# 第2天 HTML+CSS

【学习目标】

## 1能够使用<form><input>等标签编写注册表单

a)掌握form标签的作用和action,method属性含义

b)掌握input标签type属性不同取值

c)掌握select标签和textarea标签

d)掌握各个标签上属性的用法

## 2熟悉CSS样式的3种导入方式

## 3熟悉CSS选择器的使用

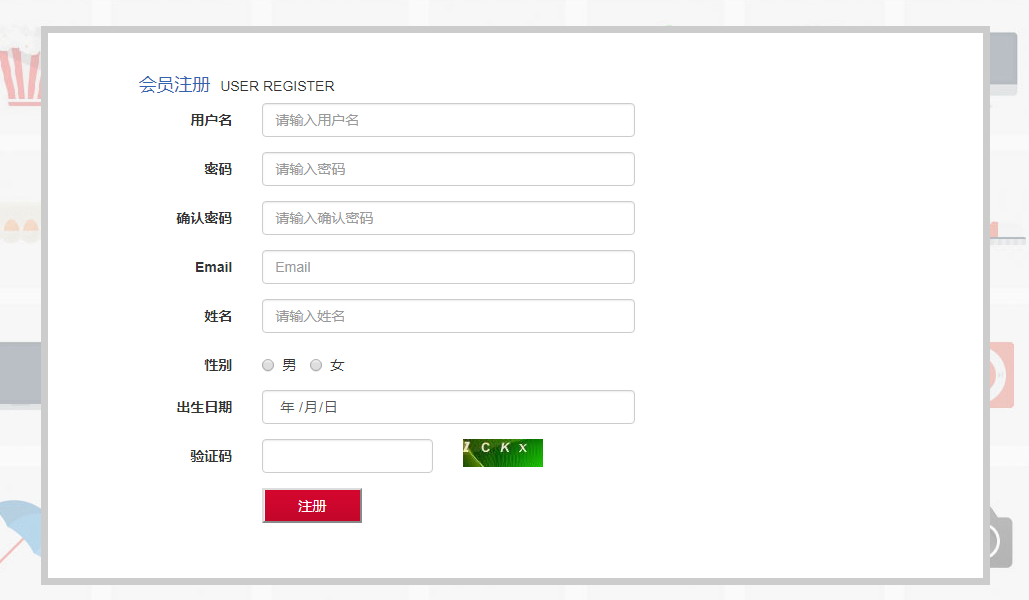
## 4掌握DIV页面布局

## 5掌握CSS盒子模型

# 网站用户注册页面显示

## 案例介绍

所有HTML标签中，表单标签是最重要的。在实际开发中，最经典的案例就是用户注册，几乎覆盖了表单标签的所有元素。效果如下：



## 案例相关标签

* + 1. 表单标签：<form>

描述：<form>表单标签，在html页面创建一个表单，表单标签在浏览器上没有任何显示。如果数据需要提交到服务器，则负责搜集数据的标签必须存放在表单标签中。

属性：

1)action属性：请求路径，确定表单提交到服务器的地址（路径）。

2)method属性：请求方式。常用取值：GET、POST。

GET：默认值

特点：提交的数据追加在请求路径上。例如：/1.html?username=jack&password=1234，数据格式为key/value对，追加是使用?连接，之后每一对数据之间使用&连接。

因为请求路径长度有限，所以GET方式提交的请求数据有限定。

因为在地址栏中显示数据，所以数据安全性不高。

POST：

特点：提交的数据不在请求路径上追加（即不显示在地址栏中），安全性更好，提交的数据大小无限定。

* + 1. 输入域标签：<input>

<input>标签一般用于获取用户输入信息，type属性值不同，则标签展现形式和搜集的信息也不同，是在页面中常用的标签。

1.type属性：

text：单行文本框。用户可在其中输入文本。默认宽度为20个字符；

password：密码框。在该控件中输入内容不明文显示，以黑圆显示。

radio：单选按钮。表示一组互斥选项按钮中的一个。当一个按钮被选中时，该组中之前选中的按钮变为非选中状态。设置name属性 name 代表一组控件

submit：提交按钮。提交按钮会把表单数据发送到服务器。一般不写name属性，否则将“提交”两个字提交到服务器。

因为不同项目注册需要的字段不同，需要完成的案例中没有覆盖所有的表单元素，以下标签的使用也需要掌握。

checkbox：复选框。用法与radio基本一致，在一组复选框中可以选任意多个选项。

file：文件上传组件，提供“浏览”按钮，可以选择所需要上传的文件。

hidden：隐藏域。数据会发送给服务器，页面不进行显示。

reset：重置按钮。将表单恢复到初始状态。

image：图形提交按钮，通过src给按钮设置图片。

button：普通按钮。通常结合Javascript实现动作效果。

2.name属性：元素名。如果需要将表单数据提交到服务器，则必须提供name属性值，服务器通过该属性值获取提交的数据。

3.value属性：设置input标签的默认值。submit和reset按钮的value为设置按钮显示文字。

4.size：元素的大小

5.checked：设置单选按钮或者复选框被选中

6.readonly：设置元素只读，不能输入

7.disabled：是否可用。加该属性后，元素不可用

8.maxlength：允许输入的最大长度

* + 1. 下拉列表标签：<select> + <option>

1.作用：展现下拉列表，可以进行单选或者多选。需要结合字标签<option>指定每一项列表项。其中标签上的属性如下：

name：用于将数据发送给服务器时的参数名称

multiple：不写该属性，则默认是单选，若取值为”multiple”表示多选

size：多选时，可见选项的数目

子标签<option>：下拉列表中的每一个选项，属性如下：

selected：勾选当前列表项

value:发送给服务器端的选项值

* + 1. 文本域标签：<textarea>

多行文本框，可以输入任意多的文本信息。属性如下：

cols：文本域的列数

rows：文本域的行数

* + 1. 按钮标签<button>(了解)

<input type=”button/reset/submit”> 按钮标签一般很少使用，提供“普通按钮|重置|提交”功能。不同浏览器默认值不同。

## 案例分析

* + 1. 知识点分析

为了结构更好的显示表单。本案例需要使用table表格布局。需要提供2列来显示数据，其余行进行单元格合并即可。最终需要创建10行2列的表格。



## 案例实现

* + 1. 布局

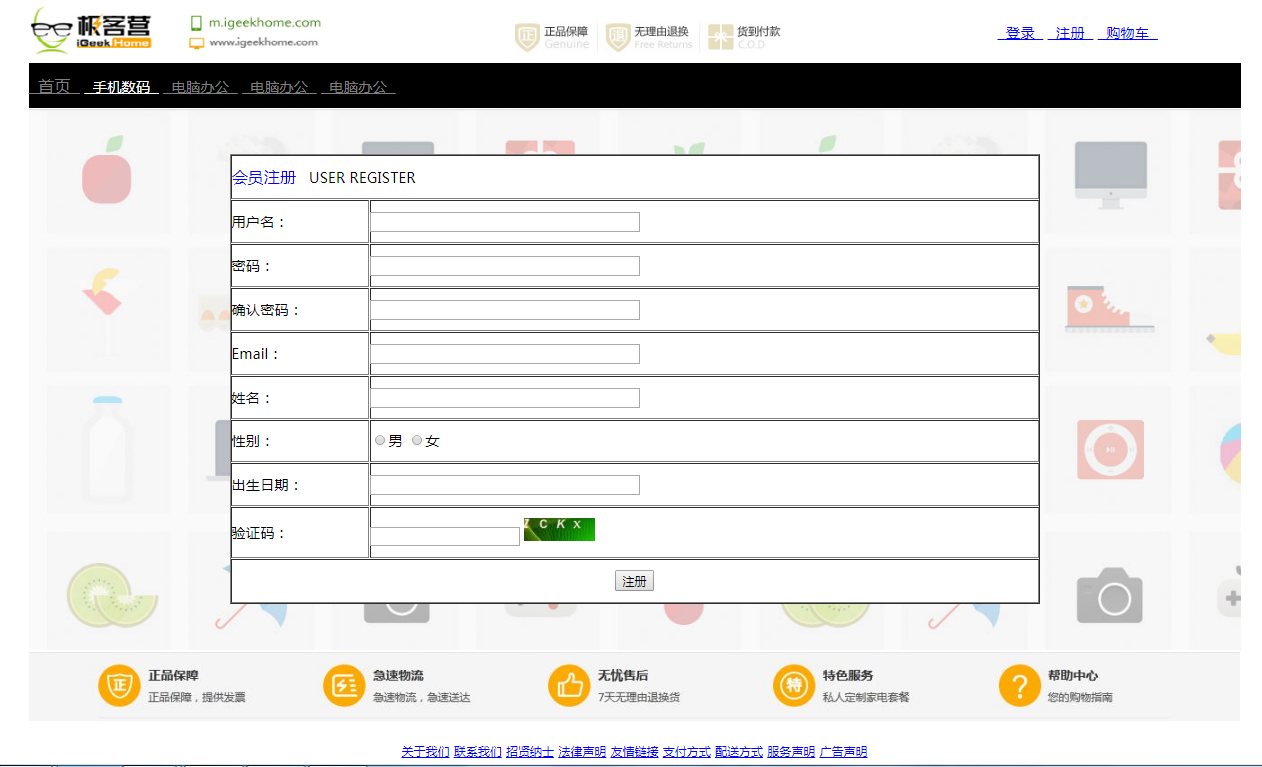
整体页面也使用表格布局，需要五行一列表格。在第三行插入一个十行两列的表格。布局代码如下：

|  |
| --- |
| <!--五行一列表格-->  <table border="0" width="1300px" align="center">  <!--logo-->  <tr>  <td height="58">  </td>  </tr>  <!--第二行：导航栏-->  <tr height="50">  <td bgcolor="black">  </td>  </tr>  <!--第三行：注册表单-->  <tr>  <td height="600" background="../img/regist\_bg.jpg">  <!--嵌入一个10行2列表格-->  <table width="900" height="500" border="1" align="center" bgcolor="white">  <tr>  <td colspan="2"></td>  </tr>  <tr>  <td width="150"> </td>  <td></td>  </tr>  <tr>  <td> </td>  <td></td>  </tr>  <tr>  <td> </td>  <td></td>  </tr>  <tr>  <td> </td>  <td></td>  </tr>  <tr>  <td> </td>  <td></td>  </tr>  <tr>  <td> </td>  <td></td>  </tr>  <tr>  <td> </td>  <td></td>  </tr>  <tr>  <td> </td>  <td></td>  </tr>  <tr align="center">  <td colspan="2"></td>  </tr>  </table>  </td>  </tr>  <!--第四行：广告-->  <tr>  <td>  <img src="../img/footer.jpg">  </td>  </tr>  <!--第五行：友情链接-->  <tr align="center">  <td>  </td>  </tr>  </table> |

* + 1. 表单

|  |
| --- |
| <form action="#" method="get">  <!--嵌入一个10行2列表格-->  <table width="900" height="500" border="1" align="center" bgcolor="white">  <tr>  <td colspan="2">  <font color="blue" size="4">会员注册</font>&nbsp;&nbsp;&nbsp;USER REGISTER  </td>    </tr>  <tr>  <td width="150">用户名：</td>  <td><input type="text" name="username" size="40"></td>  </tr>  <tr>  <td>密码：</td>  <td><input type="password" name="userpass" size="40"></td>  </tr>  <tr>  <td>确认密码：</td>  <td><input type="password" name="repassword" size="40"></td>  </tr>  <tr>  <td>Email：</td>  <td><input type="text" name="email" size="40" ></td>  </tr>  <tr>  <td>姓名：</td>  <td><input type="text" name="realname" size="40"></td>  </tr>  <tr>  <td>性别：</td>  <td>  <input type="radio" name="sex" value="男">男  <input type="radio" name="sex" value="女">女  </td>  </tr>  <tr>  <td>出生日期：</td>  <td>  <input type="text" name="birthday" size="40">  </td>  </tr>  <tr>  <td>验证码：</td>  <td>  <input type="text" name="checkcode">  <img src="../img/yanzhengma.png">  </td>  </tr>  <tr align="center">  <td colspan="2"><input type="submit" value="注册"></td>  </tr>  </table>  </form> |

页面实现效果如下：



# 使用DIV+CSS重写网站首页案例

## 需求分析

上次课中我们使用了table表格对网站首页进行了布局。但是表格布局有自身的缺陷，不灵活。在此，我们采用UI工程师经常使用的DIV+CSS方式进行布局，这种布局方式更为灵活。

## 相关知识点

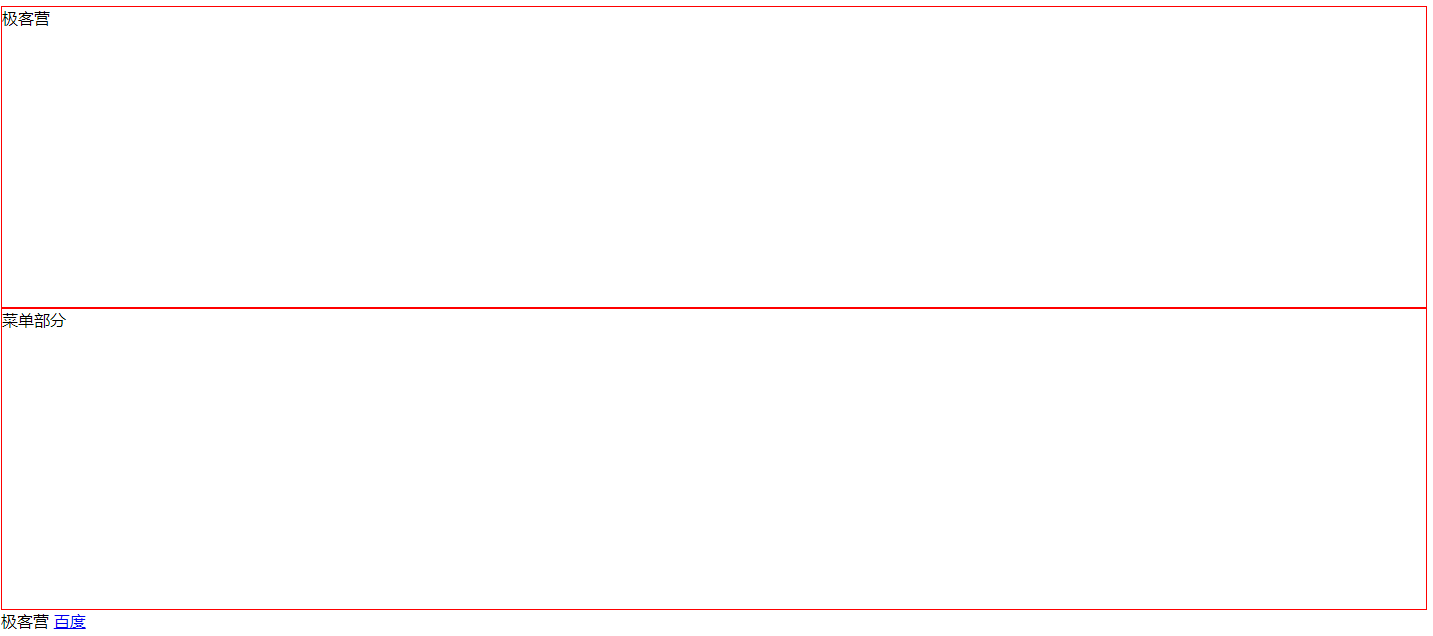
2.2.1什么是 DIV？

div就是html的一个普通标签，主要用于区域的划分。div特性：独占一行。独自使用该标签一般不能实现复杂效果，必须结合CSS样式进行渲染。

div是块级元素，我们所学的html基本标签中还有p，h1-h6，li等都是块级元素。块级元素的特点是：独占一行，可以通过CSS样式设置宽度和高度。

|  |
| --- |
| <style>  div{  border: 1px solid red;  width: 100%;  height: 300px;  }  </style>  <body>  <div>  极客营  </div>  <div>  菜单部分  </div>    极客营 <a href="#">百度</a>  </body> |

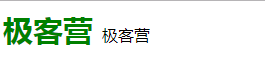
页面效果：



span标签：一般结合DIV使用，对页面局部文字进行特殊设定。也必须使用CSS设定样式，否则单独使用无任何含义。

|  |
| --- |
| <style>  span{  font-size: 30px;  color: green;  font-weight: bold;  }  </style>  <body>  <span>  极客营  </span>  极客营  </body> |

页面效果：



2.2.2 CSS概述

1.CSS是什么

CSS通常称为CSS样式或层叠样式表，主要 用于设置HTML页面中文本内容(字体，大小，对齐方式等)、图片外观(高度、边框样式、边距等) 以及版面的布局等外观显示样式。

CSS可以使HTML页面更美观，CSS色系的搭配可以让用户更舒服，CSS+DIV布局更加灵活，更容易绘制出用户需要的结构。

2.CSS名词解释

CSS(Cascading Style Sheets)：层叠样式表

样式：给html标签添加需要显示的效果。

层叠：使用不同的添加方式，给用一个html标签添加样式，最后所有样式叠加在一起，共同作用于该标签。

2.2.3 CSS样式规则

使用HTML时，需要遵从一定的规范。CSS亦是如此，若想熟练使用CSS对网页进行修饰，首先需要了解CSS样式规则。具体格式如下：

选择器{

属性1:属性值1;

属性2:属性值2….

}

在上面的样式规则中，”选择器”用于指定CSS样式作用的HTML对象，花括号内是对该对象设置的具体样式。属性和属性值以键值对的方式出现，使用英文“:”分割。多个属性之间使用用为“;”分隔。例如：

|  |
| --- |
| <style>  h2{  color:red;  font-size:18px;  }  </style> |

初学者在书写CSS样式时，除了要遵循CSS规则，还需要注意CSS代码结构中的几个特点：

1）CSS样式”选择器严格区分大小写”，属性和属性值不区分大小写

2）多个属性之间必须用应为状态下的分号隔开，最后一个属性后的分号可以省略，但是，为了便于增加新样式最好保留。

3）如果属性的值是由多个单词组成且中间包含空格，则必须为这个属性加上英文状态下的引号。例如：

p{ font-family:”Time New Roman”;}

4）在编写CSS代码时，为了提高代码的可读性，通常会加上CSS注释，例如：

/\*这是CSS注释文本，此文本不会显示 在浏览器窗口中\*/

5）在CSS代码中空格是不被解析的，花括号以及分号前后的空格可有可无。因此，可以使用空格键、Tab键、回车键等对样式进行排版，即所谓的格式化CSS代码，这样可以提高代码的可读性。例如：

|  |
| --- |
| h1{font-size:20px;color:red} |

|  |
| --- |
| h1{  font-size:20px; /\*定义字体大小\*/  color:red /\*定义字体颜色\*/  } |

上述两端代码所呈现效果是一样的，但是，第二种写法的可读性更高。需要注意的是，属性值和单位之间是不允许出现空格的，否则浏览器解析会出错。例如下面这段代码是不正确的。

|  |
| --- |
| h1{font-size:20 px; } /\*20和单位px之间有空格\*/ |

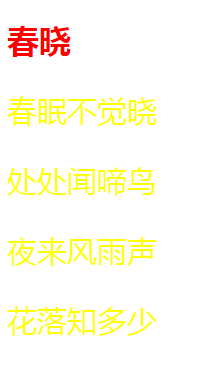
2.2.4.CSS选择器

1）标签选择器

标签选择器作用于该页面中的所有相同标签元素。

|  |
| --- |
| <style>  /\*样式表：  选择器：1.标签选择器  基本样式：1.文本样式  \* \*/  p{  font-size: 30px;  color: yellow;  }  h1{  color: red;  }  </style>  </head>  <body>  <h1>春晓</h1>  <p>春眠不觉晓</p>  <p>处处闻啼鸟</p>  <p>夜来风雨声</p>  <p>花落知多少</p>  </body> |

效果如下：



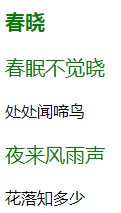
2）类选择器

类选择器主要作用于同一类名的元素

类选择器使用分为两步：第一步：在页面元素上添加class属性，并取值。第二步：在样式表中通过.class名称编写样式表。

|  |
| --- |
| <style>  .mystyle{  font-size: 20px;  color: green;  }  </style>  </head>  <body>  <h1 class="mystyle">春晓</h1>  <p class="mystyle">春眠不觉晓</p>  <p>处处闻啼鸟</p>  <p class="mystyle">夜来风雨声</p>  <p>花落知多少</p>  </body> |

效果如下：

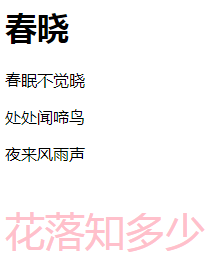


3）ID选择器

ID选择器一般用于页面某个元素与其他元素样式都不相同时使用。使用步骤与类选择器类似。第一步：在元素上添加id属性，并取值。第二步：在样式表中使用#id名称编写样式表。

|  |
| --- |
| <style>  #onlyStyle{  font-size: 40px;  color: pink;  }  </style>  </head>  <body>  <h1 class="mystyle">春晓</h1>  <p class="mystyle">春眠不觉晓</p>  <p>处处闻啼鸟</p>  <p class="mystyle">夜来风雨声</p>  <p id="onlyStyle">花落知多少</p>  </body> |

效果图如下：



4)层级选择器

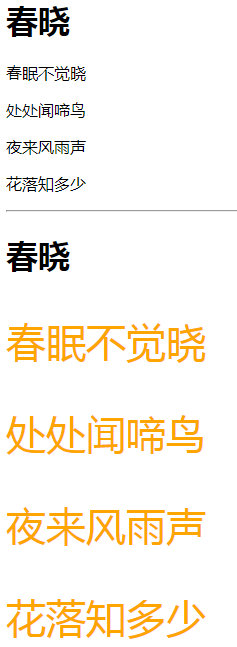
语法格式：选择器1 选择器2{样式表}

含义：过滤页面中选择器1下面的符合选择器2的任何层级子元素。

例子：

|  |
| --- |
| <html>  <head>  <meta charset="UTF-8">  <title>层级选择器</title>  <style>  div p{  查找页面中div中所有p标签的子元素,为其设置样式  font-size: 40px;  color: orange;  }  </style>  </head>  <body>  <h1>春晓</h1>  <p>春眠不觉晓</p>  <p>处处闻啼鸟</p>  <p>夜来风雨声</p>  <p>花落知多少</p>  <hr />  <div>  <h1>春晓</h1>  <p>春眠不觉晓</p>  <p>处处闻啼鸟</p>  <p>夜来风雨声</p>  <p>花落知多少</p>  </div>  </body>  </html> |

效果:



通过例子可以说明,层级选择器可以对页面中某些元素进行过滤，实现单独设置的好处。

5）属性选择器

属性选择器通过过滤标签上的属性，实现过滤功能。

例子：过滤页面中所有input标签，并且type属性值为text的元素，为其设置样式

|  |
| --- |
| <style>  input[type='text']{  background-color: grey;  }  </style>  </head>  <body>  用户名:<input type="text"><br />  密码：<input type="password" />  </body> |

效果：



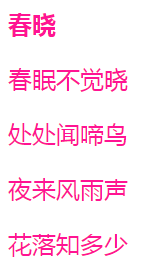
6）组合选择器

组合选择器主要作用是当很多选择器拥有自己的样式，同时又都具有相同的样式时，可以将多个选择器组合在一起应用相同的样式表。

例子：

|  |
| --- |
| <style>  h1,p{  font-size: 24px;  color: deeppink;  }  </style>  </head>  <body>  <h1>春晓</h1>  <p>春眠不觉晓</p>  <p>处处闻啼鸟</p>  <p>夜来风雨声</p>  <p>花落知多少</p>  </body> |

效果：



2.2.5引入CSS样式

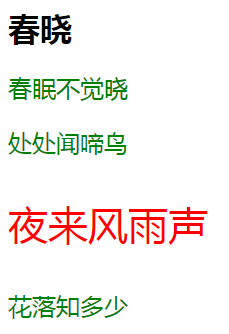
CSS使用非常灵活，既可以嵌入在HTML文档中，也可以是一个独立的文件。如果是单独的文件，则必须以.css为扩展名。CSS和HTML结合的3种常用方式：

1）行内样式

行内样式是通过标签的style属性来设置元素的样式的。

|  |
| --- |
| <style type="text/css">  p{  font-size: 24px;  color: green;  }  </style>  </head>  <body>  <h1>春晓</h1>  <p>春眠不觉晓</p>  <p>处处闻啼鸟</p>  <p style="font-size: 40px; color: red;">夜来风雨声</p>  <p>花落知多少</p>  </body> |

效果：



行内样式通过标签的属性来控制样式，这样并没有做到结构与表现（HTML展现结构，CSS显示效果）相分离，所以一般建议少使用。学习阶段有时候为了快速编程，偶尔使用。

2.内嵌样式

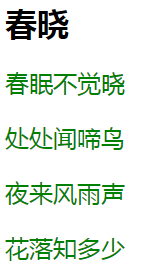
内嵌样式又称为内部样式，是将CSS代码集中写在HTML文档的<head>头部标签中，并且使用<style>标签定义。

给当前HTML文件中的多个标签设置样式。

在HTML的head标签中使用style标签定义CSS

|  |
| --- |
| <style type="text/css">  p{  font-size: 24px;  color: green;  }  </style>  </head>  <body>  <h1>春晓</h1>  <p>春眠不觉晓</p>  <p>处处闻啼鸟</p>  <p>夜来风雨声</p>  <p>花落知多少</p>  </body> |

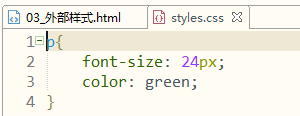
效果图：



内嵌CSS样式只对其所在的HTML页面有效，可以对多处标签统一设置样式。因此，仅设计一个页面时，使用内嵌样式是个不错的选择。但是如果是一个网站，不建议使用这种方式，因为不能充分发挥CSS代码的重用优势。

3.外部样式

外部样式又称为链入式，是将所有的样式放在一个或多个以.css为扩展名的外部样式表文件中，在HTML文件中使用<link rel="stylesheet" href="css文件地址">将外部样式引入。



在03\_外部样式.html中代码如下:

|  |
| --- |
| <html>  <head>  <meta charset="UTF-8">  <title>外部样式</title>  <!--引入外部样式表-->  引入外部样式文件,如需要可以引入多个CSS文件  <link rel="stylesheet" href="styles.css" />  </head>  <body>  <h1>春晓</h1>  <p>春眠不觉晓</p>  <p>处处闻啼鸟</p>  <p>夜来风雨声</p>  <p>花落知多少</p>  </body>  </html> |

效果与内嵌样式相同,此处图略.

2.2.6 CSS常用样式

1.边框和尺寸：border,width,height

border：设置边框的样式

格式：宽度 样式 颜色

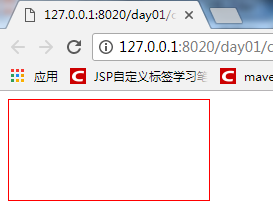
例如：style="border:1px solid #f00" 设置1像素实线红色边框。

样式可取值： solid实线,none无边框,double双线,dashed虚线等

width/height：设置元素的高度和宽度，注意：只有块级元素才可以设置该属性。

|  |
| --- |
| <style>  div{  width: 200px;  height: 100px;  border: 1px solid red;  }  </style>  <body>  <div></div>  </body> |

效果：



2.CSS布局：float、clear

通常默认的排版方式，将页面中的元素从上到下一一罗列，而实际开发中，需要左右方式进行排版，就需要使用浮动样式。

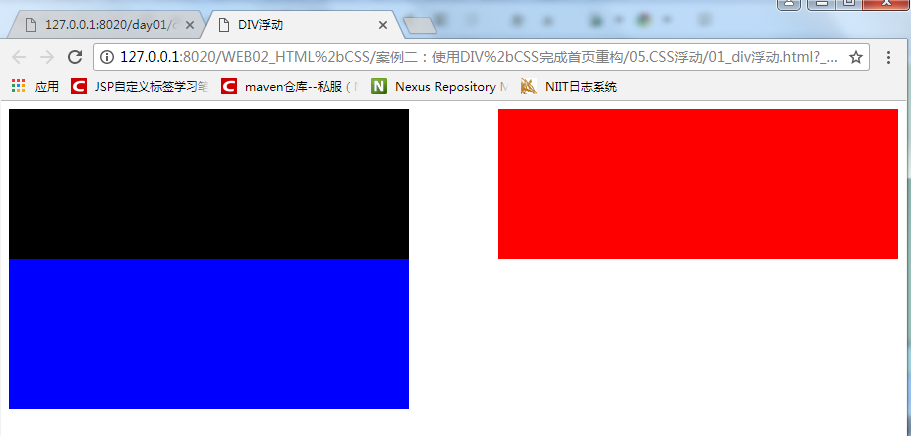
基本语法：选择器{float：属性值;}

常用的属性值：left：元素向左浮动 right：元素向右浮动 none：元素不浮动（默认值）

例子1：第一个DIV元素向右侧浮动

|  |
| --- |
| <style>  #one{  background-color: red;  width: 400px;  height: 150px;  float: right;  }  #two{  background-color: black;  width: 400px;  height: 150px;    }  #three{  background-color: blue;  width: 400px;  height: 150px;  }    </style>  </head>  <body>  <div id="one"></div>  <div id="two"></div>  <div id="three"></div>  </body> |

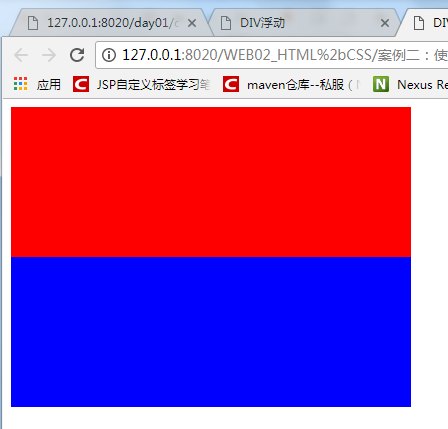
效果：



例子2：第一个DIV元素向左侧浮动

|  |
| --- |
| <style>  #one{  background-color: red;  width: 400px;  height: 150px;  float: left;  }  #two{  background-color: black;  width: 400px;  height: 150px;    }  #three{  background-color: blue;  width: 400px;  height: 150px;  }    </style>  </head>  <body>  <div id="one"></div>  <div id="two"></div>  <div id="three"></div>  </body> |

效果:将第二个DIV完全覆盖。主要是由于第一个DIV浮动之后，脱离文档流，第二个DIV成为正常文档流中的第一个元素，因此两者重叠。



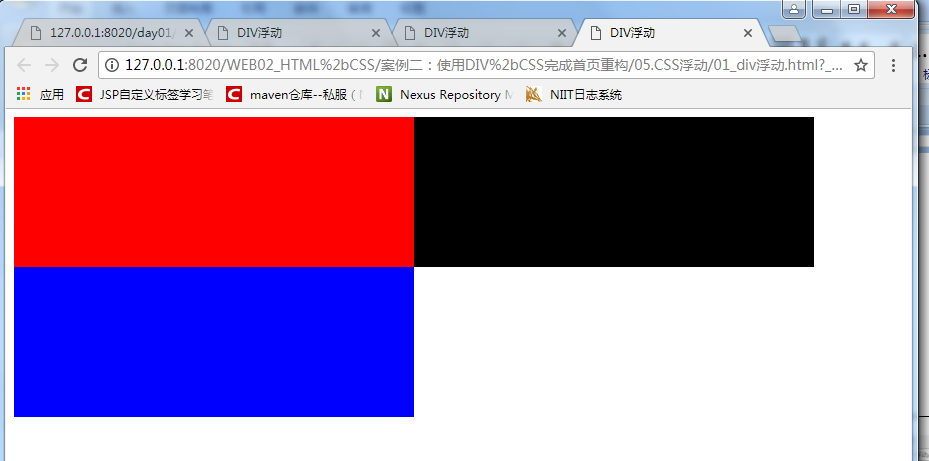
例子3：三个DIV都向左侧浮动

|  |
| --- |
| <style>  #one{  background-color: red;  width: 400px;  height: 150px;  float: left;  }  #two{  background-color: black;  width: 400px;  height: 150px;  float: left;  }  #three{  background-color: blue;  width: 400px;  height: 150px;  float: left;  }    </style>  </head>  <body>  <div id="one"></div>  <div id="two"></div>  <div id="three"></div>  </body> |

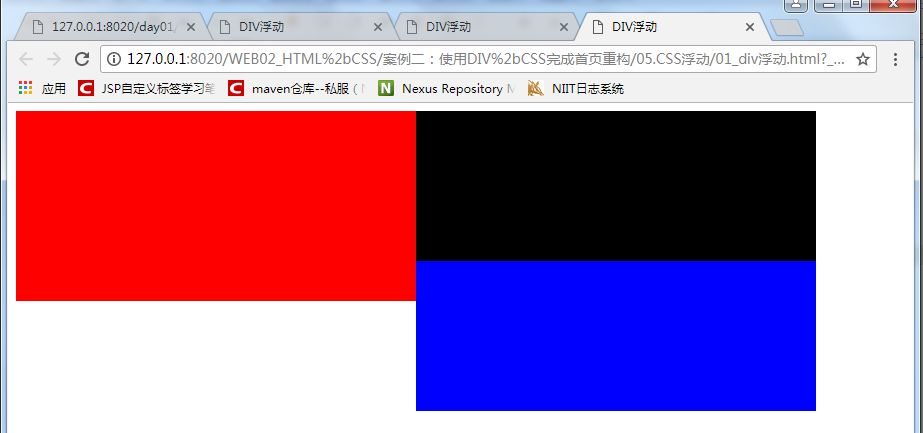
效果：



注意：如果三个DIV宽度和已经超出外部元素（当前是浏览器）的宽度，则后面的元素无法浮动上去，会出现下图所示效果：



如果第一个DIV高度设置高一些，则会出现“卡住”的现象。如图所示：



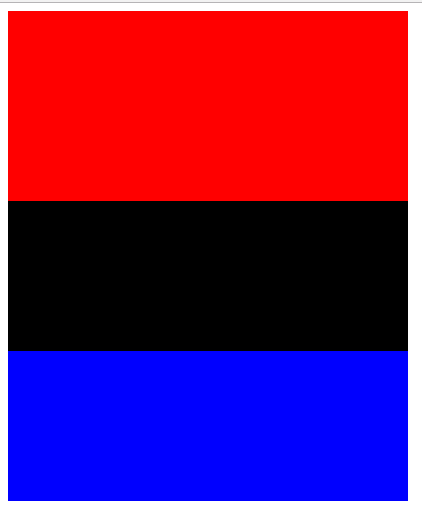
由于浮动元素不再占用原文档流的位置，所以它会对页面中其他元素的排版产生影响。如果要避免影响，需要使用clear属性进行清除浮动。

在上面浮动例子2中，第一个DIV左侧浮动，其他两个DIV不设置浮动，则第一个DIV会将第二个DIV完全覆盖。如若想第二个DIV正常放在第一个DIV下面，则可以使用清浮动样式。

|  |
| --- |
| <style>  #one{  background-color: red;  width: 400px;  height: 190px;  float: left;  }  #two{  background-color: black;  width: 400px;  height: 150px;    }  #three{  background-color: blue;  width: 400px;  height: 150px;    }  .clear{  /\*清除两边浮动\*/  clear: both;  }  </style>  </head>  <body>  <div id="one"></div>  <div class="clear"></div>  <div id="two"></div>  <div id="three"></div>  </body> |

在浮动元素下面新增DIV，主要作用就是清除浮动元素，并且不占用页面位置。

效果：



3.类型转换：display

HTML提供丰富的标签，根据标签展现的不同，一般分为两类：块级元素标签和行内元素标签。

块级元素：以区域块方式展现。每个元素独占一行或者多行。

常见的块级元素：<h1>-<h6>,<div>,<p>,<ul>等。

行内元素：不独自占一行，与其他行内元素在一行中显示。

常见的行内元素：<a>,<span>,<img>等

在开发中，希望行内元素具有块级元素的特性，或者希望块级元素具有行内元素的特性，则都是通过display进行转换。

取值：inline:将元素将显示为行内元素

block将元素将以块级元素展现

inline-block:将元素显示为内联元素,但对象的内容作为块级元素呈现.

none:将元素隐藏,并且不占用页面空间。

4.字体：color,font-size

color：设置字体颜色

font-size：设置字体大小

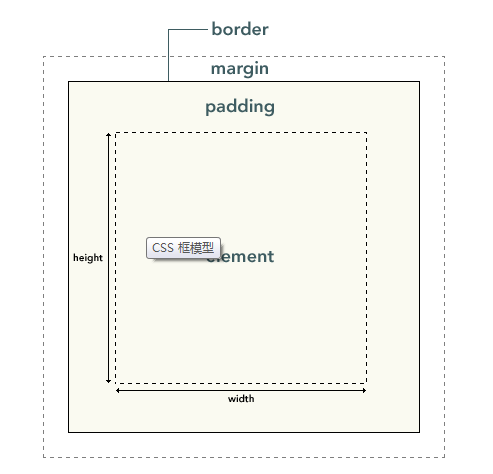
text-align：设置内容对齐方式，取值left 左对齐, right右对齐, center居中对齐

line-height:设置行高，一般用于文本垂直居中。

5.背景色background-color

2.2.7 盒子模型

CSS框模型(Box Model)规定了元素框处理元素内容、内边距、外边距和边距的方式。



元素框的最内部分是实际的内容，直接包围内容的是内边距。内边距呈现了元素的背景。内边距的边缘是边框。边框以外是外边距，外边距默认是透明的，因此不会遮挡其后的任何元素。

内边距、边框和外边距都是可选的，默认值是零。但是，许多元素将由用户代理样式表设置外边距和内边距。可以通过将元素的 margin 和 padding 设置为零来覆盖这些浏览器样式。这可以分别进行，也可以使用通用选择器对所有元素进行设置：

\* {

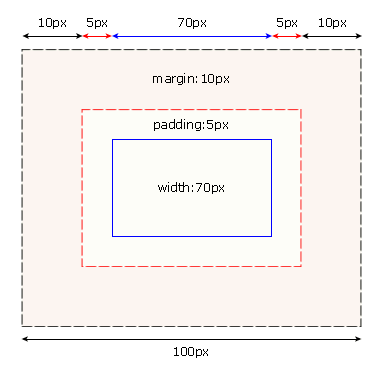
margin: 0px;

padding: 0px;

}

在 CSS 中，width 和 height 指的是内容区域的宽度和高度。增加内边距、边框和外边距不会影响内容区域的尺寸，但是会增加元素框的尺寸。

假设框的每个边上有 10 个像素的外边距和 5 个像素的内边距。如果希望这个元素框达到 100 个像素，就需要将内容的宽度设置为 70 像素，请看下图：



#box {

width: 70px;

margin: 10px;

padding: 5px;

}

提示：内边距、边框和外边距可以应用于一个元素的所有边，也可以应用于单独的边。

提示：外边距可以是负值，而且在很多情况下都要使用负值的外边距。

1.内边距:padding

内边距代表边框与内容之间的填充距离.

可以单独设置某一边的内边距,如设置padding-top/padding-right/padding-bottom/padding-left。也可以使用一条语句完成设置，语法格式如下：

padding：值1;

如果取值为一个值,则 该值代表上、右、下、左四个方向内边距都为该值

如果取值为两个值，则第一个值代表上/下方向取值，第二个值代表左右方向取值

如果取值为三个值，则第一个值代表上，第二个值代表左右，第三个值代表下

如果取值为四个值，则四个值分别代表上、右、下、左四个方向的内边距值。

注意：设置内边距会将边框大小撑大。

2.外边距：margin

margin代表边框与外部元素之间的距离，使用方式与padding类似。

3.边框：border

border为元素设置边框。

## 首页重构案例布局分析

分析首页布局，使用一个DIV作为盒子，将页面全部内容包含在内。并设置该盒子水平居中。

内部嵌套八个DIV，每个DIV作为一行。

第一行：logo部分。嵌套三个DIV，通过浮动设置水平排布

第二行：导航栏。使用UL+LI 并通过display样式设置水平排布

第三行：轮播广告。直接插入图片

第四行：最新商品。内部嵌入DIV，使布局如下图所示：



第五行：广告。直接插入图片

第六行：与第四行相同

第七行：广告。直接插入图片

第八行：友情链接。直接写入a,p标签，编辑文本。

## 代码实现

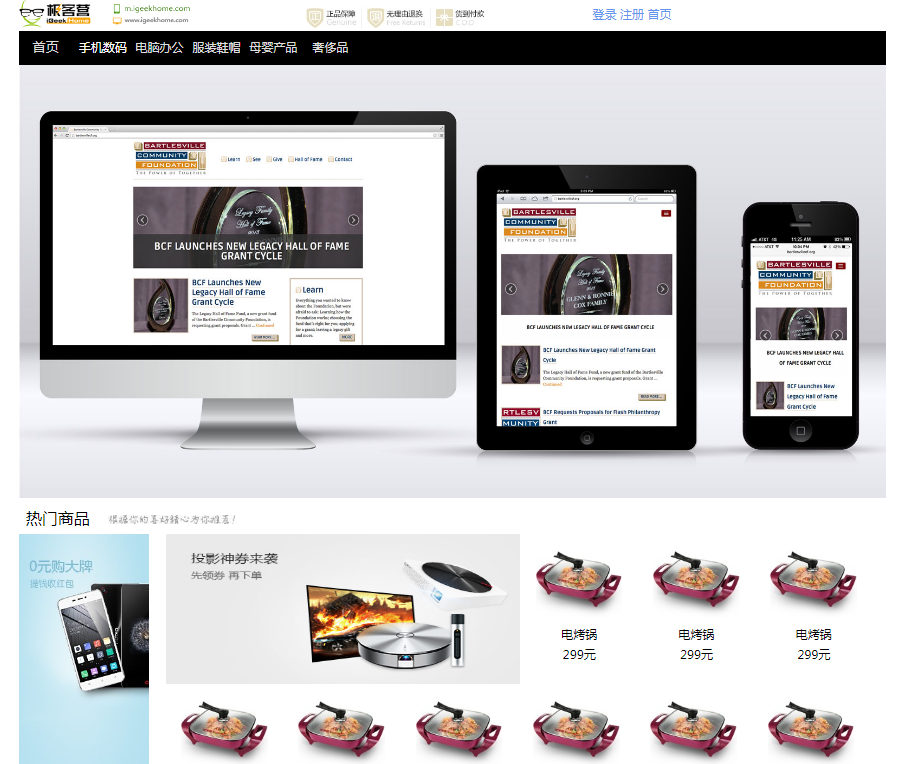
html页面骨架

|  |
| --- |
| <body>  <div id="container">  <!--logo部分-->  <div id="logo">  <div class="top">  <img src="../img/logo.png" height="46">  </div>  <div class="top">  <img src="../img/header.png" height="46">  </div>  <div class="top">  <a href="#">登录</a>  <a href="#">注册</a>  <a href="#">首页</a>  </div>  </div>  <div class="clear"></div>  <!--导航部分-->  <div id="menu">  <ul>  <a href="#"><li style="font-size: 20px;">首页</li></a>  <a href="#"><li style="color: white;">手机数码</li></a>  <a href="#"><li>电脑办公</li></a>  <a href="#"><li>服装鞋帽</li></a>  <a href="#"><li>母婴产品</li></a>  <a href="#"><li>奢侈品</li></a>  </ul>  </div>    <!--轮播-->  <div id="adv1">  <img src="../img/1.jpg" width="100%">  </div>    <!--热门商品-->  <div class="hot">  <div class="hot\_top">  <span>热门商品</span>&nbsp;&nbsp;&nbsp;&nbsp;  <img src="../img/title2.jpg">  </div>  <div class="hot\_bottom">  <div class="hot\_bottom\_left">  <img src="../img/big01.jpg" height="100%">  </div>  <!--右侧部分插入10个DIV-->  <div class="hot\_bottom\_right">  <div class="adv\_big"><img src="../img/middle01.jpg" width="100%" height="100%"></div>  <div class="adv\_small">  <img src="../img/small03.jpg">  <p>电烤锅</p>  <p>299元</p>  </div>  <div class="adv\_small">  <img src="../img/small03.jpg">  <p>电烤锅</p>  <p>299元</p>  </div>  <div class="adv\_small">  <img src="../img/small03.jpg">  <p>电烤锅</p>  <p>299元</p>  </div>  <div class="adv\_small">  <img src="../img/small03.jpg">  <p>电烤锅</p>  <p>299元</p>  </div>  <div class="adv\_small">  <img src="../img/small03.jpg">  <p>电烤锅</p>  <p>299元</p>  </div>  <div class="adv\_small">  <img src="../img/small03.jpg">  <p>电烤锅</p>  <p>299元</p>  </div>  <div class="adv\_small">  <img src="../img/small03.jpg">  <p>电烤锅</p>  <p>299元</p>  </div>  <div class="adv\_small">  <img src="../img/small03.jpg">  <p>电烤锅</p>  <p>299元</p>  </div>  <div class="adv\_small">  <img src="../img/small03.jpg">  <p>电烤锅</p>  <p>299元</p>  </div>  </div>  </div>  </div>    <!--广告-->  <div id="adv2">  <img src="../img/ad.jpg" width="100%">  </div>    <!--最新商品-->  <div class="hot">  <div class="hot\_top">  <span>最新商品</span>&nbsp;&nbsp;&nbsp;&nbsp;  <img src="../img/title2.jpg">  </div>  <div class="hot\_bottom">  <div class="hot\_bottom\_left">  <img src="../img/big01.jpg" height="100%">  </div>  <!--右侧部分插入10个DIV-->  <div class="hot\_bottom\_right">  <div class="adv\_big"><img src="../img/middle01.jpg" width="100%" height="100%"></div>  <div class="adv\_small">  <img src="../img/small03.jpg">  <p>电烤锅</p>  <p>299元</p>  </div>  <div class="adv\_small">  <img src="../img/small03.jpg">  <p>电烤锅</p>  <p>299元</p>  </div>  <div class="adv\_small">  <img src="../img/small03.jpg">  <p>电烤锅</p>  <p>299元</p>  </div>  <div class="adv\_small">  <img src="../img/small03.jpg">  <p>电烤锅</p>  <p>299元</p>  </div>  <div class="adv\_small">  <img src="../img/small03.jpg">  <p>电烤锅</p>  <p>299元</p>  </div>  <div class="adv\_small">  <img src="../img/small03.jpg">  <p>电烤锅</p>  <p>299元</p>  </div>  <div class="adv\_small">  <img src="../img/small03.jpg">  <p>电烤锅</p>  <p>299元</p>  </div>  <div class="adv\_small">  <img src="../img/small03.jpg">  <p>电烤锅</p>  <p>299元</p>  </div>  <div class="adv\_small">  <img src="../img/small03.jpg">  <p>电烤锅</p>  <p>299元</p>  </div>  </div>  </div>  </div>    <!--广告-->  <div id="adv3">  <img src="../img/footer.jpg" width="100%">  </div>    <!--友情链接-->  <div id="foot">  <p>  <a href="#">关于我们</a>  <a href="#">联系我们</a>  <a href="#">招贤纳士</a>  <a href="#">法律声明</a>  <a href="#">友情链接</a>  <a href="#">支付方式</a>  <a href="#">配送方式</a>  <a href="#">服务声明</a>  <a href="#">广告声明</a>  </p>  <p>  Copyright © 2009-2017 极客商城 版权所有  </p>  </div>    </div>  </body> |

CSS样式代码

|  |
| --- |
| <style>  \*{  margin: 0px;  padding: 0px;  font-size: 13px;  }  a{  color:cornflowerblue;  text-decoration: none;  }  #container{  width: 1300px;    /\*居中显示\*/  margin: auto;  }  /\*#logo{  height: 50px;  border: 1px solid green;  }\*/  .top{  width: 430px;    height: 50px;  float: left;  /\*文本内容垂直居中\*/  line-height:50px ;  }    #menu{  height: 50px;    background-color: black;  }  #menu ul li{  /\*display: inline;\*/  display: inline-block;  width: 80px;  height: 48px;  text-align: center;  line-height: 48px;  color:gainsboro;  }  .clear{  clear: both;  }  .hot{    height: 500px;  }  .hot\_top{  height: 50px;    padding-left: 10px;  }  .hot\_top span{  font-size: 24px;  line-height: 50px;  }  .hot\_bottom\_left{  width: 220px;  height: 450px;  float: left;  }  .hot\_bottom\_right{  width: 1080px;  height: 450px;    float: left;  }  .adv\_big{  width:532px ;  height: 225px;    float: left;  }  .adv\_small{  width: 176px;  height: 225px;    float: left;  text-align: center;  }  p{  line-height: 30px;  }  #foot{  text-align: center;  }    </style> |

效果图:





至此，使用DIV+CSS完成首页重构案例已经完成。

希望同学们能通过今天的学习，掌握HTML中表单的基本使用和CSS样式的应用。