

# HTML 表单&CSS

## 今日内容介绍

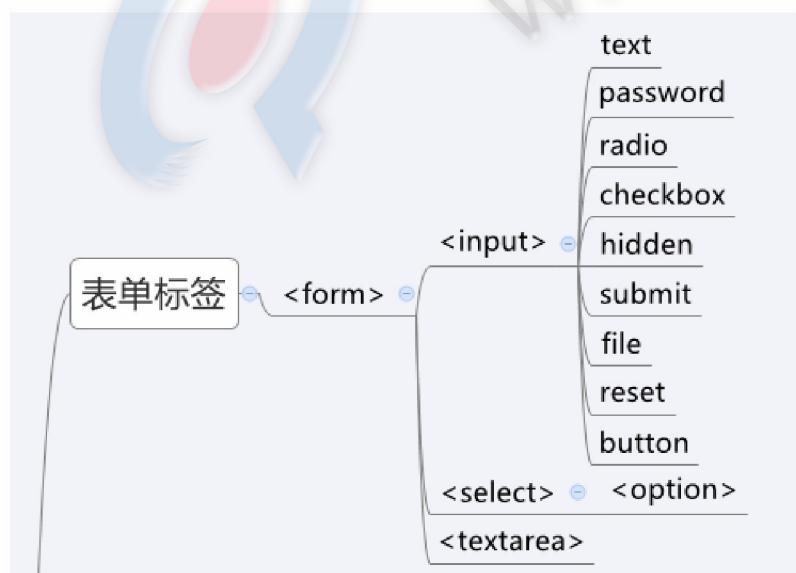
- ◆ 使用 html 的表单标签编写“注册页面”
- ◆ 使用 DIV 和 CSS 重写网站首页

## 今日内容学习目标

- ◆ 能够使用<form><input>等标签编写注册表单
- ◆ 熟悉的 CSS 样式的 3 种导入方式
- ◆ 熟悉 CSS 选择器的使用
- ◆ 了解 CSS 基本属性使用
- ◆ 了解 CSS 盒子模型

## 今日学习标签总览

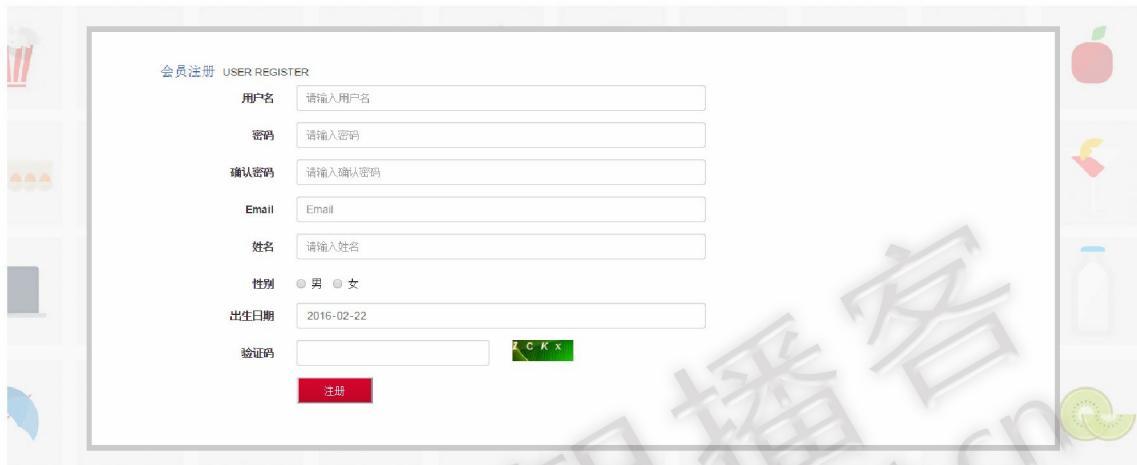
今天讲学习表单标签，提供总览方便大家更好的吸收。



# 第1章 网站用户注册页面显示

## 1.1 案例介绍

所有的 html 标签中，表单标签是最重要的。在实际开发中，最经典的实例就是用户注册，覆盖了表单标签的所有的元素。效果图如下：



## 1.2 相关知识点

本案例中使用的标签如下：

- <form>
- <input>
- <select>

### 1.2.1 案例相关标签

#### 1.2.1.1 表单标签：<form>

<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，负责搜集数据的标签必须存放在表单标签体内容。

- **action 属性：**请求路径，确定表单提交到服务器的地址（路径）
- **method 属性：**请求方式。常用的取值：GET、POST
  - ◆ **GET：**默认值
    - 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 k/v，追加是使用?连接，之后每一对数据使用&连接
    - 因为请求路径长度有限，所有 GET 请求提交的数据有限。
  - ◆ **POST：**
    - 提交的数据不再请求路径上追加（及不显示在地址栏上）

- 提交的数据大小不显示

```
<body>
    <!--表单-->
    <form action="" method="">
        <!--此处的内容可能被提交到服务器-->
    </form>
    <!--此处的内容在<form>标签外部，此处数据不能提交到表单-->
</body>
```

### 1.2.1.2 输入域标签 : <input>

- <input> 标签用于获得用户输入信息, type 属性值不同, 搜集方式不同。最常用的标签。

- type 属性

- ◆ text: 文本框, 单行的输入字段, 用户可在其中输入文本。默认宽度为 20 个字符
    - ◆ password: 密码框, 密码字段。该字段中的字符以黑圆显示。
    - ◆ radio: 单选框, 表示一组互斥选项按钮中的一个。当一个按钮被选中, 之前选中的按钮就变为非选中的。
    - ◆ 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, 以下标签使用也需要大家掌握。

- ◆ checkbox: 复选框
    - ◆ file: 文件上传组件, 提供“浏览”按下可以选择需要上传文件。
    - ◆ hidden: 隐藏字段。数据会发送给服务器, 但浏览器不进行显示。
    - ◆ reset: 重置按钮。将表单恢复到默认值。
    - ◆ image: 图形提交按钮, 通过 src 给按钮设置图片。
    - ◆ button: 普通按钮, 常用于与 JavaScript 结合使用。
  - name: 元素名, 如果需要表单数据提交到服务器, 必须提供 name 属性值, 服务器通过属性值获得提交的数据。
  - value 属性: 设置 input 标签的默认值。submit 和 reset 为按钮显示数据
  - size: 大小
  - checked 属性: 单选框或复选框被选中。
  - readonly: 是否只读
  - disabled: 是否可用
  - maxlength: 允许输入的最大长度

### 1.2.1.3 下拉列表标签: <select>

- <select> 下拉列表。可以进行单选或多选。需要使用子标签<option>指定列表项
  - name 属性: 发送给服务器的名称
  - multiple 属性: 不写默认单选, 取值为“multiple”表示多选。
  - size 属性: 多选时, 可见选项的数目。

- <option> 子标签：下拉列表中的一个选项（一个条目）。
  - ◆ selected : 勾选当前列表项
  - ◆ value : 发送给服务器的选项值。

#### 1.2.1.4 文本域标签： <textarea>

- <textarea> 文本域。多行的文本输入控件。
  - cols 属性：文本域的列数
  - rows 属性：文本域的行数

#### 1.2.1.5 按钮标签： <button> （了解）

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

### 1.3 案例分析

#### 1.3.1 知识点分析

为了结构更好的显示表单，本案例我们需要使用 table 表格布局。根据“验证码”我们需要提供 3 列来显示数据，其他需要将中和右合并。最终需要创建 3\*10 表格。



会员注册 USER REGISTER

用户名	请输入用户名	
密码	请输入密码	
确认密码	请输入确认密码	
Email	Email	
姓名	请输入姓名	
性别	<input type="radio"/> 男 <input checked="" type="radio"/> 女	
出生日期	年 -月-日	
验证码	<input type="text"/>	
<input type="button" value="注册"/>		

## 1.4 案例实现

#### 1.4.1 布局

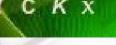
```
<td colspan="2"></td>
</tr>
</table>
```

## 1.4.2 表单

```
<table width="600">
<tr>
<td colspan="3">
<font color="#3164af">会员注册</font> USER REGISTER
</td>
</tr>
<tr>
<td align="right">用户名</td>
<td colspan="2"><input type="text" name="loginname" size="60"/> </td>
</tr>
<tr>
<td align="right">密码</td>
<td colspan="2"><input type="password" name="loginpwd" size="60"/> </td>
</tr>
<tr>
<td align="right">确认密码</td>
<td colspan="2"><input type="password" name="reloginpwd" size="60"/> </td>
</tr>
<tr>
<td align="right">Email</td>
<td colspan="2"><input type="text" name="email" size="60"/> </td>
</tr>
<tr>
<td align="right">姓名</td>
<td colspan="2"><input name="text" name="username" size="60"/> </td>
</tr>
<tr>
<td align="right">性别</td>
<td colspan="2">
<input type="radio" name="gender" value="男" checked="checked" />男
<input type="radio" name="gender" value="女" />女
</td>
</tr>
<tr>
<td align="right">出生日期</td>
<td colspan="2"><input type="text" name="birthday" size="60"/> </td>
</tr>
```

```
<tr>
    <td width="80" align="right">验证码</td>
    <td width="100"><input type="text" name="verifyCode" /> </td>
    <td> </td>
</tr>
<tr>
    <td></td>
    <td colspan="2">
        <input type="submit" value="注册" />
    </td>
</tr>
</table>
```

### 会员注册 USER REGISTER

用户名	<input type="text"/>
密码	<input type="password"/>
确认密码	<input type="password"/>
Email	<input type="text"/>
姓名	<input type="text"/>
性别	<input checked="" type="radio"/> 男 <input type="radio"/> 女
出生日期	<input type="text"/>
验证码	<input type="text"/> 

## 第2章 使用 DIV+CSS 重写网站首页案例

### 2.1 案例介绍

上次课中我们使用了表格<table>对网站首页进行布局，但是表格布局有自身的缺陷，并不是特别灵活，我们将使用 UI 工程师经常使用 DIV+CSS 方式进行布局，这种布局方式更为灵活。



## 2.2 相关知识点：

### 2.2.1 什么是 div

div 就是 html 一个普通标签，进行区域划分。特性：独自占一行。独自不能实现复杂效果。必须结合 CSS 样式进行渲染。

div 通常其是块级元素

<div> 可定义文档中的分区或节（division/section）。

<div> 标签可以把文档分割为独立的、不同的部分。它可以用作严格的组织工具，并且不使用任何格式与其关联。

如果用 id 或 class 来标记 <div>，那么该标签的作用会变得更加有效。

### 2.2.2 CSS 的概述

#### 2.2.2.1 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 页面中的文本内容（字体、大小、对其方式等）、图片的外形（高宽、边框样式、边距等）以及版面的布局等外观显示样式。

CSS 可以是 HTML 页面更好看，CSS 色系的搭配可以让用户更舒服，CSS+DIV 布局更佳灵活，更容易绘制出用户需要的结构。

#### 2.2.2.2 CSS 名词解释

CSS (Cascading Style Sheets) : 指层叠样式表

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，给同一个 HTML 标签添加样式，最后所有的样式都叠加到一起，共同作用于该标签。

#### 2.2.2.3 CSS 样式规则

使用 HTML 时，需要遵从一定的规范。CSS 亦如此，要想熟练的使用 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;属性 2:属性值;...}

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方式出现，使用英文冒号 “:” 分隔。多个属性之间使用英文分号 “;” 分隔。例如：

```
<style>
    h2 {
        color:red;
        font-size:100px;
    }
</style>
```

初学者在书写 CSS 样式时，除了要遵循 CSS 样式规则，还必须注意 CSS 代码结构中的几个特点，具体如下：

- 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。
- 多个属性之间必须用英文状态下的分号隔开，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。
- 如果属性的值由多个单词组成且中间包含空格，则必须为这个属性值加上英文状态下的引号。例如：

```
p { font-family:"Times New Roman"; }
```

- 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 */
```

- 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、回车键等对样式代码进行排版，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; }
```

和

```
h1{
    font-size:20px;          /* 定义字体大小属性 */
    color:red;               /* 定义颜色属性 */
}
```

- 上述两段代码所呈现的效果是一样的，但是，第二种书写方式的可读性更高。需要注意的是，属性的值和单位之间是不允许出现空格的，否则浏览器解析时会出错。例如，下面这行代码就是不正确的。

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

- 

## 2.2.2.4 引入 CSS 样式

CSS 使用非常灵活，及可以嵌入在 HTML 文档中，也可以是一个单独的文件，如果是单独的文件，则必须以.css 为扩展名。CSS 和 HTML 的结合 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。

```
<!-- 方式 1：行内样式
    color : 表示字体颜色
    font-size : 表示字体大小
-->
<a style="color: #f00; font-size: 30px;">传智播客</a>
```

← → C 127.0.0.1:8020/day02/01.demo00\_css.html

## 传智播客

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 展示结构、CSS 显示效果）相分离，所以一般很少使用。学习阶段有时候为了快速编程，偶有使用。

### 2) 内部样式

内部样式又称为内嵌式，是将 CSS 代码集中卸载 HTML 文档的<head>头部标签体中，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>标签来定义 CSS

```
<!--方式 2：内部样式
background-color : 表示背景色
-->
<style type="text/css">
body{
    background-color: #ddd;
}
</style>
```

← → C 127.0.0.1:8020/day02/01.demo00\_css.html

## 传智播客

内嵌式 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，使用内嵌式是个不错的选择。但如果是一个网站，不建议使用这种方式，因为他不能充分发挥 CSS 代码的重用优势。

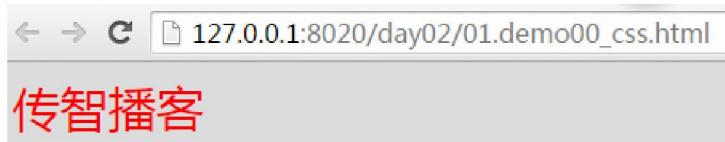
### 3) 外部样式

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，通过<link>标签将样式连接到 HTML 文档中。

```
<!--方式 3：外部样式
rel="stylesheet" ,固定值，表示样式表
type="text/css" , 固定值，表示 css 类型
href ， 表示 css 文件位置
font-family 表示使用的字体系列，多个字体使用逗号分隔。例如：“字体 1, 字体 2, 字体 3”，此时优先使用“字体 1”，如果“字体 1”系统不存在，再使用“字体 2”，以此类推。
-->
<link rel="stylesheet" type="text/css" href="css/demo00.css"/>
```

01.demo00\_css.html demo00.css

```
1 body{
2     font-family: "微软雅黑";
3 }
```



链入式最大的好处是同一个 CSS 样式表可以被不同的 HTML 页面链接使用，同时一个 HTML 页面也可以通过多个<link />标记链接多个 CSS 样式表。

## 优先级

```
element.style {  
    color: #f00;  
}  
  
font {  
    font-family: "éšjäé";  
}  
  
font {  
     font-size: 100px;  
}
```

### 2.2.3 选择器

要想将 CSS 样式应用于特定的 HTML 元素，首先需要找到该目标元素。在 CSS 中，执行这一任务的样式规则部分被称为选择器，本小节将对 CSS 基础选择器进行详细地讲解，具体如下：

#### 2.2.3.1 元素选择器

标记选择器是指用 HTML 标记名称作为选择器，按标记名称分类，为页面中某一类标记指定统一的 CSS 样式。其基本语法格式如下：

```
标记名{属性 1:属性值 1; 属性 2:属性值 2; 属性 3:属性值 3; }
```

该语法中，所有的 HTML 标记名都可以作为标记选择器，例如 body、h1、p、strong 等。用标记选择器定义的样式对页面中该类型的所有标记都有效。

例如：

```
h1{  
    color: #F00;  
    font-size: 50px;  
}  
  
<h1>传智播客</h1>
```

# 传智播客

标记选择器最大的优点是能快速为页面中同类型的标记统一样式，同时这也是他的缺点，不能设计差异化样式。

### 2.2.3.2 ID 选择器

id 选择器使用 “#” 进行标识，后面紧跟 id 名，其基本语法格式如下：

```
#id名{属性 1:属性值 1; 属性 2:属性值 2; 属性 3:属性值 3; }
```

该语法中，id 名即为 HTML 元素的 id 属性值，大多数 HTML 元素都可以定义 id 属性，元素的 id 值是唯一的，只能对应于文档中某一个具体的元素。

例如：

```
#demo1{  
    color:#0f0;  
}  
  
<h1 id="demo1">黑马训练营</h1>
```

## 黑马训练营

“标签选择器”和“id 选择器”共同作用的效果

### 2.2.3.3 类选择器

类选择器使用 “.”（英文点号）进行标识，后面紧跟类名，其基本语法格式如下：

```
.类名{属性 1:属性值 1; 属性 2:属性值 2; 属性 3:属性值 3; }
```

该语法中，类名即为 HTML 元素的 class 属性值，大多数 HTML 元素都可以定义 class 属性。类选择器最大的优势是可以为元素对象定义单独或相同的样式。

例如：

```
.myClass{  
    font-size: 25px;  
}  
  
<h1 class="myClass">Java 培训</h1>
```

## Java培训

“标签选择器”和“类选择器”共同作用的效果

类选择器的高级用法：给指定的标签设置 class 样式

```
标签.类名{属性 1:属性值 1; 属性 2:属性值 2; 属性 3:属性值 3; }
```

### 2.2.3.4 扩展：属性选择器

属性选择器，在标签后面使用中括号标记，其基本语法格式如下：

```
标签名[标签属性='标签属性值']{属性 1:属性值 1; 属性 2:属性值 2; 属性 3:属性值 3; }
```

该选择器，是对“元素选择器”的扩展，对一组标签进一步过滤。

例如：

```
<style>  
    input[type="text"]{  
        background-color: yellow;
```

```
}

input[type="password"]{
    background-color: green;
}

</style>
```

### 2.2.3.5 扩展：包含选择器

包含选择器，两个标签之间使用空格，给指定父标签的后代标签设置样式，可以方便在区域内编写样式。

父标签 后代标签{属性 1:属性值 1; 属性 2:属性值 2; 属性 3:属性值 3; }

该选择器，是对“元素选择器”的扩展，对一个标签内部所有后代标签进行过滤。

```
<style>

#d1 div{
    color: red;
}

</style>
```

### 2.2.4 CSS 的样式：

#### 2.2.4.1 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" , 1 像素实边红色。
    - 样式取值: solid 实线, none 无边, double 双线 等
- width、height: 用于设置标签的宽度、高度。

```
<style type="text/css">

div{
    border:1px solid #000; /*1 像素, 实边, 黑色*/
    width:200px;
    height:200px;
}

</style>
```



#### 2.2.4.2 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

选择器{float:属性值;}

常用属性值:

left: 元素向左浮动

right: 元素向右浮动

none: 元素不浮动（默认值）

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。

选择器{clear:属性值;}

常用属性值:

left: 不允许左侧有浮动元素（清除左侧浮动的影响）

right: 不允许右侧有浮动元素（清除右侧浮动的影响）

both: 同时清除左右两侧浮动的影响

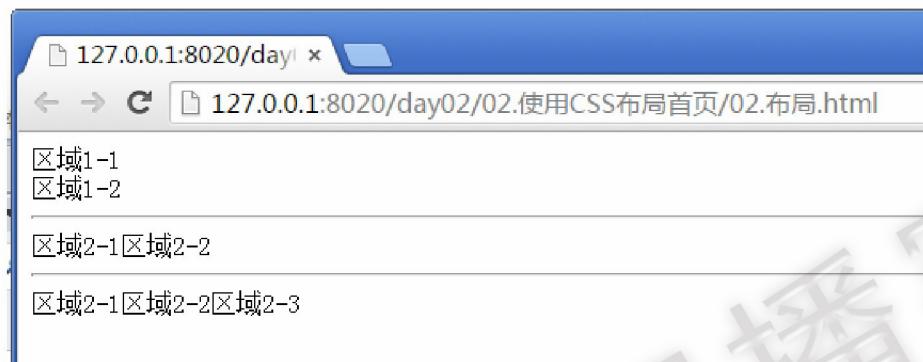
例如:

```
<!--默认上下布局-->
<div>
    <div>区域 1-1</div>
    <div>区域 1-2</div>
</div>

<hr />

<!--浮动左右布局-->
<div>
    <div style="float: left;">区域 2-1</div>
    <div style="float: left;">区域 2-2</div>
</div>
```

```
<!--取消浮动，另起一行布局-->
<div style="clear:both"></div>
<hr />
<div>
    <div style="float: left;">区域 2-1</div>
    <div style="float: left;">区域 2-2</div>
    <div style="float: left;">区域 2-3</div>
</div>
```



### 2.2.4.3 转换：display

HTML 提供丰富的标签，这些标签被定义成了不同的类型，一般分为：块标签和行内标签。

- 块标签：以区域块方式出现。每个块标签独自占据一整行或多整行。
  - 常见的块元素：`<h1>`、`<div>`、`<ul>`等
- 行内元素：不必在新的一行开始，同时也不强迫其他元素在新的一行显示。
  - 常见的行内元素：`<span>`、`<a>` 等

在开发中，希望行内元素具有块元素的特性，需要使用 `display` 进行转换

选择器{display:属性值}

常用的属性值：

`inline`: 此元素将显示为行内元素（行内元素默认的 `display` 属性值）  
`block`: 此元素将显为块元素（块元素默认的 `display` 属性值）  
`inline-block` : 将对象呈递为内联对象，但是对象的内容作为块对象呈递。  
`none`: 此元素将被隐藏，不显示，也不占用页面空间。

例如：

```
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <style type="text/css">
            span{
                border :1px solid #000;
                width:100px;
```

```
height:40px;
}
</style>
</head>
<body>
<!--默认显示一行，边框环绕，高宽没有作用--&gt;
&lt;span&gt;显示 1-1&lt;/span&gt;
&lt;span&gt;显示 1-2&lt;/span&gt;

<!--每一行显示，高宽有作用--&gt;
&lt;span style="display: block;"&gt;显示 2-1&lt;/span&gt;
&lt;span style="display: block;"&gt;显示 2-2&lt;/span&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```



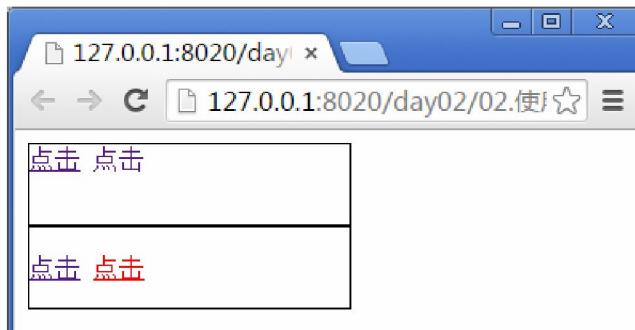
#### 2.2.4.4 字体：color、font-size

color: 颜色，字体颜色

例如：

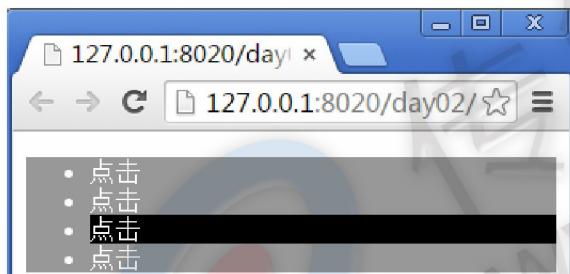
```
<div style="height: 50px; width: 200px; border: 1px solid #000;">
    <a href="">点击</a>
    <a href="" style="text-decoration: none;">点击</a>
</div>

<div style="height: 50px; width: 200px; border: 1px solid #000;">
    <a href="" style="line-height: 50px;">点击</a>
    <a href="" style="color: red;">点击</a>
</div>
```



#### 2.2.4.5 背景色：background-color

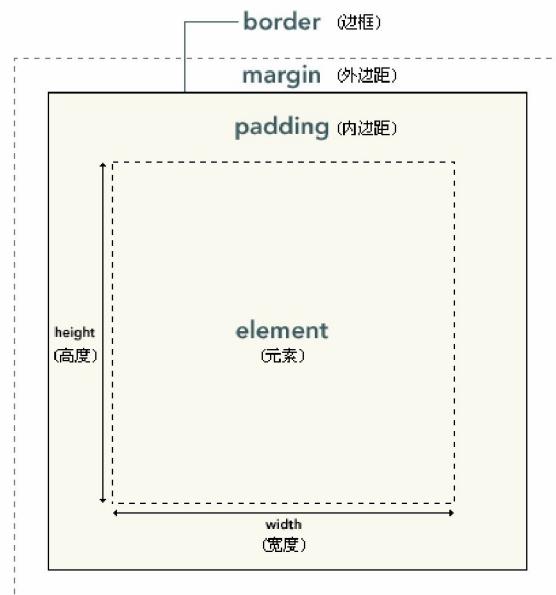
```
<ul style="background-color: #999;color: #FFF;">
    <li>点击</li>
    <li>点击</li>
    <li style="background-color: #000;">点击</li>
    <li>点击</li>
</ul>
```



#### 2.2.5 CSS 的盒子模型

##### 2.2.5.1 什么是盒子模型

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



### 2.2.5.2 内边距: padding.

例如, 如果您希望所有 `h1` 元素的各边都有 10 像素的内边距, 只需要这样:

```
h1 {padding: 10px;}
```

您还可以按照上、右、下、左的顺序分别设置各边的内边距, 各边均可以使用不同的单位或百分比值:

```
h1 {padding: 10px 0.25em 2ex 20%;}
```

#### 单边内边距属性

也通过使用下面四个单独的属性, 分别设置上、右、下、左内边距:

- [padding-top](#)
- [padding-right](#)
- [padding-bottom](#)
- [padding-left](#)

### 2.2.5.3 边框: border

- [border-top-style](#)
- [border-right-style](#)
- [border-bottom-style](#)
- [border-left-style](#)

## 2.2.5.4 外边距：margin

- margin-top
- margin-right
- margin-bottom
- margin-left

## 2.3 案例分析

整个案例需要最外层提供一个 `div`，用于作为父容器，方便这个页面的布局。整个首页信息量比较大，整个实现过程，将采用分步骤，一个区域一个区域独立完成。每一个区域布局完成之后，一并填充数据。最终将整个页面编写完成。

## 2.4 案例实现

### 2.4.1 框架

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <style type="text/css">

            </style>
    </head>
    <body>
        <!--父容器，暂时没有，方便之后开发整体布局-->
        <div>

            </div>
    </body>
</html>
```

### 2.4.2 header 信息

- 样式

```
.header{  
    height:60px;  
}  
.header .top{  
    float: left;  
    width:33.2%;  
    height:100%;  
}  
  
ul{  
    list-style: none;  
    margin: 0;  
}  
ul li{  
    display: inline-block;  
}  
  
.top a{  
    text-decoration: none;  
    line-height: 60px;  
}
```

#### ● html 代码

```
<!--#1 header 信息-->  
<div class="header">  
    <div class="top">  
          
    </div>  
    <div class="top">  
          
    </div>  
    <div class="top">  
        <ul>  
            <li><a href="">登录</a></li>  
            <li><a href="">注册</a></li>  
            <li><a href="">购物车</a></li>  
        </ul>  
    </div>  
</div>
```

### 2.4.3 菜单

#### ● css 样式

```
.menu{
```

```
background-color: #333;
}

.menu a{
    text-decoration: none;
    display: block;
    padding: 10px 15px;
    color: #fff;
    line-height: 30px;
}

.menu a.show{
    background-color: #000;
}
```

#### ● html 代码

```
<!--#2 菜单-->
<div class="menu">
    <ul>
        <li><a href="">首页</a></li>
        <li><a href="" class="show">电脑办公</a></li>
        <li><a href="">手机数码</a></li>
        <li><a href="">鞋靴箱包</a></li>
    </ul>
</div>
```

#### 2.4.4 轮播图

```
<!--#3 轮播图-->
<div style="width: 100%;">
    
</div>
```

#### 2.4.5 最新商品信息

##### ● css 样式

```
.product{
    width:100%
}

.product .left{
    float: left;
    width:14%;
    height: 500px;
```

```
}

.product .right{
    float: left;
    width:85%;
}

.product .first{
    float:left;
    width:50%;
    height:250px;
}

.product .item{
    float:left;
    width:16.5%;
    height:250px;
}

.product .first img,.left img{
    height: 100%;
    width: 100%;
}

.product .right img{
    height: 90%;
    width: 90%;
}
```

## ● html 代码

```
<!--#4 商品列表-->
<div class="product">
    <div>
        <h2>最新商品 </h2>
    </div>
    <div class="left">
        
    </div>
    <div class="right">
        <div class="first"></div>
        <div class="item"> </div>
        <div class="item"> </div>
    </div>
```

```
</div>
```

## 2.4.6 广告

- css 样式

```
.clear{  
    clear: both;  
}
```

- html 代码

```
<!--#5 广告-->  
<div class="clear"></div>  
<div style="height: 100px; width: 100%;">  
      
</div>
```

## 2.4.7 footer 信息

```
<!--#6 商品列表, 复制#4-->  
<!--#7 其他-->  
<div class="clear"></div>  
<div>  
      
</div>  
<div align="center">  
    关于我们  
    联系我们  
    招贤纳士  
    法律声明  
    <a href="">友情链接</a>  
    支付方式  
    配送方式  
    服务声明  
    广告声明 <br/>  
    Copyright © 2005-2016 传智商城 版权所有  
</div>
```

## 2.5 案例总结

### 2.5.1 div 和 span 比较

- div 块级元素，默认独自占一行
- span 行内元素，默认所有的在一行。都可以进行区域划分。

通过 css 对其进行转换

display: block、none