400 px。

4) 要熟知盒子模型的各个属性，正确计算盒子的宽度和高度，实现精确布局。哪怕有 1 像素的宽度，也会对网页的精确布局产生影响，如下例题。

例题 2：若 main 区域的代码修改为如下：

```
#main {
background-color: #FF0;
width: 360px;          /*设置元素宽度*/
padding-left:60px;    /*设置元素左内边距，实现元素内容与左侧边框有 60 像素距离*/
margin-top: 5px;
margin-bottom: 5px;
height: 80px;
}
```

分析：ID 为 main 的<div>标签盒子的总宽度为 360 px + 60 px = 420 px，超出了 ID 为 container 的<div>标签的宽度 20 px，所以效果图出现了异常，main 区域部分右侧有 20 px 的距离突出，效果如下图 3。



Figure 3. Web layout 2

图 3. 网页布局 2

### 2.2.2. “左中右” 布局

在“左中右”布局中，首先插入若干个<div>标签并按照前后相继的顺序排列，然后设置 float 和 clear 属性，使<div>标签浮动，实现“左中右”布局，最后，设置 CSS 样式控制其大小和外观。

例题 3：在例题 1 的基础上完成页面，效果如下图 4。

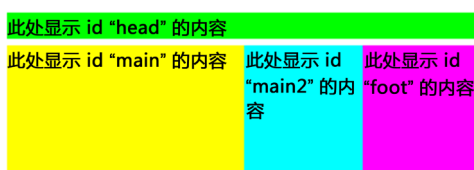


Figure 4. Web layout 3

图 4. 网页布局 3

与例题 1 相比，主要修改 main，main2，foot 区域部分，代码如下：

```
#main {
background-color: #FF0;
margin-top: 5px;
margin-bottom: 5px;
width:200px;      /*设置元素宽度为 200px*/
float:left;      /*设置元素向左浮动*/
height:100px;
}
#main2 {
background-color:#0FF;
margin-top: 5px;
margin-bottom: 5px;
height: 100px;
width:100px; /*设置元素宽度为 100px*/
float:left; /*设置元素向左浮动*/
}
#foot {
background-color: #F0F; /**/
margin-top: 5px;
height: 100px;
width:100px; /*设置元素宽度为 100px*/
float:left; /*设置元素向左浮动*/
}
```

分析：通过设置 main，main2，foot 区域部分的宽度，总和为  $200 + 100 + 100 = 400$ ，与所包含的 container 部分宽度相等，设置各个部分 float 属性为 left，保证这三个部分向左浮动并在一行中。

若只将 main2 区域的 float 属性设置为 right，则实现 main2 区域在该行向右浮动，效果如下图 5。当设置了 float 属性为 left 或 right 时，只要前驱<div>标签没有占满一行，其相继的<div>标签就向左浮动或向右浮动占据该行的空白位置。

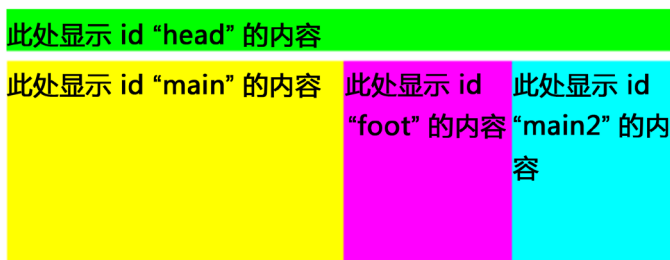


Figure 5. Web layout 4

图 5. 网页布局 4

例题 4：在例题 3 的基础上完成页面，效果如下图 6。



Figure 6. Web layout 5

图 6. 网页布局 5

与例题 3 相比，主要修改 main2，foot 区域部分，修改部分代码如下：

```
#main2 {
background-color:#0FF;
margin-top: 5px;
margin-bottom: 5px;
height: 100px;           /*此部分与例题 3 相同*/
width:200px;           /*设置元素宽度为 200px，使 main 和 main2 宽度之和为 400px*/
float:left;
}
#foot {
background-color: #F0F; /**/
margin-top: 5px;
height: 50px;
clear:left; /*设置 clear 属性，清除向左浮动*/
}
```

分析：设置 foot 区域的 clear 属性，取消浮动效果，使 foot 区域回到前后相继的正常排列状态，出现在下一行。此部分没有设置 foot 区域的宽度，默认与包含它的 container 区域宽度相同。

若想实现更复杂的布局，可以结合浮动和宽高属性设置实现。如修改 main2 部分的高度为 height: 60 px；修改 foot 部分的高度为 height: 40 px；保证这两部分的高度之和与 main 部分相等，依然设置 foot 部分浮动属性 float:left；，则完成后效果如下图 7。

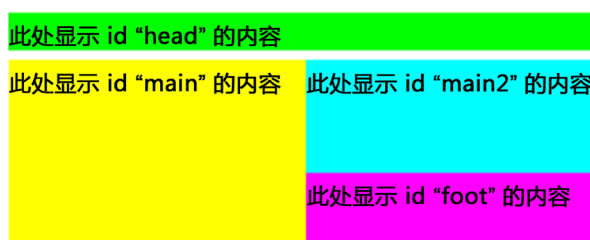


Figure 7. Web layout 6

图 7. 网页布局 6

### 3. 总结

通过以上的例题和分析可以得出如下结论：

1) 盒子模型是基础, 定义和规定了网页元素矩形区域的各种 CSS 属性。

2) <div>标签是盒子模型的主要载体, 具有分割网页的功能。在<div>和</div>之间可以容纳各种 HTML 元素, 是实现网页布局的基础元素。

3) CSS + Div 布局技术设计 CSS 样式的两个重要属性。position 属性决定了<div>标签的前后相继的排列顺序, float 属性决定了<div>标签在网页中的排列和定位。

#### 4. 结束语

本文通过例题对 CSS + Div 布局技术进行了举例和分析, 总结了使用规则, 提出了使用时的注意事项, 特别是通过对“上中下”和“左中右”两种基本布局形式的剖析和学习, 达到对 CSS + Div 布局技术的灵活应用。

#### 参考文献

- [1] 温谦. 网页设计 CSS 标准教程[M]. 北京: 人民邮电出版社, 2010: 88-90.
- [2] 刘瑞新. 网页设计与制作 html + css + javascript [M]. 北京: 机械工业出版社, 2013: 112-113.
- [3] 修毅. 网页设计与制作[M]. 北京: 人民邮电出版社, 2015: 138-139.
- [4] 王海波. 网页设计与制作——基于计算思维[M]. 北京: 机械工业出版社, 2014: 100-105.
- [5] 李毅. 网页制作与开发教程[M]. 北京: 人民邮电出版社, 2015: 193-196.

#### 知网检索的两种方式:

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>  
下拉列表框选择: [ISSN], 输入期刊 ISSN: 2161-8801, 即可查询
2. 打开知网首页 <http://cnki.net/>  
左侧“国际文献总库”进入, 输入文章标题, 即可查询

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: [csa@hanspub.org](mailto:csa@hanspub.org)