

Tarena Teaching System

# HTML和CSS

学员用书



Java企业应用及互联网  
高级工程师培训课程

# 目 录

<b>Unit01</b> .....	1
1. HTML 基础语法 .....	4
1.1. HTML 概述 .....	4
1.1.1. Web 的工作原理 .....	4
1.1.2. Web 的相关技术 .....	5
1.1.3. 什么是 HTML .....	5
1.1.4. Web 浏览器 .....	6
1.2. HTML 基础语法 .....	6
1.2.1. 标记 .....	6
1.2.2. 元素和属性 .....	7
1.2.3. 通用属性 .....	8
1.2.4. 注释 .....	8
1.3. 文档结构 .....	8
1.3.1. HTML 的文档结构 .....	8
1.3.2. 文档类型声明 .....	9
1.3.3. <head> 元素 .....	9
1.3.4. 文档头部内容 --<title> .....	9
1.3.5. 文档头部内容 --<meta> .....	10
2. 文本标记 .....	10
2.1. 文本标记概述 .....	10
2.1.1. 文本标记的作用 .....	10
2.1.2. 文本与特殊字符 .....	10
2.2. 使用文本标记 .....	11
2.2.1. 标题元素 <hn> .....	11
2.2.2. 段落元素 <p> .....	11
2.2.3. 换行元素   .....	11
2.2.4. 分区元素 <span>和<div> .....	12
2.2.5. 行内元素与块级元素 .....	12
3. 图像和链接 .....	12
3.1. URL .....	12
3.1.1. URL .....	12
3.1.2. 相对路径和绝对路径 .....	13
3.2. 图像 .....	13
3.2.1. 图像元素 <img> .....	13
3.3. 链接 .....	13

---

3.3.1. 链接元素 <a> .....	13
3.3.2. 锚点 .....	14
4. 列表 .....	14
4.1. 列表的作用.....	14
4.1.1. 列表的作用 .....	14
4.2. 使用列表 .....	14
4.2.1. 有序列表 <ol> .....	14
4.2.2. 无序列表 <ul> .....	15
4.2.3. 列表嵌套 .....	15
5. 表格 .....	15
5.1. 表格的作用.....	15
5.1.1. 表格的作用 .....	15
5.2. 使用表格 .....	16
5.2.1. 创建表格 .....	16
5.2.2. 表格的常用属性 .....	16
5.2.3. 表格标题 <caption> .....	16
5.3. 表格的复杂应用 .....	17
5.3.1. 行分组.....	17
5.3.2. 不规则表格 .....	17
5.3.3. 表格的嵌套 .....	18
6. 表单 .....	18
6.1. 表单概述 .....	18
6.1.1. 表单的作用 .....	18
6.1.2. 表单元素 <form> .....	19
6.1.3. 表单控件 .....	19
6.2. <input> 元素 .....	19
6.2.1. <input> 元素 .....	19
6.2.2. 文本框与密码框 .....	20
6.2.3. 单选框和复选框 .....	20
6.2.4. 按钮 .....	20
6.2.5. 隐藏域和文件选择框 .....	21
经典案例 .....	22
1. 创建 HTML 文档并设置头元素 .....	22
2. 使用文本标记为 HTML 页面添加内容 .....	23
3. 为 HTML 页面添加图像和链接 .....	27
4. 为 HTML 页面添加导航目录 .....	31
5. 为 HTML 页面添加数据表格 .....	36
6. 创建基础表单 .....	44

---

课后作业 .....	52
<b>Unit02 .....</b>	<b>56</b>
1. 表单 .....	58
1.1. 其他控件 .....	58
1.1.1. <label> 元素 .....	58
1.1.2. 选项框 .....	58
1.1.3. <textarea> 元素 .....	59
1.1.4. 为控件分组 .....	59
2. CSS 概述 .....	59
2.1. CSS 概述 .....	59
2.1.1. CSS 的作用 .....	59
2.1.2. 什么是 CSS .....	60
2.2. 使用 CSS 样式表 .....	61
2.2.1. 使用 CSS 样式表 .....	61
2.2.2. 内联方式使用 CSS .....	61
2.2.3. 内部样式表 .....	61
2.2.4. 外部样式表 .....	62
3. CSS 语法 .....	63
3.1. CSS 语法规规范 .....	63
3.1.1. CSS 语法规规范总结 .....	63
3.1.2. CSS 样式表特征 .....	63
3.1.3. 样式优先级 .....	63
3.2. CSS 选择器 .....	64
3.2.1. 元素选择器 .....	64
3.2.2. 类选择器 .....	64
3.2.3. id 选择器 .....	65
3.2.4. 选择器分组 .....	66
3.2.5. 派生选择器 .....	66
3.2.6. 伪类选择器 .....	66
4. 尺寸与框模型 .....	67
4.1. 尺寸与边框 .....	67
4.1.1. 样式单位 .....	67
4.1.2. 尺寸 .....	67
4.1.3. 边框 .....	68
4.2. 框模型 .....	69
4.2.1. 框模型 .....	69
4.2.2. 内边距 padding .....	70
4.2.3. 外边距 margin .....	70

---

5. 背景 .....	71
5.1. 背景色 .....	71
5.1.1. 背景色 background-color .....	71
5.2. 背景图像 .....	71
5.2.1. 背景图片 background-image .....	71
5.2.2. 背景重复 background-repeat .....	72
5.2.3. 背景定位 background-position .....	72
5.2.4. 背景图片的固定 background-attachment .....	73
经典案例 .....	74
1. 创建复杂表单 .....	74
2. 使用 CSS 样式 .....	81
3. 测试 CSS 样式优先级 .....	85
4. CSS 选择器 (1) .....	89
5. CSS 选择器 (2) .....	92
6. NetCTOSS--admin_list 页面布局 .....	97
7. NetCTOSS--admin_list 页面背景 .....	102
课后作业 .....	106
<b>Unit03</b> .....	108
1. 文本格式化 .....	110
1.1. 文本格式化 .....	110
1.1.1. 控制字体 .....	110
1.1.2. 控制文本格式 .....	110
2. 表格样式 .....	111
2.1. 表格常用样式属性 .....	111
2.1.1. 表格常用样式属性 .....	111
2.1.2. 垂直方向对齐 .....	111
2.2. 表格特有样式属性 .....	112
2.2.1. 边框合并 .....	112
2.2.2. 边框边距 .....	112
3. 浮动定位 .....	113
3.1. 定位概述 .....	113
3.1.1. 定位概述 .....	113
3.1.2. 普通流定位 .....	113
3.2. 浮动定位 .....	114
3.2.1. 浮动概述 .....	114
3.2.2. 浮动定位 .....	114
3.2.3. float 属性 .....	115
3.2.4. clear 属性 .....	116

---

4. 显示 .....	116
4.1. 显示方式 .....	116
4.1.1. 显示方式 .....	116
4.1.2. display 属性 .....	116
4.2. 光标 .....	117
4.2.1. 光标 .....	117
4.2.2. cursor 属性 .....	117
5. 列表样式 .....	118
5.1. 列表样式 .....	118
5.1.1. 列表项标志 list-style-type .....	118
5.1.2. 列表项图像 list-style-image .....	118
6. 定位 .....	119
6.1. 定位机制 .....	119
6.1.1. 什么是定位 .....	119
6.1.2. 定位属性 .....	119
6.1.3. 相对定位 .....	120
6.1.4. 绝对定位 .....	121
6.1.5. 堆叠顺序 .....	122
6.1.6. 固定定位 .....	123
经典案例 .....	124
1. NetCTOSS--admin_list 页面文本 .....	124
2. NetCTOSS--admin_list 页面表格 .....	129
3. NetCTOSS--admin_list logo 图片 .....	136
4. NetCTOSS--admin_list 导航栏 .....	141
5. NetCTOSS--admin_list 操作提示框 .....	149
课后作业 .....	159



# HTML 和 CSS

## Unit01

知识体系.....Page 4

HTML 基础语法	HTML 概述	Web 的工作原理
		Web 的相关技术
		什么是 HTML
		Web 浏览器
HTML 基础语法	HTML 基础语法	标记
		元素和属性
		通用属性
		注释
文档结构	文档结构	HTML 的文档结构
		文档类型声明
		<head> 元素
		文档头部内容 --<title>
		文档头部内容 --<meta>
文本标记	文本标记概述	文本标记的作用
		文本与特殊字符
使用文本标记	使用文本标记	标题元素 <hn>
		段落元素 <p>
		换行元素  
		分区元素 <span>和<div>
		行内元素与块级元素
图像和链接	URL	URL
		相对路径和绝对路径
	图像	图像元素 <img>
	链接	链接元素 <a>
列表	使用列表	锚点
		列表的作用
		有序列表 <ol>
		无序列表 <ul>
		列表嵌套

表格	表格的作用	表格的作用
	使用表格	创建表格
		表格的常用属性
		表格标题 <caption>
表格的复杂应用	行分组 不规则表格 表格的嵌套	行分组
		不规则表格
		表格的嵌套
表单	表单概述	表单的作用
		表单元素 <form>
		表单控件
	<input> 元素	<input> 元素
		文本框与密码框
		单选框和复选框
		按钮
		隐藏域和文件选择框

## 经典案例.....Page 22

创建 HTML 文档并设置头元素	HTML 文档的结构
	文档类型声明
	<head> 元素
	文档头部内容 --<title>
	文档头部内容 --<meta>
使用文本标记为 HTML 页面添加内容	文本与特殊字符
	标题元素<hn>
	段落元素<p>
	换行元素 
	分区元素<span>和<div>
	行内元素与块级元素
为 HTML 页面添加图像和链接	图像元素<img>
	链接元素<a>
	锚点
为 HTML 页面添加导航目录	无序列表<ul>
	列表嵌套
为 HTML 页面添加数据表格	创建表格
	表格的常用属性
	表格标题<caption>
	行分组
	不规则表格
	表格的嵌套
创建基础表单	表单元素<form>
	表单控件

	<input> 元素
	文本框与密码框
	单选框和复选框
	按钮
	隐藏域和文件选择框

课后作业.....Page 52

达内IT培训集团

## 1. HTML 基础语法

### 1.1. HTML 概述

#### 1.1.1. 【HTML 概述】Web 的工作原理



知识讲解

**Web 的工作原理 (续3)**

Client

Server

value

### 1.1.2. 【HTML 概述】Web 的相关技术

知识讲解

**Web 的相关技术**

- 客户端技术：运行于客户端，由浏览器来解释运行
  - HTML
  - CSS
  - JavaScript
  - 浏览器兼容问题
- 服务器端技术：运行于服务器端，可以和数据库进行交互
  - ASP.NET ( Active Server Page )
  - PHP ( Hypertext Preprocessor )
  - JSP ( Java Server Page )

+

### 1.1.3. 【HTML 概述】什么是 HTML

知识讲解

**什么是 HTML**

- HTML ( HyperText Markup Language ) : 超文本标记语言，一种纯文本类型的语言
- 用来设计网页的标记语言
- 用该语言编写的文件，以 .html 或者 .htm 为后缀
- 由浏览器解释执行
- HTML 页面上，可以嵌套用脚本语言编写的程序段，如：VBScript, JavaScript

+

#### 1.1.4. 【HTML 概述】Web 浏览器

**Web 浏览器**

**知识讲解**

- 主要功能
  - 代理访问者提交请求
  - 作为 HTML 解释器和内嵌脚本程序执行器
  - 用图形化的方式显示 HTML 文档
- 主要Web浏览器产品
  - IE
  - Firefox
  - Chrome
  - Opera
  - Safari

Tarena  
达内科技

### 1.2. HTML 基础语法

#### 1.2.1. 【HTML 基础语法】标记

**标记**

**知识讲解**

- HTML 用于描述功能的符号称为“标记”，比如 `<p>`、`<h1>` 等
  - 标记在使用时必须使用尖括号括起来
  - 有封闭类型标记，也有非封闭类型的标记

Some text here.

Some text here.

标记会导致不同的显示效果

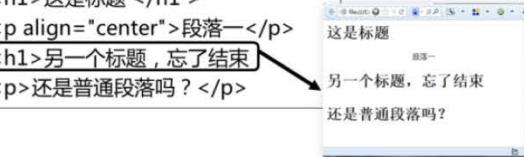
Tarena  
达内科技

**标记（续1）**

**知识讲解**

- 封闭类型标记（也叫双标记）：必须成对出现
  - `<标记>` 内容 `</标记>`
- 封闭类型的标记必须成对出现
  - 如果一个应该封闭的标记没有被封闭，则会产生意想不到的错误

```
<h1>这是标题 </h1 >
<p align="center">段落一</p>
<h1>另一个标题，忘了结束
<p>还是普通段落吗？</p>
```



Tarena  
达内科技

知识讲解

### 标记（续2）

- 非封闭类型标记，也叫做空标记，或者单标记
  - <标记>或者<标记 />
- 不能包含内容

普通文本一<br>普通文本二  
普通文本三<br />普通文本四

普通文本一  
普通文本二 普通文本三  
普通文本四

+

#### 1.2.2. 【HTML 基础语法】元素和属性

知识讲解

### 元素和属性

- 元素
  - 即标记
  - 每一对尖括号包围的部分，如 <body></body> 包围的部分就叫做 body 元素
- 属性
  - 用来修饰元素
  - 每个属性都有值
  - 属性的声明必须位于开始标记里
  - 一个标记的属性可能不止一个，多个属性之间用空格隔开
  - 多个属性之间不区分先后顺序

+

知识讲解

### 元素和属性（续1）

开始标记

<body>  
  <p align= "center" >段落一</p>  
</body>

结束标记 属性名称 属性值 body 元素 p 元素

+

### 1.2.3. 【HTML 基础语法】通用属性

知识讲解

+ +

**通用属性**

**Tarena**  
达内科技

- 每个元素都有自己所特有的属性
- 有些属性是绝大多数元素都支持的属性，称为通用属性（或标准属性）
  - id
  - title
  - class
  - style

### 1.2.4. 【HTML 基础语法】注释

知识讲解

+ +

**注释**

**Tarena**  
达内科技

- 为代码添加适当的注释是一种良好的编码习惯
- 注释只在编辑文档情况下可见，在浏览器展示页面时并不会显示
- 添加注释的语法如下：
  - <!-- 注释的文本内容 -->

```
<!--注释部分-->
<h1>这是标题</h1 >
```



- 1、 “” 之间的任何内容都不会显示在浏览器中
- 2、 注释不可以嵌套在其他注释中

## 1.3. 文档结构

### 1.3.1. 【文档结构】HTML 的文档结构

知识讲解

+ +

**HTML 的文档结构**

**Tarena**  
达内科技

- 文档类型声明
- html 页面
  - 文件头 : <head></head>
  - 文档主体部分 : <body></body>

文档类型声明

```
<!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>                文件头
    </head>
    <body>                文档主体
    </body>
</html>
```

文件头
html 页面

### 1.3.2. 【文档结构】文档类型声明

知识讲解

**文档类型声明**

- 在文档的起始用DOCTYPE声明指定版本和风格
  - 让浏览器清楚文档的版本、类型和风格
- Strict DTD  
`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`
- Transitional DTD  
`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`
- Frameset DTD  
`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">`

+

### 1.3.3. 【文档结构】<head> 元素

知识讲解

**<head> 元素**

- <head> 元素用于为页面定义全局信息
  - 所有其他头元素的容器
  - 紧跟在起始标签 <html> 之后
- 可包含
  - title、meta、script、style、link等

```
<head>
  <title>HTML 文档</title>
  <meta http-equiv="content-type"
        content="text/html;charset=utf-8" />
  <meta name="keywords" content="html,css" />
</head>
```

+

### 1.3.4. 【文档结构】文档头部内容——<title>

知识讲解

**文档头部内容 - - <title>**

- 标题元素 <title></title> 用于为文档定义标题
  - 标题元素的内容出现在浏览器顶部
  - 没有属性
  - 必须出现在 <head> 元素中
  - 一个文档只能有一个标题元素

```
<!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>  
    <title>我的第一个HTML页面</title>  
  </head>  
  <body>  
  </body>  
</html>
```

+



### 1.3.5. 【文档结构】文档头部内容 --<meta>

知识讲解

**文档头部内容 --<meta>**

- 元数据元素 <meta> 用于定义网页的基本信息
- 为空标记
- 常用属性有 : content、http-equiv

```
<head>
  <title>我的第一个HTML页面</title>
  <meta http-equiv="content-type"
        content="text/html;charset=utf-8" />
</head>
```

Tarena  
达内科技

+

## 2. 文本标记

### 2.1. 文本标记概述

#### 2.1.1. 【文本标记概述】文本标记的作用

知识讲解

**文本标记的作用**

- 文本是网页上的重要组成部分
- 直接书写的文本会用浏览器默认的样式显示
- 包含在标记中的文本则会被显示为标记所拥有的样式
  - 特殊字符
  - 注释
  - 标题元素
  - 段落元素
  - 换行元素
  - 分区元素

Tarena  
达内科技

+

#### 2.1.2. 【文本标记概述】文本与特殊字符

知识讲解

**文本与特殊字符**

- 空格折叠
  - 多个空格或制表符压缩成单个空格，即只显示一个空格
- 特殊字符（如空格），需要进行转义

```
The &lt;p&gt; element.&nbsp;&nbsp;&nbsp;&copy; 2011 by tarena.
```

↓

```
The <p> element. © 2011 by tarena.
```

Tarena  
达内科技

+

## 2.2. 使用文本标记

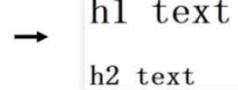
### 2.2.1. 【使用文本标记】标题元素 <hn>

知识讲解

**标题元素 <hn>**

- 标题元素让文字以醒目的方式显示，往往用于文章的标题
- 基本语法：`<h#> ... </h#>`
  - # = 1, 2, 3, 4, 5, 6
  - 从`<h1>`到`<h6>`，即 标题1-标题6

```
<h1>h1 text</h1>
<h2>h2 text</h2>
```



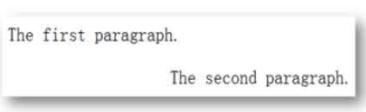
### 2.2.2. 【使用文本标记】段落元素 <p>

知识讲解

**段落元素 <p>**

- `<p>` 元素提供了结构化文本的一种方式
- `<p>` 元素中的文本会用单独的段落显示
  - 与前后的文本都换行分开
  - 添加一段额外的垂直空白距离，作为段落间距
  - 常用属性： `align`

```
<p>The first paragraph.</p>
<p align="right">The second paragraph.</p>
```



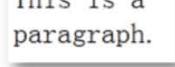
### 2.2.3. 【使用文本标记】换行元素 <br>

知识讲解

**换行元素 <br>**

- 使用 `<br>` 元素在任何地方创建手工换行
  - 空标记

```
<p>This is a <br/> paragraph.</p>
```



#### 2.2.4. 【使用文本标记】分区元素 <span>和<div>

知识讲解

### 分区元素 <span>和<div>

- 分区元素用于为元素分组，常用于页面布局
- 块分区元素：`<div></div>`
- 行内分区元素：`<span></span>`
  - 设置同一行文字内的不同格式

```
<div style="color:blue;">
  <p>first</p>
  <p>second</p>
</div>
<p>
  一周畅销榜
</p>
```

first  
second  
一周畅销榜

+

#### 2.2.5. 【使用文本标记】行内元素与块级元素

知识讲解

### 行内元素与块级元素

- 块级元素
  - 默认情况下，块级元素会独占一行，即元素前后都会自动换行
  - 如：`<p>`、`<div>`、`<hn>`
- 行内元素
  - 不会换行，可以和其他行内元素位于同一行
  - 如：`<span>`、`<a>`

+

### 3. 图像和链接

#### 3.1. URL

##### 3.1.1. 【URL】URL

知识讲解

### URL

- URL(Uniform Resource Locator)：统一资源定位器，用来标识网络中的任何资源
  - 文本、图片、音视频文件，段落，或其他超文本
- 完整 URL 的组成
  - 协议、机名、路径名、文件名

<http://www.w3.org/TR/CSS2/syndata.html>

http : 协议名（用://分隔）  
www.w3.org : 主机名  
/TR/CSS2/ : 路径  
syndata.html : 要链接的文件名

+

### 3.1.2. 【URL】相对路径和绝对路径

---

---

---

---

---

---

---

**相对路径和绝对路径**

**知识讲解**

- 相对路径
  - 指文件的位置是相对于当前文件的位置。它包括目录名，或指向一个可以从当前目录出发找到该文件的路径

```
./default/a.html
Images/a.gif
```

- 绝对路径
  - 指的是文件从最高级目录下开始的完整的路径，无论当前路径是什么，使用绝对路径总是能找到要链接的文件

```
http://www.w3.org/TR/CSS2/syndata.html
```

+

## 3.2. 图像

### 3.2.1. 【图像】图像元素 <img>

---

---

---

---

---

---

---

**图像元素 <img>**

**知识讲解**

- 使用 <img> 元素将图像添加到页面
  - 空标记
- 必须属性：src
- 常用属性：width、height

```

```

+

## 3.3. 链接

### 3.3.1. 【链接】链接元素 <a>

---

---

---

---

---

---

---

**链接元素 <a>**

**知识讲解**

- 使用 <a> 元素创建一个超级链接，语法如下：
- <a href="" target="">文本</a>
  - href 属性：链接 URL
  - target 属性：目标，可取值为 \_blank、\_self 等

```
<a href="http://www.google.com.hk">To Google</a>
<br />
<a href="teacher/teacher.asp" target="_blank">
    To other page
</a>
```

+

[To Google](http://www.google.com.hk)
  
[To other page](teacher/teacher.asp)

### 3.3.2. 【链接】锚点

**锚点**

- 锚点是文档中某行的一个记号
  - 用于链接到文档中的某个位置
- 定义锚点
  - <a name=" anchorname1" >锚点一</a>
- 链接到锚点：在锚点名前加上#
  - <a href="#" #anchorname1" >回到锚点一</a>

```
<a name="top"></a>
<br /><br /><br /><br /><br />
<a href="#top">Return to top</a>
```

定义锚点
→
使用链接，导航到锚点

## 4. 列表

### 4.1. 列表的作用

#### 4.1.1. 【列表的作用】列表的作用

**列表的作用**

- 列表是指将具有相似特征或者具有先后顺序的几行文字进行对齐排列
 

<ul style="list-style-type: none"> <li>• one</li> <li>• two</li> <li>• three</li> </ul>	<ul style="list-style-type: none"> <li>1. one</li> <li>2. two</li> <li>3. three</li> </ul>
---	--
- 所有的列表都由列表类型和列表项组成
  - 列表类型：有序列表`<ol>`和无序列表`<ul>`
  - 列表项：`<li>`，用于指示具体的列表内容

### 4.2. 使用列表

#### 4.2.1. 【使用列表】有序列表 `<ol>`

**有序列表 `<ol>`**

- `<ol>` 元素编写有序列表，用于列出页面上有特定次序的一些项目
- `<ol>` 元素中也只能包含列表项元素 `<li>`

```
<ol>
<li>one</li>
<li>two</li>
<li>three</li>
</ol>
```

1. one
→
2. two
→
3. three

14

#### 4.2.2. 【使用列表】无序列表 <ul>

**无序列表 <ul>**

• <ul> 元素表示无序列表，用于列出页面上没有特定次序的一些项目

• <ul> 元素中只能包含具体的列表项元素 <li>

- 列表中包含的每一项都必须包含在起始标记 <li> 和结束标记 </li> 之间

```
<ul>
<li>one</li>
<li>two</li>
<li>three</li>
</ul>
```

+<
• one
• two
• three

#### 4.2.3. 【使用列表】列表嵌套

**列表嵌套**

• 将列表元素嵌套使用，可以创建多层列表

- 常用于创建文档大纲、导航菜单等

```
<ul>
<li>
    Web基础知识
    <ul>
        <li>Web的工作原理</li>
    </ul>
</li>
<li>
    HTML快速入门
    <ul>
        <li>基础语法</li>
    </ul>
</li>
</ul>
```

+<
• Web基础知识
◦ Web的工作原理
• HTML快速入门
◦ 基础语法

## 5. 表格

### 5.1. 表格的作用

#### 5.1.1. 【表格的作用】表格的作用

**表格的作用**

- 表格通常用来组织结构化的信息
- 表格是一些被称作单元格的矩形框按照从左到右，从上到下的顺序排列在一起而形成的
- 表格的数据保存在单元格里
  - 显示表格数据
  - 设置页面布局

	行	
	第 1 行 第 1 列	第 1 行 第 2 列
	第 2 行 第 1 列	第 2 行 第 2 列
	第 3 行 第 1 列	第 3 行 第 2 列
		第 3 行 第 3 列

+<
列
行

15

## 5.2. 使用表格

### 5.2.1. 【使用表格】创建表格

知识讲解

### 创建表格

- 定义表格：使用成对的 `<table></table>` 标记
- 创建表行：使用成对的 `<tr></tr>` 标记
- 创建单元格：使用成对的 `<td></td>` 标记

```
<table border="1">
<tr>
  <td>第1行，第1列</td>
  <td>第1行，第2列</td>
</tr>
<tr>
  <td>第2行，第1列</td>
  <td>第2行，第2列</td>
</tr>
</table>
```

第1行, 第1列 第1行, 第2列  
第2行, 第1列 第2行, 第2列

+

### 5.2.2. 【使用表格】表格的常用属性

知识讲解

### 表格的常用属性

- `<table>`元素
  - border
  - width/height
  - align
  - cellpadding：单元格边框与内容之间的间距
  - cellspacing：单元格之间的间距
- `<tr>`元素
  - align valign
- `<td>`元素
  - align valign width/height colspan/rowspan

+

### 5.2.3. 【使用表格】表格标题 `<caption>`

知识讲解

### 表格标题 `<caption>`

- 使用 `<caption>` 元素为表格定义标题
  - 默认情况下，标题将在表格上方居中显示
- `<caption>` 标签必须紧随 `<table>` 标签之后，且只能对每个表格定义一个标题

```
<table border="1">
<caption>我的表格</caption>
<tr>
  <td>第1行，第1列</td>
  <td>第1行，第2列</td>
</tr>
</table>
```

我的表格

第1行, 第1列 第1行, 第2列

+

### 5.3. 表格的复杂应用

#### 5.3.1. 【表格的复杂应用】行分组

知识讲解

**行分组**

- 表格可以划分为3个部分：表头、表主体和表尾
- 表头行分组：`<thead></thead>`
- 表主体行分组：`<tbody></tbody>`
- 表尾行分组：`<tfoot></tfoot>`
  - 包含一个或者多个 `<tr>` 元素

知识讲解

**行分组（续1）**

```
<table border="1" width="300 " >
<thead align="center">
<tr>
<td>姓名</td>
<td>年龄</td>
</tr>
</thead>
<tbody align="right">
<tr>
<td>mary</td>
<td>18</td>
</tr>
<tr>
<td>jerry</td>
<td>20</td>
</tr>
</tbody>
</table>
```

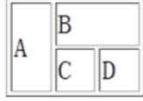
姓名	年龄
mary	18
jerry	20

#### 5.3.2. 【表格的复杂应用】不规则表格

知识讲解

**不规则表格**

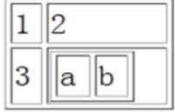
- 设置单元格 `<td>` 的跨行或者跨列属性
- 跨列：`colspan`
  - 水平方向延伸单元格，值为一正整数，代表此单元格水平延伸的单元格数
- 跨行：`rowspan`
  - 垂直方向延伸单元格，值为一正整数，代表此单元格垂直延伸的单元格数

A单元格跨行

B单元格跨列

### 5.3.3. 【表格的复杂应用】表格的嵌套

**表格的嵌套**

- 嵌套表格
  - 在单元格中放置另外一个表
  - 即 <td> 元素中再包含 <table> 元素
- 使用嵌套的表格以设计复杂表格或者复杂布局



嵌套的表格

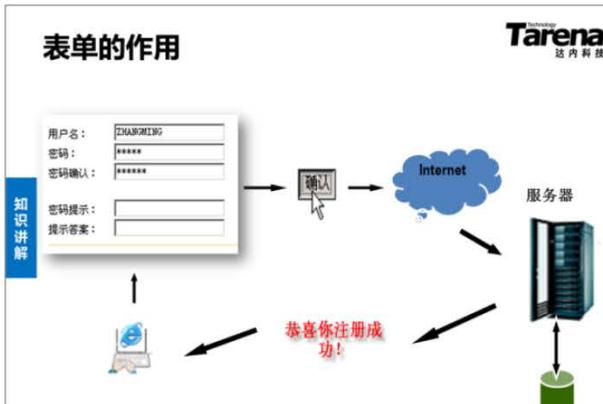
知识讲解
+<br/>练习

## 6. 表单

### 6.1. 表单概述

#### 6.1.1. 【表单概述】表单的作用

**表单的作用**



知识讲解
+<br/>练习

**表单的作用 (续1)**

- 表单用于显示、收集信息，并提交信息到服务器
- 表单有两个基本部分
  - 实现数据交互的可见的界面元素，比如文本框或按钮
  - 提交后的表单处理
- 界面元素
  - 使用 <form> 元素创建表单
  - 在 <form> 元素中添加其他表单可以包含的控件元素

知识讲解
+<br/>练习

### 6.1.2. 【表单概述】表单元素 <form>

**表单元素 <form>**

**知识讲解**

- 定义表单：使用成对的 <form></form> 标记
- 主要属性
  - action：定义表单被提交时发生动作，通常包含服务端脚本的URL（比如JSP、PHP）
  - method：指出表单数据提交的方式，取值为 get 或者 post
  - enctype：表单数据进行编码的方式
  - name：表单名称

+

### 6.1.3. 【表单概述】表单控件

**表单控件**

**知识讲解**

- 表单可以包含很多不同类型的表单控件
- 表单控件元素是包含在表单元素中具有可视化外观的 HTML 元素，用于访问者输入信息
- 表单控件元素有
  - input 元素：文本输入控件、按钮、单选和复选按钮、选项框、文件选择框和隐藏控件等
  - textarea 元素
  - select 和 option 元素
  - 其他元素

+

## 6.2. <input> 元素

### 6.2.1. 【<input> 元素】<input> 元素

**<input> 元素**

**知识讲解**

- <input> 元素用于收集用户信息
- 该元素是一个单标记，语法为：<input />
- 主要属性
  - type：根据不同的 type 属性值，可以创建各种类型的输入字段，比如文本框、复选框等
  - value：控件的数据
  - name：控件的名称

+

### 6.2.2. 【<input> 元素】文本框与密码框

**文本框与密码框**

• 文本框 : <input type = "text" />

• 密码框 : <input type = "password" />

• 主要属性

- value 属性 : 由访问者自由输入的任何文本
- maxlength 属性 : 限制输入的字符数
- readonly 属性 : 设置文本控件只读

姓名:

密码:

### 6.2.3. 【<input> 元素】单选框和复选框

**单选框和复选框**

• 单选框 : <input type= "radio" />

• 复选框 : <input type= "checkbox" />

• 主要属性

- value : 文本 , 当提交form时 , 如果选中了此单选按钮 ,  
那么value就被发送到服务器
- name : 用于实现分组 , 一组单选框或者复选框的名称必  
须相同
- checked : 设置选中

性别:  女士  男士

角色:  超级管理员  账单管理员

### 6.2.4. 【<input> 元素】按钮

**按钮**

• 提交按钮 : <input type= "submit" />

- 传送表单数据给服务器端或其它程序处理

• 重置按钮 : <input type= "reset" />

- 清空表单的内容并把所有表单控件设置为最初的默认值

• 普通按钮 : <input type= "button" />

- 用于执行客户端脚本

• 主要属性

- value : 按钮的标题文本

### 6.2.5. 【<input> 元素】隐藏域和文件选择框

**隐藏域和文件选择框**

- 隐藏域 : <input type= "hidden" />
  - 在表单中包含不希望用户看见的信息
- 文件选择框 : <input type= "file" />
  - 选择要上传的文件

浏览...

知识讲解

+

达内IT培训

21

## 经典案例

### 1. 创建 HTML 文档并设置头元素

- 问题

创建一个标准结构的 HTML 文档，并为创建的 HTML 文档添加文档标题，设置文档的编码格式以及页面的刷新频率。

- 方案

使用 HTML 编写网页文档时，首先需要在文档的开始添加文档类型声明。文档类型声明有三种：过渡型、严格型和框架型。

目前，为了保证兼容性，大多数页面都是使用过渡型，其声明为：

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

除了文档类型信息，剩下的就是整个页面的内容。HTML 文档的整个页面需要包含在开始标记 `<html>` 和结束标记 `</html>` 之间。在 `<html>` 元素中，包含两个主要的部分，文件头部部分（`<head>`元素）和主体部分（`<body>`元素）。代码如下所示：

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
  </head>
  <body>
  </body>
</html>
```

- 步骤

实现此案例需要按照如下步骤进行。

#### 步骤一：创建文件

首先创建一个纯文本文件，并修改名称为 `01_doc.html`；

#### 步骤二：创建 html 文档的结构

使用文本编辑工具，打开上一步中所创建的文件。

在此文件中，添加 HTML 代码，以创建一个标准结构的 HTML 文档（使用过渡型文档类型声明）。代码如下所示：

```
<!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>
  </head>
  <body>
  </body>
</html>
```

```
</html>
```

### 步骤三：设置头部内容

为 `<head>` 元素添加内容，为创建的 HTML 文档添加文档标题，并设置文档的编码格式以及页面的刷新频率。代码如下所示：

```
<!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>

        <title>第一个 HTML 文档</title>
        <meta http-equiv="content-type" content="text/html; charset=utf-8" />
        <meta http-equiv="refresh" content="10" />

    </head>
    <body>
        </body>
</html>
```

- **完整代码**

本案例的完整代码如下所示：

```
<!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>
        <title>第一个 HTML 文档</title>
        <meta http-equiv="content-type" content="text/html; charset=utf-8" />
        <meta http-equiv="refresh" content="10" />
    </head>
    <body>
        </body>
</html>
```

## 2. 使用文本标记为 HTML 页面添加内容

- **问题**

修改上一个案例中所创建的 `01_doc.html` 文档，为页面添加文本标记，实现如图 - 1 所示的页面效果：



图 - 1

## • 方案

图 - 1 所示的页面中，文本 “Java 语言基础 <Day03>” 为一级标题，使用 `<h1>` 元素创建。其中，特殊字符部分（如 `<` 和 `>`）需要使用转义符，而红色文本需要使用 `<span>` 元素标识出来，并添加样式。代码如下：

```
<h1 align="center">
    Java 语言基础<span style="color:Red;">&lt;Day03&gt;</span>
</h1>
```

文本 “1 个人所得税计算器” 为二级标题，使用 `<h2>` 元素创建；文本 “1.1 问题”、“1.2 方案” 和 “1.3 实现” 为三级标题，使用 `<h3>` 元素创建。其中，一个空格的文本，可以直接使用空格，或者使用转义符 `&ampnbsp`。代码如下所示：

```
<h2>1 个人所得税计算器</h2>
<h3>1.1&ampnbsp问题</h3>
```

其他文本使用段落元素 `<p>` 创建，空白行使用换行元素 `<br>` 创建。

## • 步骤

实现此案例需要按照如下步骤进行。

## 步骤一：创建第一行文本

使用文本编辑器打开上一个案例中的文件 01\_doc.html，并在 <body> 元素中，使用 <h1> 元素创建第一行文本。代码如下所示：

```
<!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>
    <title>第一个 HTML 文档</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta http-equiv="refresh" content="10" />
</head>
<body>
    <h1 align="center">
        Java&nbsp;语言基础 <span style="color:Red;">&lt;Day03&gt;</span>
    </h1>
</body>
</html>
```

## 步骤二：创建其他行文本

继续使用 <h2>、<h3> 和 <p> 元素创建其他行文本，代码如下所示：

```
<!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>
    <title>第一个 HTML 文档</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta http-equiv="refresh" content="10" />
</head>
<body>
    <h1 align="center">
        Java&nbsp;语言基础 <span style="color:Red;">&lt;Day03&gt;</span>
    </h1>

    <h2>1&nbsp;个人所得税计算器</h2>
    <h3>1. 1&nbsp;问题</h3>
    <p>计算个人所得税的缴纳情况。用户从控制台输入税前工资的金额，系统根据用户输入的工  
资金额计算应缴纳的税额。</p>
    <h3>1. 2&nbsp;方案</h3>
    <p>使用 if 语句来完成该程序。</p>
    <h3>1. 3&nbsp;实现</h3>
    <p>使用记事本建立一个名称为 IncomeTax.java 的文本文件。定义一个类，  
代码如下：</p>

</body>
</html>
```

## 步骤三：创建空白行

使用 <br> 元素创建空白行。代码如下所示：

```

<!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>
    <title>第一个 HTML 文档</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta http-equiv="refresh" content="10" />
</head>
<body>
    <h1 align="center">
        Java&nbsp;语言基础 <span style="color:Red;">&lt;Day03&gt;</span>
    </h1>
    <h2>1&nbsp;个人所得税计算器</h2>
    <h3>1.1&nbsp;问题</h3>
    <p>计算个人所得税的缴纳情况。用户从控制台输入税前工资的金额，系统根据用户输入的工资本金额计算应缴纳的税额。</p>
    <h3>1.2&nbsp;方案</h3>
    <p>使用 if 语句来完成该程序。</p>
    <h3>1.3&nbsp;实现</h3>
    <p>使用记事本建立一个名称为 IncomeTax.java 的文本文件。定义一个类，代码如下：</p>

    <p>
        public&nbsp;class&nbsp;IncomeTax<br />
        {<br /><br /><br /><br />}
    </p>

</body>
</html>

```

## • 完整代码

本案例的完整代码如下所示：

```

<!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>
    <title>第一个 HTML 文档</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta http-equiv="refresh" content="10" />
</head>
<body>
    <h1 align="center">
        Java&nbsp;语言基础 <span style="color:Red;">&lt;Day03&gt;</span>
    </h1>
    <h2>1&nbsp;个人所得税计算器</h2>
    <h3>1.1&nbsp;问题</h3>
    <p>计算个人所得税的缴纳情况。用户从控制台输入税前工资的金额，系统根据用户输入的工资本金额计算应缴纳的税额。</p>
    <h3>1.2&nbsp;方案</h3>
    <p>使用 if 语句来完成该程序。</p>
    <h3>1.3&nbsp;实现</h3>
    <p>使用记事本建立一个名称为 IncomeTax.java 的文本文件。定义一个类，代码如下：</p>
    <p>
```

```
public class IncomeTax<br />
{<br /><br /><br /><br />
}
</p>
</body>
</html>
```

### 3. 为 HTML 页面添加图像和链接

- 问题

继续修改上节中所创建的 01\_doc.html 文档，为图 - 1 所示的页面中的“1.1 问题”部分添加图像，页面效果如图 - 2 所示。



图 - 2

然后，为图 - 1 所示的页面中的“1.2 方案”部分添加图像，页面效果如图 - 3 所示。

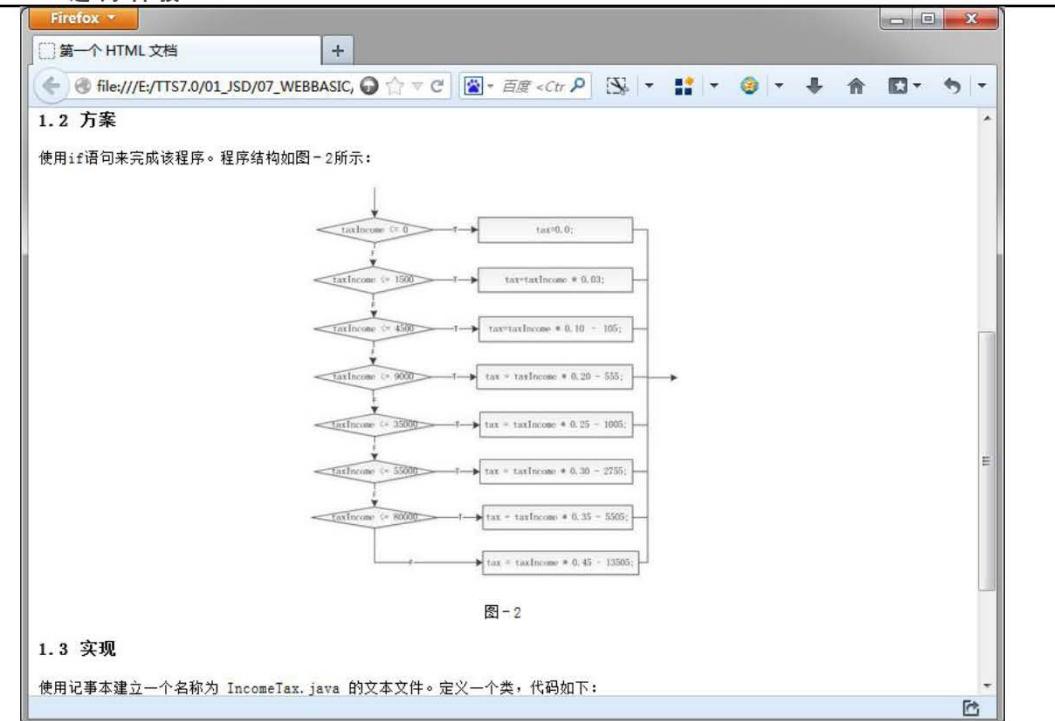


图 - 2

### 1.3 实现

使用记事本建立一个名称为 `IncomeTax.java` 的文本文件。定义一个类，代码如下：

图 - 3

并为图 - 2 和图 - 3 上的图像添加链接，单击该图像后，将在新的浏览器窗口（或者页签）中打开 TTS 的页面（地址为 <http://tts7.tarena.com.cn>）。

另外，页面底部需要添加链接“[To Top](#)”，单击该链接后，将定位到当前页面的顶端。

## • 方案

使用元素 `<img>` 为页面添加图像，使用 `<a>` 元素为图像添加链接。为了在新窗口或者新的页签上打开页面，需要设置 `<a>` 元素的 `target` 属性的值为 `_blank`。代码如下所示：

```
<a href="http://tts7.tarena.com.cn" target="_blank">
    
</a>
```

使用 `<p>` 元素设置图像下方的文本标题，并使用 `<div>` 元素作为图像和图像标题的父元素，以设置居中显示。代码如下所示：

```
<div align="center">
    <a href="http://tts7.tarena.com.cn" target="_blank">
        
    </a>
    <br />
    <p>图-1</p>
</div>
```

使用 `<a>` 元素为页面定义链接，以实现当前页面的链接定位。代码如下所示：

<a href="#">To Top</a>

## • 步骤

实现此案例需要按照如下步骤进行。

### 步骤一：定义带链接的图像

使用文本编辑器打开上一个案例中的文件 `01_doc.html`，并分别添加两个带链接的图像。图像和 HTML 文件位于同一个目录，因此，使用相对路径即可。

代码如下所示：

```

<!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>
        <title>第一个 HTML 文档</title>
        <meta http-equiv="content-type" content="text/html; charset=utf-8" />
        <meta http-equiv="refresh" content="10" />
    </head>
    <body>
        <h1 align="center">
            Java&nbsp;语言基础 <span style="color:Red;">&lt;Day03&gt;</span>
        </h1>
        <h2>1&nbsp;个人所得税计算器</h2>
        <h3>1.1&nbsp;问题</h3>
        <p>计算个人所得税的缴纳情况。用户从控制台输入税前工资的金额，系统根据用户输入的工资本金额计算应缴纳的税额。运行界面如图 - 1 所示：</p>

        <div align="center">
            <a href="http://tts7.tarena.com.cn" target="_blank"></a><br />
            <p>图 - 1</p>
        </div>

        <h3>1.2&nbsp;方案</h3>

        <p>使用 if 语句来完成该程序。程序结构如图 - 2 所示：</p>
        <div align="center">
            <a href="http://tts7.tarena.com.cn" target="_blank"></a><br />
            <p>图 - 2</p>
        </div>

        <h3>1.3&nbsp;实现</h3>
        <p>使用记事本建立一个名称为 IncomeTax.java 的文本文件。定义一个类，代码如下：</p>
        <p>
            public class IncomeTax<br />
            {<br /><br /><br /><br />
            }
        </p>
    </body>
</html>

```

## 步骤二：创建锚点

在页面底部，使用 `<a>` 元素定义锚点，用于导航到页面顶端。代码如下所示：

```

<!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>
    <title>第一个 HTML 文档</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta http-equiv="refresh" content="10" />
</head>
<body>
    <h1 align="center">
        Java语言基础 <span style="color:Red;">&lt;Day03&gt;</span>
    </h1>
    <h2>1.个人所得税计算器</h2>
    <h3>1.1问题</h3>
    <p>计算个人所得税的缴纳情况。用户从控制台输入税前工资的金额，系统根据用户输入的工资本金额计算应缴纳的税额。运行界面如图 - 1 所示：</p>
    <div align="center">
        <a href="http://tts7.tarena.com.cn" target="blank"></a><br />
        <p>图 - 1</p>
    </div>
    <h3>1.2方案</h3>
    <p>使用 if 语句来完成该程序。程序结构如图 - 2 所示：</p>
    <div align="center">
        <a href="http://tts7.tarena.com.cn" target="blank"></a><br />
        <p>图 - 2</p>
    </div>
    <h3>1.3实现</h3>
    <p>使用记事本建立一个名称为 IncomeTax.java 的文本文件。定义一个类，代码如下：</p>
    <p>
        public class IncomeTax<br />
        {<br /><br /><br />
    </p>
    <a href="#">To Top</a>

    </body>
</html>

```

### • 完整代码

本案例的完整代码如下所示：

```

<!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>
    <title>第一个 HTML 文档</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />

```

```
<meta http-equiv="refresh" content="10" />
</head>
<body>
    <h1 align="center">
        Java语言基础 <span style="color:Red;">&lt;Day03&gt;</span>
    </h1>
    <h2>1.个人所得税计算器</h2>
    <h3>1.1.问题</h3>
    <p>计算个人所得税的缴纳情况。用户从控制台输入税前工资的金额，系统根据用户输入的工资本金额计算应缴纳的税额。运行界面如图-1 所示：</p>
    <div align="center">
        <a href="http://tts7.tarena.com.cn" target="_blank"></a><br />
        <p>图-1</p>
    </div>
    <h3>1.2.方案</h3>
    <p>使用 if 语句来完成该程序。程序结构如图-2 所示：</p>
    <div align="center">
        <a href="http://tts7.tarena.com.cn" target="_blank"></a><br />
        <p>图-2</p>
    </div>
    <h3>1.3.实现</h3>
    <p>使用记事本建立一个名称为 IncomeTax.java 的文本文件。定义一个类，代码如下：</p>
    <p>
        public class IncomeTax<br />
        {<br /><br /><br /><br />
        }
    </p>
    <a href="#">To Top</a>
</body>
</html>
```

## 4. 为 HTML 页面添加导航目录

- 问题

继续修改上一个案例中所创建的 01\_doc.html 文档，为页面添加导航用的目录，页面效果如图 - 4 所示。

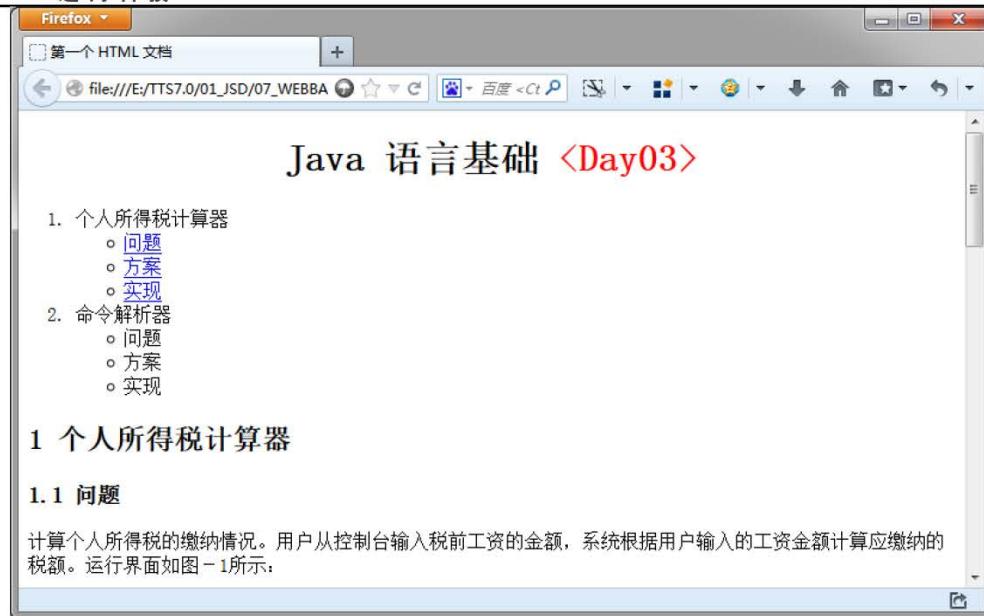


图 - 4

由图 - 4 可以看出，页面上添加了当前文档的目录，单击图 - 4 中的链接，将分别定位到文档中相应的位置。其中，仅“1.个人所得税计算器”目录下的链接可用，其他链接仅为演示示例而已。

比如，单击“1.个人所得税计算器”目录下的“方案”链接，则定位到文档中相应的位置，页面效果如图 - 5 所示。

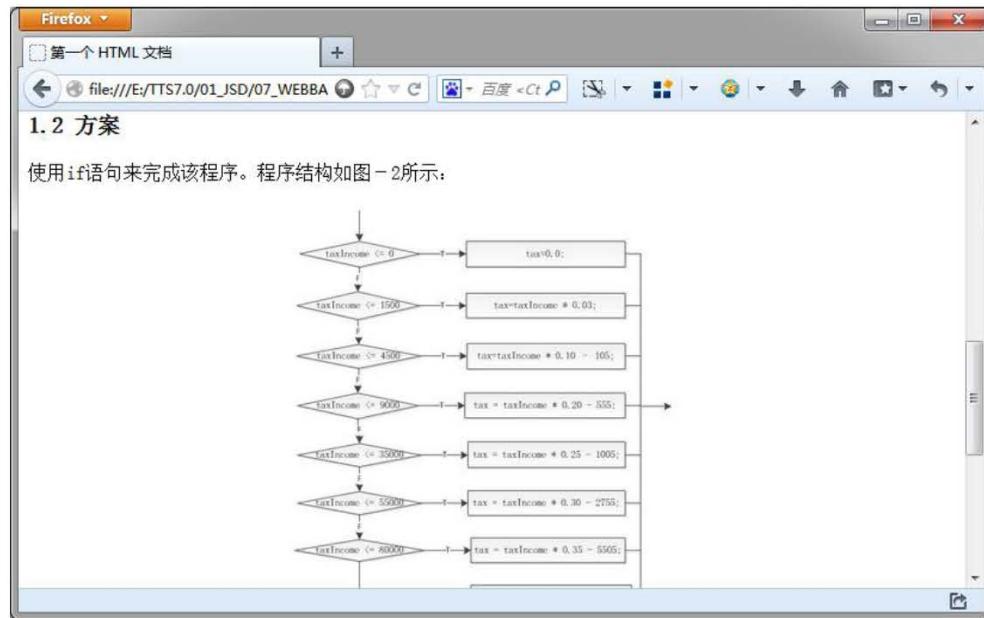


图 - 5

### • 方案

为实现当前页面上不同位置之间的跳转，需要使用锚点。

首先，需要在目标位置（跳往的目的地）使用 `<a>` 元素创建锚点。代码如下所示：

```
<h3><a name="question"></a>1.1&nbsp;问题</h3>
```

然后，使用链接文本，导航到锚点所代表的目标位置。代码如下所示：

```
<a href="#question">问题</a>
```

另外，创建导航目录时，需要使用嵌套的列表。代码如下所示：

```
<ol>
  <li>
    个人所得税计算器
    <ul>
      <li><a href="#question">问题</a></li>
      <li><a href="#project">方案</a></li>
      <li><a href="#code">实现</a></li>
    </ul>
  </li>
  <li>
    命令解析器
    <ul>
      <li>问题</li>
      <li>方案</li>
      <li>实现</li>
    </ul>
  </li>
</ol>
```

## • 步骤

实现此案例需要按照如下步骤进行。

### 步骤一：定义锚点

使用文本编辑器打开上一个案例中的文件 `01_doc.html`，修改原有 `<h3>` 元素中的代码，添加 `<a>` 元素以分别设置锚点。代码如下所示：

```
<!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>
    <title>第一个 HTML 文档</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta http-equiv="refresh" content="10" />
  </head>
  <body>
    <h1 align="center">
      Java&nbsp;语言基础 <span style="color:Red;">&lt;Day03&gt;</span>
    </h1>
    <h2>1&nbsp;个人所得税计算器</h2>

    <h3><a name="question"></a>1.1&nbsp;问题</h3>
```

<p>计算个人所得税的缴纳情况。用户从控制台输入税前工资的金额，系统根据用户输入的工资金额计算应缴纳的税额。运行界面如图 - 1 所示 :</p>

```

<div align="center">
    <a href="http://tts7.tarena.com.cn" target="_blank"></a><br />
    <p>图 - 1</p>
</div>

```

### <h3><a name="project"></a>1.2 &ampnbsp方案</h3>

<p>使用 if 语句来完成该程序。程序结构如图 - 2 所示 :</p>

```

<div align="center">
    <a href="http://tts7.tarena.com.cn" target="_blank"></a><br />
    <p>图 - 2</p>
</div>

```

### <h3><a name="code"></a>1.3 &ampnbsp实现</h3>

<p>使用记事本建立一个名称为 IncomeTax.java 的文本文件。定义一个类，代码如下 :</p>

```

<p>
    public class IncomeTax<br />
    {<br /><br /><br />
    }
</p>
<a href="#">To Top</a>
</body>
</html>

```

## 步骤二：定义导航目录

使用嵌套的列表定义导航目录，并在列表中，使用链接元素 <a> 定义链接。代码如下所示：

```

<!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>
    <title>第一个 HTML 文档</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta http-equiv="refresh" content="10" />
</head>
<body>
    <h1 align="center">
        Java &ampnbsp语言基础 <span style="color:Red;">&lt;Day03&gt;</span>
    </h1>

    <ol>
        <li>
            个人所得税计算器
            <ul>
                <li><a href="#question">问题</a></li>
                <li><a href="#project">方案</a></li>
            </ul>
        </li>
    </ol>

```

```

<li><a href="#code">实现</a></li>
</ul>
</li>
<li>
    命令解析器
    <ul>
        <li>问题</li>
        <li>方案</li>
        <li>实现</li>
    </ul>
</li>
</ol>

<h2>1.个人所得税计算器</h2>
<h3><a name="question"></a>1.1.问题</h3>
<p>计算个人所得税的缴纳情况。用户从控制台输入税前工资的金额，系统根据用户输入的工资本金额计算应缴纳的税额。运行界面如图 - 1 所示：</p>
<div align="center">
    <a href="http://tts7.tarena.com.cn" target="_blank"></a><br />
    <p>图 - 1</p>
</div>
<h3><a name="project"></a>1.2.方案</h3>
<p>使用 if 语句来完成该程序。程序结构如图 - 2 所示：</p>
<div align="center">
    <a href="http://tts7.tarena.com.cn" target="_blank"></a><br />
    <p>图 - 2</p>
</div>
<h3><a name="code"></a>1.3.实现</h3>
<p>使用记事本建立一个名称为 IncomeTax.java 的文本文件。定义一个类，代码如下：</p>
<p>
    public class IncomeTax<br />
    {<br /><br /><br />
    }
</p>
<a href="#">To Top</a>
</body>
</html>

```

### • 完整代码

本案例的完整代码如下所示：

```

<!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>
    <title>第一个 HTML 文档</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta http-equiv="refresh" content="10" />
</head>
<body>
    <h1 align="center">
        Java 语言基础 <span style="color:Red;">&lt;Day03&gt;</span>
    </h1>
</body>

```

```

</h1>
<ol>
    <li>
        个人所得税计算器
        <ul>
            <li><a href="#question">问题</a></li>
            <li><a href="#project">方案</a></li>
            <li><a href="#code">实现</a></li>
        </ul>
    </li>
    <li>
        命令解析器
        <ul>
            <li>问题</li>
            <li>方案</li>
            <li>实现</li>
        </ul>
    </li>
</ol>
<h2>1. 个人所得税计算器</h2>
<h3><a name="question"></a>1.1. 问题</h3>
<p>计算个人所得税的缴纳情况。用户从控制台输入税前工资的金额，系统根据用户输入的工资本金额计算应缴纳的税额。运行界面如图-1 所示：</p>
<div align="center">
    <a href="http://tts7.tarena.com.cn" target="_blank"></a><br />
    <p>图-1</p>
</div>
<h3><a name="project"></a>1.2. 方案</h3>
<p>使用 if 语句来完成该程序。程序结构如图-2 所示：</p>
<div align="center">
    <a href="http://tts7.tarena.com.cn" target="_blank"></a><br />
    <p>图-2</p>
</div>
<h3><a name="code"></a>1.3. 实现</h3>
<p>使用记事本建立一个名称为 IncomeTax.java 的文本文件。定义一个类，代码如下：</p>
<p>
    public class IncomeTax<br />
    {<br /><br /><br /><br />
    }
</p>
<a href="#">To Top</a>
</body>
</html>

```

## 5. 为 HTML 页面添加数据表格

- 问题

继续修改上一个案例中所创建的 `01_doc.html` 文档，为页面添加表格以说明个人所得税的计算方式。表格在页面上的效果如图 - 6 所示。

个人所得税的计算方式如表 - 1 所示：

表 - 1

全月应纳税所得额	税率	速算扣除数
全月应纳税所得额<1500	3%	0
1500=<全月应纳税所得额<4500	10%	105
...		
计算公式为： 应纳税额=全月应纳税所得额*适用税率-速算扣除数		

## 1.2 方案

使用 if 语句来完成该程序。程序结构如图 - 2 所示：



图 - 6

### • 方案

分析图 - 6 中所示表格，可以看出，需要为页面创建一个 5 行的表格，表格结构如图 - 7 所示：


图 - 7

对于图 - 7 所示的表格，首先需要使用 `<caption>` 元素定义表格的标题；其次，为第一行定义背景颜色为 silver；第三，为方便后四行的样式统一定义，可以使用 `<tbody>` 元素定义后四行。表格结构的代码如下所示：

```





```

```

<tr>
    ...
</tr>
<tr>
    ...
</tr>
</tbody>
</table>
    
```

另外，第 4 行和第 5 行中的三个单元格需要跨列合并为一个单元格，代码如下所示：

```
<td colspan="3">...</td>
```

而为了在第 5 行中实现文本“计算公式为：”和计算公式之间的间隔，需要使用嵌套的表格。代码如下所示：

```

<td colspan="3">
    <table>
        <tr>
            <td width="120">计算公式为：</td>
            <td>应纳税额 = 全月应纳税所得额 * 适用税率 - 速算扣除数</td>
        </tr>
    </table>
</td>
    
```

### • 步骤

实现此案例需要按照如下步骤进行。

#### 步骤一：定义表格的基本结构

使用文本编辑器打开上一个案例中的文件 `01_doc.html`，为页面添加代码，以定义表格的基本结构。代码如下所示：

```

<!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>
        <title>第一个 HTML 文档</title>
        <meta http-equiv="content-type" content="text/html; charset=utf-8" />
        <meta http-equiv="refresh" content="10" />
    </head>
    <body>
        <h1 align="center">
            Java&ampnbsp语言基础 <span style="color:Red;">&lt;Day03&gt;</span>
        </h1>
        <ol>
            <li>
                个人所得税计算器
                <ul>
                    <li><a href="#question">问题</a></li>
                    <li><a href="#project">方案</a></li>
                    <li><a href="#code">实现</a></li>
                </ul>
            </li>
            <li>
                ...
            </li>
        </ol>
    </body>
</html>
    
```

**命令解析器**

```

<ul>
    <li>问题</li>
    <li>方案</li>
    <li>实现</li>
</ul>
</li>
</ol>
<h2>1.1 个人所得税计算器</h2>
<h3><a name="question"></a>1.1 问题</h3>
<p>计算个人所得税的缴纳情况。用户从控制台输入税前工资的金额，系统根据用户输入的工资本金额计算应缴纳的税额。运行界面如图 - 1 所示：</p>
<div align="center">
    <a href="http://tts7.tarena.com.cn" target="_blank"></a><br />
    <p>图 - 1</p>
</div>

<p>个人所得税的计算方式如表 - 1 所示：</p>
<table border="1" width="600" cellpadding="5" align="center">
    <caption>表 - 1</caption>
    <tr style="background-color:Silver;">
        </tr>
        <tr>
            <td style="color:gray;">
                <tr>
                    </tr>
                    <tr>
                        <td>
                            <tr>
                                </tr>
                                <tr>
                                    <td>
                                        <tr>
                                            </tr>
                                            <tr>
                                                <td>
                                                    <tr>
                                                        </tr>
                                                        <tr>
                                                            <td>
                                                                <tr>
                                                                    </tr>
                                                                    <tr>
                                                                        <td>
                                                                            <tr>
                                                                                </tr>
                                                                                <tr>
                                                                                    <td>
                                                                                        <tr>
                                                                                            </tr>
                                                                                            <tr>
                                                                                                <td>
                                                                                                    <tr>
                                                                                                        </tr>
                                                                                                        <tr>
                                                                                                            <td>
                                                                                                                <tr>
                                                                                                                    </tr>
                                                                                                                    <tr>
                                                                                                                        <td>
                                                                                                                            <tr>
                                                                                                                                </tr>
                                                                                                                                <tr>
                                                                                                                                    <td>
                                                                                                                                        <tr>
                                                                                                                                            </tr>
                                                                                                                                            <tr>
                                                                                                                                                <td>
                                                                                                                                                    <tr>
                                                                ................................................................

```

<h3><a name="project"></a>1.2 方案</h3>

<p>使用 if 语句来完成该程序。程序结构如图 - 2 所示：</p>

```

<div align="center">
    <a href="http://tts7.tarena.com.cn" target="_blank"></a><br />
    <p>图 - 2</p>
</div>

```

<h3><a name="code"></a>1.3 实现</h3>

<p>使用记事本建立一个名称为 IncomeTax.java 的文本文件。定义一个类，代码如下：</p>

```

<p>
    public class IncomeTax<br />
    {<br /><br /><br /><br />
    }
</p>
<a href="#">To Top</a>
</body>
</html>

```

## 步骤二：创建表格的第一行

为表格创建第一行的内容：定义单元格，并分别设置单元格的宽度以设置列宽。代码如下：

```
<!--其他部分代码，略-->
<table border="1" width="600" cellpadding="5" align="center">
    <caption>表 - 1</caption>

    <tr style="background-color:Silver;">
        <td width="350">全月应纳税所得额</td>
        <td width="100">税率</td>
        <td width="100">速算扣除数</td>
    </tr>

    <tbody style="color:gray; ">
        <tr>
        </tr>
        <tr>
        </tr>
        <tr>
        </tr>
        <tr>
        </tr>
        <tr>
        </tr>
    </tbody>
</table>
<!--其他部分代码，略-->
```

### 步骤三：创建表格的第二行和第三行

继续创建表格的第二行和第三行。代码如下所示：

```
<!--其他部分代码，略-->
<table border="1" width="600" cellpadding="5" align="center">
    <caption>表 - 1</caption>
    <tr style="background-color:Silver;">
        <td width="350">全月应纳税所得额</td>
        <td width="100">税率</td>
        <td width="100">速算扣除数</td>
    </tr>
    <tbody style="color:gray; ">

        <tr>
            <td>全月应纳税所得额<1500</td>
            <td>3%</td>
            <td>0</td>
        </tr>
        <tr>
            <td>1500=<1500</td>
            <td>10%</td>
            <td>105</td>
        </tr>

        <tr>
        </tr>
        <tr>
        </tr>
    </tbody>
</table>
```

```

    </tr>
</tbody>
</table>
<!--其他部分代码，略-->

```

#### 步骤四：创建表格的第四行

定义表格的第四行。此行中只有一个单元格，且该单元格跨三列。代码如下所示：

```

<!--其他部分代码，略-->
<table border="1" width="600" cellpadding="5" align="center">
  <caption>表 - 1</caption>
  <tr style="background-color:Silver;">
    <td width="350">全月应纳税所得额</td>
    <td width="100">税率</td>
    <td width="100">速算扣除数</td>
  </tr>
  <tbody style="color:gray;">
    <tr>
      <td>全月应纳税所得额<1500</td>
      <td>3%</td>
      <td>0</td>
    </tr>
    <tr>
      <td>1500=<1500</td>
      <td>10%</td>
      <td>105</td>
    </tr>

    <tr>
      <td colspan="3">...</td>
    </tr>

    <tr>
    </tr>
  </tbody>
</table>
<!--其他部分代码，略-->

```

#### 步骤五：创建表格的第五行

表格的第五行也只有一个单元格（跨3列），且此单元格中需要使用嵌套的表格来实现页面的布局。代码如下所示：

```

<!--其他部分代码，略-->
<table border="1" width="600" cellpadding="5" align="center">
  <caption>表 - 1</caption>
  <tr style="background-color:Silver;">
    <td width="350">全月应纳税所得额</td>
    <td width="100">税率</td>
    <td width="100">速算扣除数</td>
  </tr>
  <tbody style="color:gray;">
    <tr>

```

```

<td>全月应纳税所得额<1500</td>
<td>3%</td>
<td>0</td>
</tr>
<tr>
    <td>1500<4500</td>
    <td>10%</td>
    <td>105</td>
</tr>
<tr>
    <td colspan="3">...</td>
</tr>

<tr>
    <td colspan="3">
        <table>
            <tr>
                <td width="120">计算公式为 :</td>
                <td>应纳税额 = 全月应纳税所得额*适用税率-速算扣除数</td>
            </tr>
        </table>
    </td>
</tr>

</tbody>
</table>
<!--其他部分代码，略-->

```

### • 完整代码

本案例的完整代码如下所示：

```

<!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>
        <title>第一个 HTML 文档</title>
        <meta http-equiv="content-type" content="text/html; charset=utf-8" />
        <meta http-equiv="refresh" content="10" />
    </head>
    <body>
        <h1 align="center">
            Java语言基础 <span style="color:Red;">&lt;Day03&gt;</span>
        </h1>
        <ol>
            <li>
                个人所得税计算器
                <ul>
                    <li><a href="#question">问题</a></li>
                    <li><a href="#project">方案</a></li>
                    <li><a href="#code">实现</a></li>
                </ul>
            </li>
            <li>
                命令解析器
                <ul>
                    <li>问题</li>
                    <li>方案</li>
                    <li>实现</li>
                </ul>
            </li>
        </ol>
    </body>
</html>

```

```

        </li>
    </ol>
    <h2>1.个人所得税计算器</h2>
    <h3><a name="question"></a>1.1.问题</h3>
    <p>计算个人所得税的缴纳情况。用户从控制台输入税前工资的金额，系统根据用户输入的工资金额计算应缴纳的税额。运行界面如图-1 所示：</p>
    <div align="center">
        <a href="http://tts7.tarena.com.cn" target="_blank"></a><br />
        <p>图-1</p>
    </div>
    <p>个人所得税的计算方式如表-1 所示：</p>
    <table border="1" width="600" cellpadding="5" align="center">
        <caption>表-1</caption>
        <tr style="background-color:Silver;">
            <td width="350">全月应纳税所得额</td>
            <td width="100">税率</td>
            <td width="100">速算扣除数</td>
        </tr>
        <tbody style="color:gray;">
            <tr>
                <td>全月应纳税所得额<1500</td>
                <td>3%</td>
                <td>0</td>
            </tr>
            <tr>
                <td>1500=<1500</td>
                <td>10%</td>
                <td>105</td>
            </tr>
            <tr>
                <td colspan="3">...</td>
            </tr>
            <tr>
                <td colspan="3">
                    <table>
                        <tr>
                            <td width="120">计算公式为：</td>
                            <td>应纳税额=全月应纳税所得额*适用税率-速算扣除数
                        </td>
                    </tr>
                </table>
            </td>
        </tr>
    </tbody>
    </table>
    <h3><a name="project"></a>1.2.方案</h3>
    <p>使用 if 语句来完成该程序。程序结构如图-2 所示：</p>
    <div align="center">
        <a href="http://tts7.tarena.com.cn" target=" blank"></a><br />
        <p>图-2</p>
    </div>
    <h3><a name="code"></a>1.3.实现</h3>
    <p>使用记事本建立一个名称为 IncomeTax.java 的文本文件。定义一个类，代码如下：</p>
    <p>
        public class IncomeTax<br />
        {<br /><br /><br /><br />
    </p>
    <a href="#">To Top</a>
</body>
</html>

```

## 6. 创建基础表单

- 问题

创建页面文档 02\_form.html ,并为页面添加表单元素 ,实现如图 - 8 所示的页面效果。

The screenshot shows a Firefox browser window with the title bar "Firefox" and a tab labeled "基础表单". The main content area displays a form titled "增加管理员" (Add Admin). The form fields are as follows:

- 姓名:  10个字符以内
- 密码:  10个字符以内
- 性别:  女士  男士
- 角色:  超级管理员  账单管理员 至少选择一个角色
- 头像:  未选择文件。
- 按钮:

图 - 8

- 方案

首先使用表格实现图 - 8 所示的页面布局 , 表格布局形如图 - 9 所示 :

	文件框和按钮	
	文件框和按钮	

图 - 9

图 - 9 所示的表格中 , 第一列用于显示文本信息 , 如 “姓名”、“密码” 等 ; 第二列用于显示文本框、单选按钮和复选框按钮等 ; 第三列用于显示录入的提示信息 , 如“ 10 个字符以内”。第 5 行和第 6 行的后两列需要合并 , 用于显示文件框和按钮。表格的代码如下 :

```

<table width="500">
  <tr>
    <td width="70" height="35"> </td>
    <td width="180"></td>
    <td></td>
  </tr>
  <tr>
    <td height="35"> </td>
    <td></td>
    <td></td>
  </tr>
  <tr>
    <td height="35"> </td>
    <td></td>
    <td></td>
  </tr>
  <tr>
    <td height="55"> </td>
    <td></td>
    <td></td>
  </tr>
  <tr>
    <td height="55"> </td>
    <td colspan="2"></td>
  </tr>
  <tr>
    <td height="35"></td>
    <td colspan="2"></td>
  </tr>
</table>

```

然后，使用相应的标记添加页面元素，如文本框、密码框等。

## • 步骤

实现此案例需要按照如下步骤进行。

### 步骤一：创建 HTML 文档

首先创建一个纯文本文件，并修改名称为 `02_form.html`；然后使用文本编辑工具，打开该文件，并添加 HTML 代码，以创建一个标准结构的 HTML 文档（使用过渡型文档类型声明）。

因为此案例中，需要使用表单控件，因此，需要在 `<body>` 元素中添加 `<form>` 元素。

代码如下所示：

```

<!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>
    <title>基础表单</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  </head>
  <body>
    <form action="" method="">
    </form>
  </body>
</html>

```

### 步骤二：定义页面布局

在 `<form>` 元素中添加代码，创建页面的标题文本以及布局用的表格。代码如下所示：

```
<!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>
    <title>基础表单</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
</head>
<body>
    <form action="" method="">

        <h2>增加管理员</h2>
        <table width="500">
            </table>

    </form>
</body>
</html>
```

### 步骤三：创建表格的第一行

为表格创建第一行的内容：录入姓名的行。定义三个单元格，并分别创建单元格中的内容。其中，文本框使用 `<input>` 元素创建。代码如下所示：

```
<!--其他部分代码，略-->
<table width="500">

    <tr>
        <td width="70" height="35">姓名 : </td>
        <td width="180"><input /></td>
        <td><span>10个字符以内</span></td>
    </tr>

</table>
<!--其他部分代码，略-->
```

### 步骤四：创建表格的第二行

为表格创建第二行的内容：录入密码的行。定义三个单元格，并分别创建单元格中的内容。其中，密码框使用 `<input>` 元素创建，并设置 `type` 属性为 `password`。代码如下所示：

```
<!--其他部分代码，略-->
<table width="500">
    <tr>
        <td width="70" height="35">姓名 : </td>
        <td width="180"><input /></td>
        <td><span>10个字符以内</span></td>
    </tr>

    <tr>
```

```

<td height="35">密码 : </td>
<td><input type="password" /></td>
<td><span>10 个字符以内</span></td>
</tr>

</table>
<!--其他部分代码，略-->

```

### 步骤五：创建表格的第三行

为表格创建第三行的内容：选择性别的行。定义三个单元格，并分别创建单元格中的内容。其中，单选框使用 `<input>` 元素创建，并设置 `type` 属性为 `radio`。为了使得两个单选框作为一组，需要设置其 `name` 属性。代码如下所示：

```

<!--其他部分代码，略-->
<table width="500">
  <tr>
    <td width="70" height="35">姓名 : </td>
    <td width="180"><input /></td>
    <td><span>10 个字符以内</span></td>
  </tr>
  <tr>
    <td height="35">密码 : </td>
    <td><input type="password" /></td>
    <td><span>10 个字符以内</span></td>
  </tr>

  <tr>
    <td height="35">性别 : </td>
    <td>
      <input type="radio" name="sex" />女士
      <input type="radio" name="sex" />男士
    </td>
    <td></td>
  </tr>

</table>
<!--其他部分代码，略-->

```

### 步骤六：创建表格的第四行

为表格创建第四行的内容：选择角色的行。定义三个单元格，并分别创建单元格中的内容。其中，复选框使用 `<input>` 元素创建，并设置 `type` 属性为 `checkbox`。为了使得两个复选框作为一组，需要设置其 `name` 属性为相同的值。代码如下所示：

```

<!--其他部分代码，略-->
<table width="500">
  <tr>
    <td width="70" height="35">姓名 : </td>
    <td width="180"><input /></td>
    <td><span>10 个字符以内</span></td>
  </tr>

```

```

</tr>
<tr>
  <td height="35">密码 :</td>
  <td><input type="password" /></td>
  <td><span>10 个字符以内</span></td>
</tr>
<tr>
  <td height="35">性别 :</td>
  <td>
    <input type="radio" name="sex" />女士
    <input type="radio" name="sex" />男士
  </td>
  <td></td>
</tr>

<tr>
  <td height="55">角色 :</td>
  <td>
    <input type="checkbox" name="roles" />超级管理员<br />
    <input type="checkbox" name="roles" />账单管理员
  </td>
  <td><span>至少选择一个角色</span></td>
</tr>

</table>
<!--其他部分代码，略-->

```

## 步骤七：创建表格的第五行

为表格创建第五行的内容 选择头像的行。此行只有两个单元格 第二个单元格跨两列，为合并单元格。其中，文件选择框使用 `<input>` 元素创建，并设置 `type` 属性为 `file`。代码如下所示：

```

<!--其他部分代码，略-->





```

```

<td height="55">角色 :</td>
<td>
    <input type="checkbox" name="roles" />超级管理员<br />
    <input type="checkbox" name="roles" />账单管理员
</td>
<td><span>至少选择一个角色</span></td>
</tr>

<tr>
    <td height="55">头像 :</td>
    <td colspan="2"><input type="file" /></td>
</tr>

</table>
<!--其他部分代码，略-->

```

## 步骤八：创建表格的第六行

为表格创建第六行的内容：操作按钮的行。此行也只有两个单元格：第二个单元格跨两列，为合并单元格。其中，保存按钮使用 `<input>` 元素创建，并设置 `type` 属性为 `submit`；重填按钮使用 `<input>` 元素创建，并设置 `type` 属性为 `reset`。代码如下所示：

```

<!--其他部分代码，略-->


||
||
||
||
||
||


```

```

</tr>

<tr>
    <td height="35"></td>
    <td colspan="2">
        <input type="submit" value="保存" />
        <input type="reset" value="重填" />
    </td>
</tr>

</table>
<!--其他部分代码，略-->

```

### • 完整代码

本案例的完整代码如下所示：

```

<!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>
    <title>基础表单</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
</head>
<body>
    <form action="" method="">
        <h2>增加管理员</h2>
        <table width="500">
            <tr>
                <td width="70" height="35">姓名:</td>
                <td width="180"><input /></td>
                <td><span>10个字符以内</span></td>
            </tr>
            <tr>
                <td height="35">密码:</td>
                <td><input type="password" /></td>
                <td><span>10个字符以内</span></td>
            </tr>
            <tr>
                <td height="35">性别:</td>
                <td>
                    <input type="radio" name="sex" />女士
                    <input type="radio" name="sex" />男士
                </td>
                <td></td>
            </tr>
            <tr>
                <td height="55">角色:</td>
                <td>
                    <input type="checkbox" name="roles" />超级管理员<br />
                    <input type="checkbox" name="roles" />账单管理员
                </td>
                <td><span>至少选择一个角色</span></td>
            </tr>
            <tr>
                <td height="55">头像:</td>
                <td colspan="2"><input type="file" /></td>
            </tr>
            <tr>
                <td height="35"></td>

```

```
<td colspan="2">
    <input type="submit" value="保存" />
    <input type="reset" value="重填" />
</td>
</tr>
</table>
</form>
</body>
</html>
```

达内IT培训集团

## 课后作业

1. 简要描述标准 HTML 文档的结构
2. 简要描述行内元素和块级元素的区别
3. 锚点的作用是什么？如何创建锚点？
4. 创建如下图所示的表格

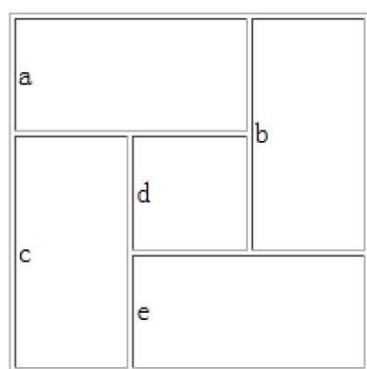


图 - 1

5. 创建如下图所示的页面

Screenshot of a Firefox browser window displaying a job application form. The title bar says "Firefox". The address bar shows "file:///E:/form". The page content is as follows:

--->>求职申请<<---

用户名:

密码:

性别:  男  女  不透露

头像:  未选择文件。

不要公开我的信息

图 - 2

## 6. 创建如下图所示的页面（提高题，选做）

有一个 HTML 页面，用于介绍三种花卉，其页面效果如图 - 3 所示：

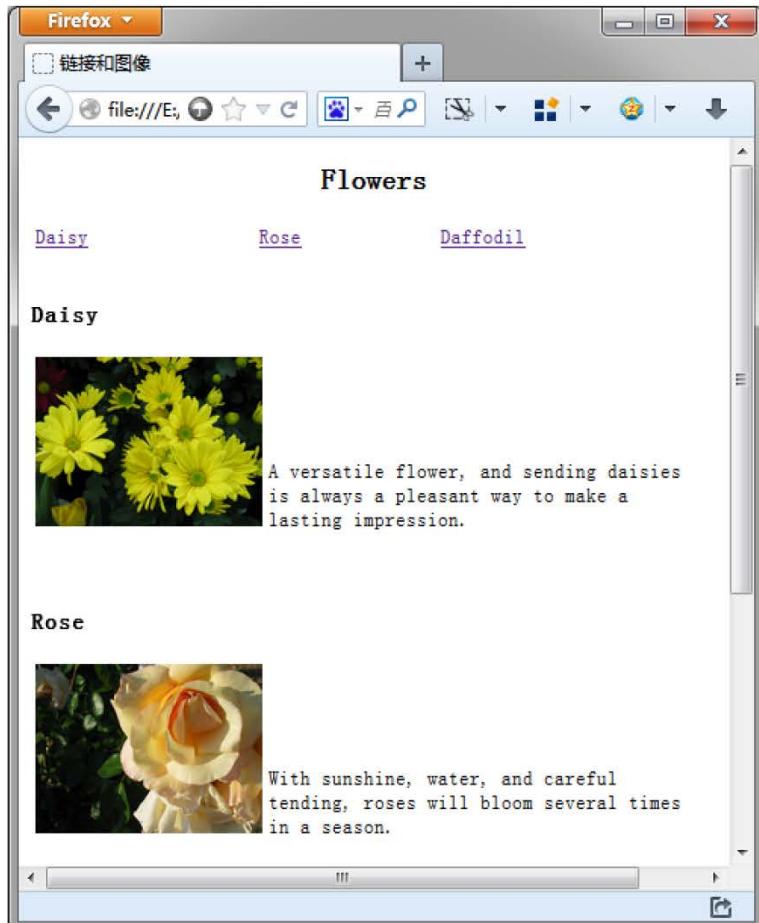


图 - 3

单击图 - 3 中页面上方的链接，可以链接到相应的位置。比如，单击图 - 3 中的链接“Daffodil”，页面将定位到页面上相应的位置，效果如图 - 4 所示：

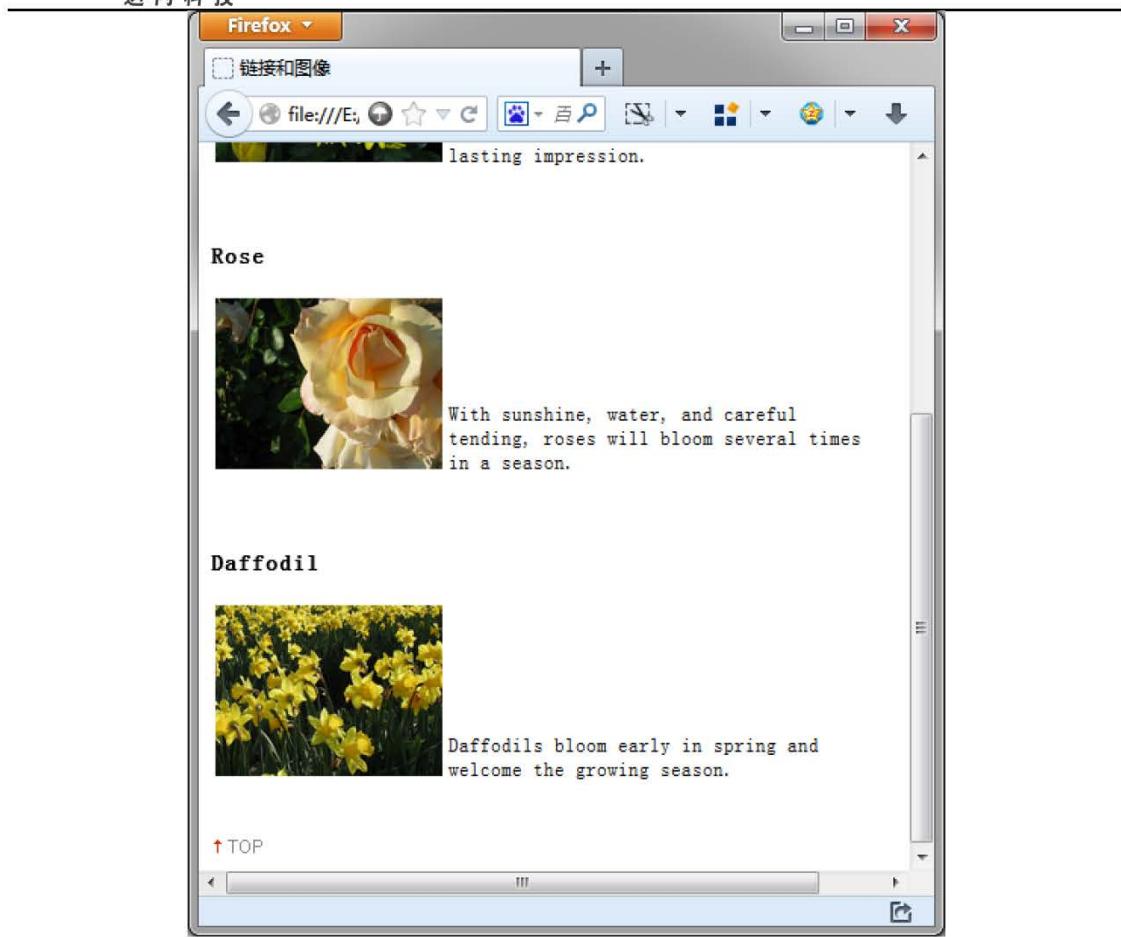


图 - 4

由图 - 4 可以看出，页面底部有图像链接（TOP），单击该图像可以返回到页面顶端。

另外，页面上的三个图片都是链接，单击后，将在新窗口（或者新页签）中打开另一个页面（介绍某种花卉的页面）。比如，单击图 - 3 中的第一个图片（Daisy，雏菊），则出现的新页面的效果如图 - 5 所示：

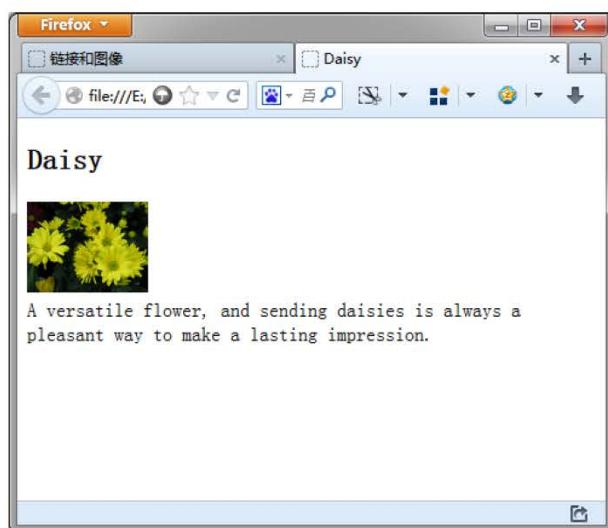


图 - 5

单击图 - 4 中的第二个图片 ( Rose , 玫瑰 ), 则出现的新页面效果如图 - 6 所示 :

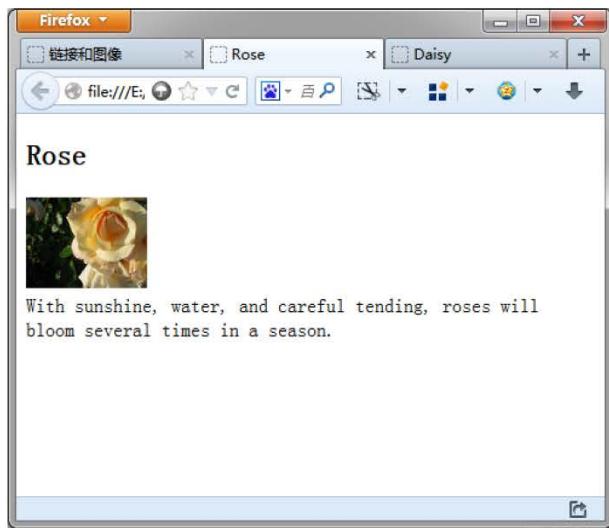


图 - 6

单击图 - 4 中的第三个图片( Daffodil , 水仙花 ), 则出现的新页面效果如图 - 7 所示 :

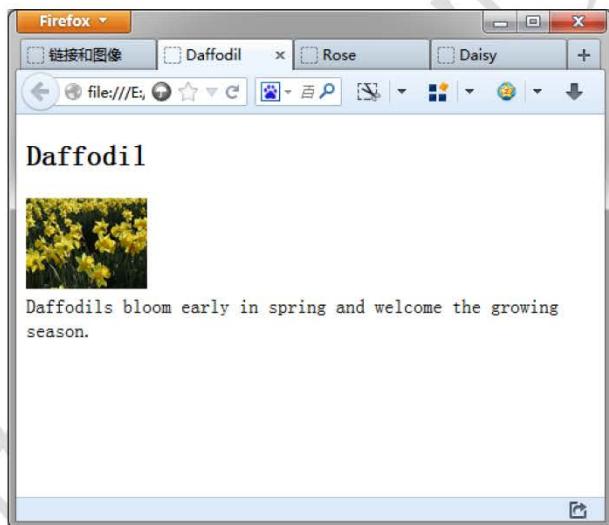


图 - 7

# HTML 和 CSS

## Unit02

知识体系.....Page 58

表单	其他控件	<label> 元素
		选项框
		<textarea> 元素
		为控件分组
CSS 概述	CSS 概述	CSS 的作用
		什么是 CSS
	使用 CSS 样式表	使用 CSS 样式表
		内联方式使用 CSS
		内部样式表
		外部样式表
CSS 语法	CSS 语法规则	CSS 语法规则总结
		CSS 样式表特征
		样式优先级
	CSS 选择器	元素选择器
		类选择器
		id 选择器
		选择器分组
		派生选择器
		伪类选择器
尺寸与框模型	尺寸与边框	样式单位
		尺寸
		边框
	框模型	框模型
		内边距 padding
		外边距 margin
背景	背景色	背景色 background-color
	背景图像	背景图片 background-image
		背景重复 background-repeat
		背景定位 background-position
		背景图片的固定

		background-attachment
--	--	-----------------------

## 经典案例.....Page 74

创建复杂表单	<label> 元素 选项框 <textarea> 元素
使用 CSS 样式	内联方式使用 CSS 内部样式表 外部样式表
测试 CSS 样式优先级	样式优先级
CSS 选择器 (1)	元素选择器 类选择器
CSS 选择器 (2)	id 选择器 选择器分组 派生选择器 伪类选择器
NetCTOSS--admin_list 页面布局	样式单位 尺寸 边框 框模型 内边距 padding 外边距 margin
NetCTOSS--admin_list 页面背景	背景色 background-color 背景图片 background-image 背景重复 background-repeat 背景定位 background-position 背景图片的固定 background-attachment

## 课后作业.....Page 106

## 1. 表单

### 1.1. 其他控件

#### 1.1.1. 【其他控件】<label> 元素

---

---

---

---

---

---

---

---

---

---

**<label> 元素**

• 语法 : <label>文本</label>

• 主要属性 :

- for : 表示与该元素相联系的控件的 ID 值

• 作用 :

- 将文本与控件联系在一起后, 单击文本, 效果就同单击控件一样

```
<input type="checkbox" name="chkHid" id="chkHid" />
<label for="chkHid">不要公开我的信息</label>
```



#### 1.1.2. 【其他控件】选项框

---

---

---

---

---

---

---

---

---

---

**选项框**

• 两种 : 下拉选项框和滚动列表

• <select> : 创建选项框

- name : 选项框命名
- size : 大于 1 , 则为滚动列表
- multiple : 设置多选

• <option> : 选项

- value : 选项的值
- selected : 预选中



---

---

---

---

---

---

---

---

---

---

**选项框 ( 续1 )**

选择课程 :

```
<select>
  <option value="1">Java</option>
  <option value="2">C++</option>
  <option value="3">PHP</option>
</select>
```

选择课程: Java

---

选择课程 : <br />

```
<select size="4">
  <option value="1">Java</option>
  <option value="2">C++</option>
  <option value="3">PHP</option>
</select>
```

选择课程:

Java  
C++  
PHP



### 1.1.3. 【其他控件】<textarea> 元素



**<textarea> 元素**

• 多行文本输入框  
– <textarea>文本</textarea>

• 主要属性：  
– cols : 指定文本区域的列数  
– rows : 指定文本区域的行数  
– readonly : 只读

**多行文本框 :**  
 line 1  
 line 2  
 |

多行文本框 : <br />
 <textarea name="txtInfo" rows="4" cols="20"></textarea>

### 1.1.4. 【其他控件】为控件分组



**为控件分组**

• <fieldset> 元素：为控件分组  
• <legend> 元素：为分组指定一个标题

<fieldset>
 <legend>地址信息</legend>
 地址 : <input type="text" /><br />
 邮编 : <input type="text" />
 </fieldset>

地址信息
   
 地址 :   
 邮编 :

## 2. CSS 概述

### 2.1. CSS 概述

#### 2.1.1. 【CSS 概述】CSS 的作用



**CSS 的作用**

• 查看如下代码

<body bgcolor="silver" text="blue">
 <font color="red">font text</font>
 <h2>h2 text</h2>
 Some text here.
 </body>

如果使用 HTML 标记中的属性来定义样式，各元素对于样式的属性定义各不相同！



知识讲解

## CSS 的作用 ( 续1 )

```
<html>
  <head>
    <style type="text/css">
      body
      {
        background-color:silver;
        color:blue;
      }
      h2
      {
        background-color:orange;
        color:red;
      }
    </style>
  </head>
```

使用 CSS 定义样式

+

知识讲解

## CSS 的作用 ( 续2 )

```
<body>
  <h2>h2 text</h2>
  Some text here.
</body>
</html>
```



使用 CSS：  
各元素使用统一的样式声明！  
且提高了样式的可重用性和可维护性！

+

### 2.1.2. 【CSS 概述】什么是 CSS

知识讲解

## 什么是 CSS

- CSS ( Cascading Style Sheets ) : 层叠样式表 , 又叫级联样式表 , 简称样式表
- 用于 HTML 文档中元素的样式定义
  - 实现了将内容与表现分离
  - 提高代码的可重用性和可维护性

+

## 2.2. 使用 CSS 样式表

### 2.2.1. 【使用 CSS 样式表】使用 CSS 样式表

+

+

使用 CSS 样式表
Tarena
达内科技

**知识讲解**

- 内联方式
  - 样式定义在单个的 HTML 元素中
- 内部样式表
  - 样式定义在 HTML 页的头元素中
- 外部样式表
  - 将样式定义在一个外部的 CSS 文件中 (.css 文件)
  - 由 HTML 页面引用样式表文件

### 2.2.2. 【使用 CSS 样式表】内联方式使用 CSS

+

+

内联方式使用 CSS
Tarena
达内科技

**知识讲解**

- 样式定义在 HTML 元素的标准属性 style 里
- CSS 语法
  - 只需要将分号隔开的一个或者多个属性/值对作为元素的 style 属性的值
  - 属性和属性值之间用 : 连接
  - 多对属性之间用 ; 隔开

```
<h1 style = "background-color: silver; color: blue;">
    文本
</h1>
```

### 2.2.3. 【使用 CSS 样式表】内部样式表

+

+

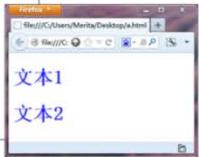
内部样式表
Tarena
达内科技

**知识讲解**

- 样式表规则位于文档头元素中的 <style> 元素内
  - 在文档的 <head> 元素内添加 <style> 元素
  - 在 <style> 元素中添加样式规则

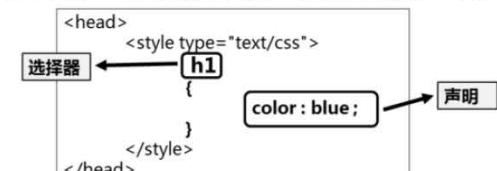
```
<html>
  <head>
    <style type="text/css">
      h1 { color: blue; }
    </style>
  </head>
  <body>
    <h1>文本1</h1>
    <h1>文本2</h1>
  </body>
</html>
```

可以在当前页面范围内重用



## 内部样式表（续1）

- 在 `<style>` 元素中添加样式规则
  - 可以定义多个样式规则
  - 每个样式规则有两个部分：选择器和样式声明
- 选择器：决定哪些元素使用这些规则
- 样式声明：一对大括号，包含一个或者多个属性/值对



### 2.2.4. 【使用 CSS 样式表】外部样式表

## 外部样式表

- 第一步：创建一个单独的样式表文件用来保存样式规则
  - 一个纯文本文件，文件后缀为.css
  - 该文件中只能包含样式规则
  - 样式规则由选择器和样式声明组成



## 外部样式表（续1）

- 第二步：在需要使用该样式表文件的页面上，使用 `<link>` 元素链接需要的外部样式表文件
  - 在文档的 `<head>` 元素内添加 `<link>` 元素



可以实现内容与表现分离  
可以被站点中的所有页面重用

### 3. CSS 语法

#### 3.1. CSS 语法规范

##### 3.1.1. 【CSS 语法规范】CSS 语法规规范总结

知识讲解

+

### CSS 语法规规范总结

• 内联样式：由样式声明组成  
<h1 style=“background-color:silver;color:blue;”>文本</h1>

• 样式表：由多个样式规则组成

- 每个样式规则有两个部分：选择器和样式声明

```
graph TD; Selector[选择器] --> H1[h1]; H1 --> Property[属性]; H1 --> Value[值]; Property --> Ellipsis[...]; subgraph Rule [h1 { text-align:center; color:red;}]
```

##### 3.1.2. 【CSS 语法规范】CSS 样式表特征

知识讲解

+

### CSS 样式表特征

- 继承性
  - 大多数 CSS 的样式规则可以被继承
- 层叠性
  - 可以定义多个样式表
  - 不冲突时，多个样式表中的样式可层叠为一个
- 优先级
  - 样式定义冲突时，按照不同样式规则的优先级来应用样式

##### 3.1.3. 【CSS 语法规范】样式优先级

知识讲解

+

### 样式优先级

- 浏览器缺省设置
- 外部样式表或者内部样式表
  - 就近优先
- 内联样式

相同的样式，如果重复定义，以最后一次的定义为准

低 → 高

## 3.2. CSS 选择器

### 3.2.1. 【CSS 选择器】元素选择器

知识讲解

#### 元素选择器

• html 文档的元素就是选择器  
– 比如 <p>、<h1> 等

```
html
{
    color : black;
}
h1
{
    color : blue ;
}
h2
{
    color : silver ;
}
```

+

### 3.2.2. 【CSS 选择器】类选择器

知识讲解

#### 类选择器

• 类选择器允许以一种独立于文档元素的方式来指定样式  
– 语法为 : .className { color:red; }

• 所有能够附带 class 属性的元素都可以使用此样式声明  
– 将元素的 class 属性的值设置为样式类名

样式表中：声明一个样式类

```
.myClass
{
    background-color: Pink;
    font-size : 35pt;
}
```

html 文档中：将元素的 class 属性的值设置为样式类的名称

```
<h2 class="myClass">h2文本</h2>
<p class="myClass">段落文本</p>
```

+

知识讲解

#### 类选择器（续1）

• 可以将类选择器和元素选择器结合起来使用，以实现对某种元素中不同样式的细分控制

• 语法为：元素选择器 .className {}  
– 先声明一个元素选择器  
– 然后使用一个点号 (.) 代表将使用类选择器  
– 然后声明一个类的名称  
– 最后使用一对大括号声明样式规则

• 将样式规则与附带 class 属性的某种元素匹配  
– 元素的 class 属性的值为分类选择器中指定的样式类名

+

### 类选择器 (续2)

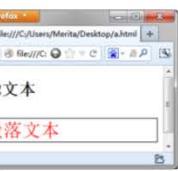
样式表中：定义一个类选择器

```
p.important
{
    color : red ;
    font-size : 20pt;
    border:1px solid black;
}
```

知识讲解

+

Tarena



h2文本

段落文本

只有 <p> 元素使用样式

html 文档中：将相应元素的 class 属性的值设置为样式类的名称

### 3.2.3. 【CSS 选择器】id 选择器

### id 选择器

- id 选择器以一种独立于文档元素的方式来指定样式
- 它仅作用于 id 属性的值
- 语法为：
  - 选择器前面需要有一个 # 号
  - 选择器本身则为文档中某个元素的 id 属性的值

知识讲解

+

Tarena

### id 选择器 (续1)

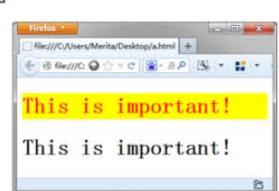
样式表中：定义一个 id 选择器

```
#mostImportant
{
    color:red;
    background:yellow;
}
```

知识讲解

+

Tarena



This is important!

This is important!

html 文档中：将元素的 id 属性的值设置为选择器的名称

<h1 id="mostImportant">This is important!</h1>  
<h1>This is important!</h1>

### 3.2.4. 【CSS 选择器】选择器分组

知识讲解

### 选择器分组

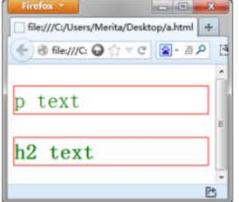
- 选择器声明为以逗号隔开的选择器列表
  - 将一些相同的规则作用于多个元素

**样式表中 : 定义选择器分组**

```
h2, p.important
{
    color:green;
    font-size:20pt;
    border:1px solid red;
}
```

**html 文档中 :**

```
<p class="important">p text</p>
<h2>h2 text</h2>
```



### 3.2.5. 【CSS 选择器】派生选择器

知识讲解

### 派生选择器

- 通过依据元素在其位置的上下文关系来定义样式
  - 选择器一端包括两个或多个用空格分隔的选择器

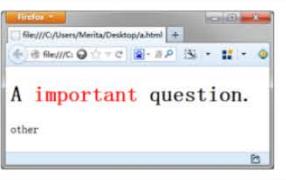
**样式表中 : 定义派生选择器**

```
h1 span
{
    color:red;
}
```

只有 <h1> 元素中的<span> 元素中的文本使用该样式

**html 文档中 :**

```
<h1>
  A <span>important</span> question.
</h1>
<span>other</span>
```



### 3.2.6. 【CSS 选择器】伪类选择器

知识讲解

### 伪类选择器

- 伪类用于向某些选择器添加特殊的效果
- 使用冒号(:)作为结合符，结合符左边是其他选择器，右边是伪类
  - 常用伪类
    - :link
    - :active
    - :visited
    - :hover
    - :focus

**a:link**

```

a:link
{
    color: black;
    font-size:15pt;
}

a:visited
{
    color: pink;
    font-size:15pt;
}

a:hover {font-size:20pt;}
a:active {color: red;}
```

## 4. 尺寸与框模型

### 4.1. 尺寸与边框

#### 4.1.1. 【尺寸与边框】样式单位

知识讲解

- % : 百分比
- in : 英寸
- cm : 厘米
- mm : 毫米
- pt : 磅(1pt等于1/72英寸)
- px : 像素(计算机屏幕上的一个点)
- em : 1em等于当前的字体尺寸，2em等于当前字体尺寸的两倍

+

知识讲解

#### 样式单位 ( 续1 )

- rgb(x,x,x) : RGB 值, 如 rgb(255,0,0)
- rgb(x%,x%,x%) : RGB百分比值, 如rgb(100%,0%,0%)
- #rrggbb : 十六进制数, 如#ff0000
- #rgb : 简写的十六进制数, 如#f00
- 表示颜色的英文单词, 如 red

+

#### 4.1.2. 【尺寸与边框】尺寸

知识讲解

#### 尺寸

- width 和 height
- overflow : 当内容溢出元素框时如何处理
  - visible
  - hidden
  - scroll
  - auto

样式表中：

```
div {  
    width:130px;  
    height:50px;  
    border:1px solid black;  
}  
div.hidden {overflow:hidden;}  
div.scroll {overflow:scroll;}  
div.auto {overflow:auto;}
```

+

## 尺寸 (续1)

html 文档中：

```
<div>
    默认为显示，即使超出尺寸，也显示全部内容
</div><br /><br />
<div class="hidden">
    文本内容超出尺寸时，隐藏文本不显示
</div><br />
<div class="scroll">
    总显示滚动条
</div><br />
<div class="auto">
    自动1：不超出不显示
</div><br />
<div class="auto">
    自动2：文本内容超出尺寸时，显示滚动条
</div><br />
```

Tarena  
达内科技



知识讲解



### 4.1.3. 【尺寸与边框】边框

## 边框

- 简写方式
  - border:width style color;
- 单边定义
  - border-left/right/top/bottom:width style color;
- border-width
  - border-left/right/top/bottom-width
- border-style
  - border-left/right/top/bottom-style
- border-color
  - border-left/right/top/bottom-color

Tarena  
达内科技

知识讲解



## 边框 (续1)

样式表中：

```
input.btn
{
    height:30px;
    width:80px;
    color:black;
    background:#f0f0f0;
    border-left:2px solid #fff;
    border-top:2px solid #fff;
    border-right:2px solid #6a6a6a;
    border-bottom:2px solid #6a6a6a;
    background:#c0c0c0;
}
```

Tarena  
达内科技



知识讲解



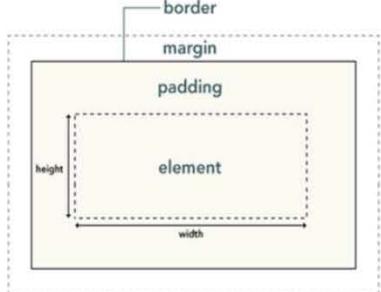
## 4.2. 框模型

### 4.2.1. 【框模型】框模型

知识讲解

### 框模型

- 框模型 ( Box Model ) 定义了元素框处理元素内容、内边距、边框和外边距的方式



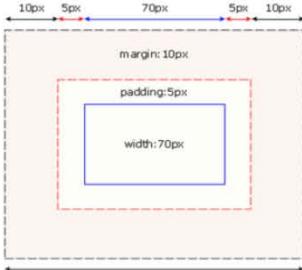
**Tarena**  
达内科技

知识讲解

### 框模型 ( 续1 )

- width 和 height 指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总尺寸

```
#box
{
    width: 70px;
    margin: 10px;
    padding: 5px;
}
```



**Tarena**  
达内科技

知识讲解

### 框模型 ( 续2 )

样式表中 :

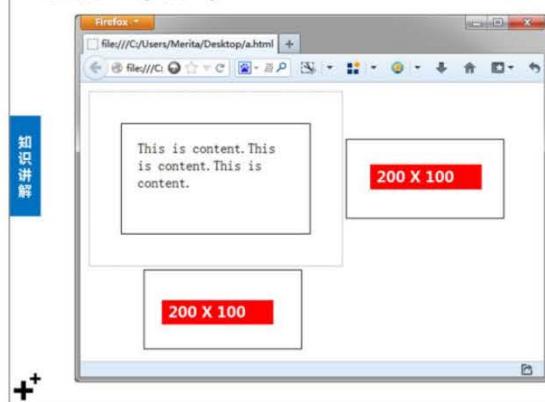
```
div.box
{
    width:200px;
    height:100px;
    padding:20px;
    margin:40px;
    border:1px solid black;
```

html 文档中 :

```
<div class="box">
    This is content.
    This is content.This is content.
</div>
```

**Tarena**  
达内科技

### 框模型 (续3)



#### 4.2.2. 【框模型】内边距 padding

### 内边距 padding

- 内边距 : 内容与框线之间的距离
  - padding : value;
- 单边设置
  - padding-top/right/bottom/left : value;
- 简写方式
  - padding : value1 value2 value3 value4;
    - top right bottom left

#### 4.2.3. 【框模型】外边距 margin

### 外边距 margin

- 外边距 : 与下一个元素之间的空间量
  - margin : value;
- 单边设置
  - margin-top/right/bottom/left : value;
- 简写方式
  - margin : value1 value2 value3 value4;
    - top right bottom left

知识讲解

外边距 margin ( 续1 )

- margin 可取值为 auto , 由浏览器计算外边距

```
div
{
    width:100px;
    height:100px;
    border:1px solid black;
    margin:0 auto;
}
```



设置居中

+

## 5. 背景

### 5.1. 背景色

#### 5.1.1. 【背景色】背景色 background-color

知识讲解

背景色 background-color

- background-color 属性用于为元素设置背景色，该属性接受任何合法的颜色值
- 为元素背景设置一种纯色

+

### 5.2. 背景图像

#### 5.2.1. 【背景图像】背景图片 background-image

知识讲解

背景图片 background-image

- 默认值是 none , 表示背景上没有放置任何图像
- 如果需要设置一个背景图像 , 需要用起始字母 url 附带一个图像的 URL 值
  - 可以是相对 URL 或者绝对 URL

```
body
{
    background-image: url("image/bg_01.gif");
}
```

+

### 5.2.2. 【背景图像】背景重复 background-repeat

知识讲解

#### 背景重复 background-repeat

• 默认情况下，背景图像在水平和垂直方向上重复出现，创建一种称为“墙纸”的效果

• 可以使用 background-repeat 属性控制背景图像的平铺效果

• background-repeat 可取值为

- repeat：在垂直方向和水平方向重复，为重复值
- repeat-x：仅在水平方向重复
- repeat-y：仅在垂直方向重复
- no-repeat：仅显示一次

+

知识讲解

#### 背景重复 background-repeat (续1)

样式表中：

```
#bar {  
    height: 76px;  
    width: 200px;  
    background-image: url(bg.jpg);  
    background-repeat: repeat-x;  
}
```

html 文档中：

```
<div id="bar"></div>
```

创建渐变色的背景效果

+

### 5.2.3. 【背景图像】背景定位 background-position

知识讲解

#### 背景定位 background-position

• background-position 属性改变图像在背景中的位置

• 该属性的取值为：

值	说明
x% y%	第一个值是水平位置，第二个值是垂直位置 表示沿着x轴（水平）和y轴（垂直）的百分比 左上角是0% 0%，右下角是100% 100%
x y	第一个值是水平位置，第二个值是垂直位置 表示沿着x轴（水平）和y轴（垂直）的绝对长度 左上角是0 0
left	在页面或者包含元素的左边显示
center	在页面或者包含元素的中间显示
right	在页面或者包含元素的右边显示
top	在页面或者包含元素的顶部显示
bottom	在页面或者包含元素的底部显示

+

知识讲解

### 背景定位 background-position (续1)

样式表中：

```
p {  
background-image: url("image/bg_02.jpg");  
background-position: center;  
background-repeat: no-repeat;  
height: 200px;  
border: 1px solid gray;  
}
```

html 文档中：

```
<p>  
这是一个具有背景图像的段落。  
</p>
```

+



#### 5.2.4. 【背景图像】背景图片的固定 background-attachment

知识讲解

### 背景图片的固定 background-attachment

• 默认情况下，背景图像会随着页面的滚动而移动  
• 可以通过 background-attachment 属性来设置  
– 默认值是 scroll：默认情况下，背景会随文档滚动  
– 可取值为 fixed：背景图像固定，并不会随着页面的其余部分滚动，常用于实现称为水印的图像

```
body  
{  
background-image: url("image/Tulips.jpg");  
background-repeat: no-repeat;  
background-attachment: fixed;  
}
```

+

## 经典案例

### 1. 创建复杂表单

- 问题

创建 01\_form.html 文档，为页面添加表单标记，实现如图 - 1 所示的页面效果：

修改个人信息

姓名: mary

性别:  女士  男士

学历: 其他

个人描述:

不要公开我的个人信息

保存 重填

图 - 1

其中，对于页面上的单选框和复选框，需要做到，单击选择框后面的文本，就象单击选择框一样。比如，单击“不要公开我的个人信息”文本，则可以实现对该文本前面的选择框的选中或者取消选中。

- 方案

首先使用表格实现图 - 1 所示的页面布局，表格布局形如图 - 2 所示：


图 - 2

图 - 2 所示的表格中，第一列用于显示文本信息，如“姓名”、“性别”等；第二列用于显示文本框、选择框等。

为实现：单击选择框后面的文本，就象单击选择框一样，需要使用 `<label>` 元素。

### • 步骤

实现此案例需要按照如下步骤进行。

#### 步骤一：创建文件

首先创建一个纯文本文件，并修改名称为 `01_form.html`；然后使用文本编辑工具，打开所创建的文件。

在此文件中，添加 HTML 代码，以创建一个标准结构的 HTML 文档（使用过渡型文档类型声明）。因为此案例中，需要使用表单控件，因此，需要在 `<body>` 元素中添加 `<form>` 元素。代码如下所示：

```
<!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>
    <title>表单</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
</head>
<body>
    <form action="" method="">
        </form>
    </body>
</html>
```

#### 步骤二：定义页面布局

在 `<form>` 元素中添加代码，创建页面的标题文本以及布局用的表格。代码如下所示：

```
<!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>
    <title>表单</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
</head>
<body>
    <form action="" method="">

        <h2>修改个人信息</h2>
        <table width="500">
            </table>

    </form>
</body>
</html>

```

### 步骤三：创建表格的第一行

为表格创建第一行的内容：修改姓名的行。定义两个单元格，并分别创建单元格中的内容。其中，文本框使用 `<input>` 元素创建，并使用 `value` 属性设置文本框中的初始值。代码如下所示：

```

<!--其他部分代码，略-->
<table width="500">

    <tr>
        <td width="70" height="35">姓名 : </td>
        <td width="180"><input type="text" value="mary" /></td>
    </tr>

</table>
<!--其他部分代码，略-->

```

### 步骤四：创建表格的第二行

为表格创建第二行的内容：修改性别选项的行。定义两个单元格，并分别创建单元格中的内容。其中，需要使用 `<label>` 元素定义文本并和单选框关联。代码如下所示：

```

<!--其他部分代码，略-->
<table width="500">
    <tr>
        <td width="70" height="35">姓名 : </td>
        <td width="180"><input type="text" value="mary" /></td>
    </tr>

    <tr>
        <td height="35">性别 : </td>
        <td width="180">
            <input type="radio" id="rexFemal" name="sex" checked />
            <label for="rexFemal">女士</label>
            <input type="radio" id="rexMal" name="sex" />
            <label for="rexMal">男士</label>
        </td>
    </tr>

```

```
</tr>

</table>
<!--其他部分代码，略-->
```

### 步骤五：创建表格的第三行

为表格创建第三行的内容：选择学历的行。其中，下拉选择框使用 `<select>` 元素创建。代码如下所示：

```
<!--其他部分代码，略-->
<table width="500">
    <tr>
        <td width="70" height="35">姓名 :</td>
        <td width="180"><input type="text" value="mary" /></td>
    </tr>
    <tr>
        <td height="35">性别 :</td>
        <td width="180">
            <input type="radio" id="rexFemal" name="sex" checked />
            <label for="rexFemal">女士</label>
            <input type="radio" id="rexMal" name="sex" />
            <label for="rexMal">男士</label>
        </td>
    </tr>

    <tr>
        <td height="35">学历 :</td>
        <td>
            <select>
                <option>其他</option>
                <option>专科</option>
                <option>本科</option>
                <option>硕士及以上</option>
            </select>
        </td>
    </tr>

```

```
</table>
<!--其他部分代码，略-->
```

### 步骤六：创建表格的第四行

为表格创建第四行的内容：录入个人描述的行。其中，多行文本框使用 `<textarea>` 元素创建，并通过 `cols` 和 `rows` 属性设置大小。代码如下所示：

```
<!--其他部分代码，略-->
<table width="500">
    <tr>
        <td width="70" height="35">姓名 :</td>
        <td width="180"><input type="text" value="mary" /></td>
    </tr>
    <tr>
```

```

<td height="35">性别 :</td>
<td width="180">
  <input type="radio" id="rexFemal" name="sex" checked />
  <label for="rexFemal">女士</label>
  <input type="radio" id="rexMal" name="sex" />
  <label for="rexMal">男士</label>
</td>
</tr>
<tr>
  <td height="35">学历 :</td>
  <td>
    <select>
      <option>其他</option>
      <option>专科</option>
      <option>本科</option>
      <option>硕士及以上</option>
    </select>
  </td>
</tr>

<tr>
  <td valign="top">个人描述 :</td>
  <td width="280">
    <textarea cols="30" rows="5"></textarea>
  </td>
</tr>

</table>
<!--其他部分代码，略-->

```

### 步骤七：创建表格的第五行

为表格创建第五行的内容。注意，需要使用 `<label>` 元素定义文本，以实现和复选框的关联。代码如下所示：

```

<!--其他部分代码，略-->





```

```

    <option>专科</option>
    <option>本科</option>
    <option>硕士及以上</option>
  </select>
</td>
</tr>
<tr>
  <td valign="top">个人描述 :</td>
  <td width="280">
    <textarea cols="30" rows="5"></textarea>
  </td>
</tr>

<tr>
  <td height="35"></td>
  <td>
    <input type="checkbox" id="chkPublic" />
    <label for="chkPublic">不要公开我的个人信息</label>
  </td>
</tr>

</table>
<!--其他部分代码，略--&gt;
</pre>

```

### 步骤八：创建表格的第六行

为表格创建第六行的内容：操作按钮的行。其中，保存按钮使用 `<input>` 元素创建，并设置 `type` 属性为 `submit`；重填按钮使用 `<input>` 元素创建，并设置 `type` 属性为 `reset`。代码如下所示：

```

<table width="500">
  <tr>
    <td width="70" height="35">姓名 :</td>
    <td width="180"><input type="text" value="mary" /></td>
  </tr>
  <tr>
    <td height="35">性别 :</td>
    <td width="180">
      <input type="radio" id="rexFemal" name="sex" checked />
      <label for="rexFemal">女士</label>
      <input type="radio" id="rexMal" name="sex" />
      <label for="rexMal">男士</label>
    </td>
  </tr>
  <tr>
    <td height="35">学历 :</td>
    <td>
      <select>
        <option>其他</option>
        <option>专科</option>
        <option>本科</option>
        <option>硕士及以上</option>
      </select>
    </td>
  </tr>

```

```

<tr>
    <td valign="top">个人描述:</td>
    <td width="280">
        <textarea cols="30" rows="5"></textarea>
    </td>
</tr>
<tr>
    <td height="35"></td>
    <td>
        <input type="checkbox" id="chkPublic" />
        <label for="chkPublic">不要公开我的个人信息</label>
    </td>
</tr>

<tr>
    <td height="35"></td>
    <td>
        <input type="submit" value="保存" />
        <input type="reset" value="重填" />
    </td>
</tr>

</table>

```

### • 完整代码

本案例的完整代码如下所示：

```

<!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>
        <title>表单</title>
        <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    </head>
    <body>
        <form action="" method="">
            <h2>修改个人信息</h2>
            <table width="500">
                <tr>
                    <td width="70" height="35">姓名:</td>
                    <td width="180"><input type="text" value="mary" /></td>
                </tr>
                <tr>
                    <td height="35">性别:</td>
                    <td width="180">
                        <input type="radio" id="rexFemal" name="sex" checked />
                        <label for="rexFemal">女士</label>
                        <input type="radio" id="rexMal" name="sex" />
                        <label for="rexMal">男士</label>
                    </td>
                </tr>
                <tr>
                    <td height="35">学历:</td>
                    <td>
                        <select>
                            <option>其他</option>
                            <option>专科</option>
                            <option>本科</option>
                            <option>硕士及以上</option>
                        </select>
                    </td>
                </tr>
            </table>
        </form>
    </body>
</html>

```

```

        </select>
    </td>
</tr>
<tr>
    <td valign="top">个人描述: </td>
    <td width="280">
        <textarea cols="30" rows="5"></textarea>
    </td>
</tr>
<tr>
    <td height="35"></td>
    <td>
        <input type="checkbox" id="chkPublic" />
        <label for="chkPublic">不要公开我的个人信息</label>
    </td>
</tr>
<tr>
    <td height="35"></td>
    <td>
        <input type="submit" value="保存" />
        <input type="reset" value="重填" />
    </td>
</tr>
</table>
</form>
</body>
</html>

```

## 2. 使用 CSS 样式

- 问题

创建一个 HTML 文档，并分别使用内联样式、内部样式表和外部样式表为页面的元素定义 CSS 样式。页面效果如图 - 3 所示。



图 - 3

图 - 3 中各元素的样式详细描述如下：

- 第一行文本为 `<p>` 元素，使用内联样式定义其样式为 25pt 的红色文本；
- 第二行文本为 `<h1>` 元素，使用内部样式表定义其样式为 20pt 的蓝色文本；

- 第三行文本为 `<h2>` 元素，使用外部样式表定义其样式为 15pt 的绿色文本，且背景色为银灰色（ silver ）。

- 方案**

使用 CSS 样式的方式有三种：

- 内联方式：样式定义在单个的 HTML 元素的标准属性 `style` 里；
- 内部样式表：样式定义在 HTML 页的头元素中；
- 外部样式表：将样式定义在一个外部的 CSS 文件中( `.css` 文件 )，并在 HTML 页面引用样式表文件。

- 步骤**

实现此案例需要按照如下步骤进行。

**步骤一：创建页面文件**

首先创建一个纯文本文件，并修改名称为 `02_demo_v01.html`，然后使用文本编辑工具，打开所创建的文件。

在此文件中，添加 HTML 代码，以创建一个标准结构的 HTML 文档（使用过渡型文档类型声明）。代码如下所示：

代码如下所示：

```
<!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>
    <title>使用 CSS 样式</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
</head>
<body>
</body>
</html>
```

**步骤二：使用内联方式为 `<p>` 元素定义样式**

在`<body>`元素中添加 `<p>` 元素，并使用 `style` 属性为其定义内联样式。代码如下所示：

```
<!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>
    <title>使用 CSS 样式</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
</head>
<body>
<!--内联样式-->
```

```
<p style="color:Red;font-size:25pt;">段落文本 ( p ) : 使用内联样式</p>
```

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

### 步骤三：使用内部样式表为 <h1> 元素定义样式

为了使用内部样式表，首先需要在页面的 <head> 元素里添加 <style> 元素，并为 <h1> 元素定义 CSS 样式，以设置其字体颜色和字体大小。

样式定义完毕后，在<body>元素中定义<h1>元素。代码如下所示：

```
<!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>
    <title>使用 CSS 样式</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />

    <style type="text/css">
        h1 {color:Blue;font-size:20pt;}
    </style>

</head>
<body>
    <!--内联样式-->
    <p style="color:Red;font-size:25pt;">段落文本 ( p ) : 使用内联样式</p>

    <!--内部样式表-->
    <h1>一级标题 ( h1 ) : 使用内部样式表</h1>

</body>
</html>
```

### 步骤四：创建外部样式表为 <h2> 元素定义样式

在当前目录下创建一个名为 demo.css 的文件。

注意此文件的存放路径。此案例中，demo.css 文件和 02\_demo\_v01.html 文件存放 在同一个目录下。

### 步骤五：在外部样式表中为 <h2> 元素定义样式

使用文本编辑器打开 demo.css 文件，并在此样式文件中添加 CSS 样式，以定义 <h2> 元素的字体颜色、字体大小和背景色。demo.css 文件中的代码如下所示：

```
h2
{
    color:Green;
    font-size:15pt;
    background-color:Silver;
}
```

### 步骤六：为 HTML 页面引入外部样式表

在 02\_demo\_v01.html 页面的 <head> 元素中使用 <link> 元素引入外部样式表，  
并在<body>元素中定义<h2>元素。代码如下所示：

```
<!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>
    <title>使用 CSS 样式</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <style type="text/css">
        h1 {color:Blue;font-size:20pt;}
    </style>

    <link type="text/css" rel="stylesheet" href="demo.css" />

</head>
<body>
    <!--内联样式-->
    <p style="color:Red;font-size:25pt;">段落文本 ( p ) : 使用内联样式</p>
    <!--内部样式表-->
    <h1>一级标题 ( h1 ) : 使用内部样式表</h1>

    <!--外部样式表-->
    <h2>二级标题 ( h2 ) : 使用外部样式表</h2>

</body>
</html>
```

- **完整代码**

02\_demo\_v01.html 文件中的代码如下所示：

```
<!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>
    <title>使用 CSS 样式</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <link type="text/css" rel="stylesheet" href="demo.css" />
    <style>
        h1 {color:Blue;font-size:20pt;}
    </style>
</head>
<body>
    <!--内联样式-->
    <p style="color:Red;font-size:25pt;">段落文本 ( p ) : 使用内联样式</p>
    <!--内部样式表-->
    <h1>一级标题 ( h1 ) : 使用内部样式表</h1>
    <!--外部样式表-->
    <h2>二级标题 ( h2 ) : 使用外部样式表</h2>
</body>
</html>
```

demo.css 文件中的代码如下所示：

```
h2
{
    color:Green;
    font-size:15pt;
    background-color:Silver;
}
```

### 3. 测试 CSS 样式优先级

- 问题

比较内联样式、内部样式表和外部样式表的优先级。

- 步骤

实现此案例需要按照如下步骤进行。

#### 步骤一：创建页面文件

首先创建一个纯文本文件，并修改名称为 `03_demo_v02.html`，然后使用文本编辑工具，打开所创建的文件。

在此文件中，添加 HTML 代码，以创建一个标准结构的 HTML 文档（使用过渡型文档类型声明）。代码如下所示：

```
<!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>
        <title>CSS 样式优先级</title>
        <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    </head>
    <body>
    </body>
</html>
```

#### 步骤二：添加 `<h3>` 元素，并为其定义内联样式

在文档中添加 `<h3>` 元素，并使用内联样式为其定义字体颜色为灰色，代码如下所示：

```
<!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>
        <title>CSS 样式优先级</title>
        <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    </head>
    <body>
        <!--样式优先级-->

        <h3 style="color:Gray;">h3 text : 使用何种样式 ? </h3>

    </body>
</html>
```

### 步骤三：在内部样式表中为元素定义样式

在页面的<head>元素中添加<style>元素，为页面定义内部样式表。并在内部样式表中继续为 <h3> 元素定义样式：字体颜色为红色，且字体大小为 30pt，代码如下所示：

```
<!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>
    <title>CSS 样式优先级</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />

    <style type="text/css">
        h3 {color:Red;font-size:30pt;}
    </style>

</head>
<body>
    <!--样式优先级-->
    <h3 style="color:Gray;">h3 text：使用何种样式？</h3>
</body>
</html>
```

此时，在浏览器中查看页面，页面显示效果如图 - 4 所示：



图 - 4

由图 - 4 可以看出，<h3>中的文本显示为灰色，字体大小为 30pt。因此，可以得出第一个结论：样式没有重复定义时，取并集（比如字体大小）；样式重复定义时，内联样式优先于内部样式表（比如字体颜色）。

### 步骤四：在外部样式表中为元素定义样式

在外部样式表 demo.css 中再为 <h3> 元素定义样式：字体颜色为黄色，字体大小为 10pt，且背景色为银灰色。demo.css 文件中的代码如下所示：

```
h3
{
    color:Yellow;
    font-size:10pt;
    background-color:Silver;
}
```

## 步骤五：先引入外部样式表，再定义内部样式表

在 `html` 文档中，先引入外部样式表，再定义内部样式表。HTML 页面的代码如下所示：

```
<!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>
    <title>CSS 样式优先级</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />

    <link type="text/css" rel="stylesheet" href="demo.css" />

    <style type="text/css">
        h3 {color:Red;font-size:30pt;}
    </style>
</head>
<body>
    <!--样式优先级-->
    <h3 style="color:Gray;">h3 text：使用何种样式？</h3>
</body>
</html>
```

至此，使用了三种方式为 `<h3>` 元素定义 CSS 样式。`<h3>` 元素中的文本的最终显示效果如图 - 5 所示。



图 - 5

由图 - 5 可以看出，`<h3>` 元素中的文本的字体颜色为灰色，即内联样式的优先级最高；而 `<h3>` 元素中的文本的字体大小为 `30pt`，似乎内部样式的优先级高于外部样式表。

## 步骤六：先定义内部样式表，再引入外部样式表

为了最终确定外部样式表和内部样式表的优先级，我们将代码略作修改：在 `html` 文档中，先定义内部样式表，再引入外部样式表，代码如下所示：

```
<!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>
    <title>CSS 样式优先级</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

```

<style type="text/css">
    h3 {color:Red;font-size:30pt;}
</style>
<link type="text/css" rel="stylesheet" href="demo.css" />

</head>
<body>
    <!--样式优先级-->
    <h3 style="color:Gray;">h3 text : 使用何种样式 ? </h3>
</body>
</html>

```

此时，页面的显示效果如图 - 6 所示。



图 - 6

由图 - 6 可以看出，`<h3>` 元素中文本的字体大小为 `10pt`，即外部样式表的优先级高于内部样式表。

### 步骤七：结论

最后，`<h3>` 元素中的文本的背景色为银灰色，表示各种样式表中的样式会统一应用到元素中，而重复定义的样式则以优先级为准。

因此，对于 CSS 样式优先级，有如下结论：

- 1、可以为页面定义多重样式；
- 2、多重样式定义不冲突时，多个样式表中的样式可层叠为一个；
- 3、当多重样式定义冲突时，按照不同样式规则的优先级来应用样式；
- 4、内联样式的优先级最高，而内部样式表和外部样式表的优先级取决于样式定义的先后位置，后定义的样式会覆盖先定义的样式，即就近优先。

- **完整代码**

03\_demo\_v02.html 文件中的代码如下所示：

```

<!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>
        <title>CSS 样式优先级</title>
        <meta http-equiv="content-type" content="text/html; charset=utf-8" />
        <style type="text/css">

```

```

        h3 {color:Red;font-size:30pt; }
    </style>
    <link type="text/css" rel="stylesheet" href="demo.css" />
</head>
<body>
    <!--样式优先级-->
    <h3 style="color:Gray;">h3 text: 使用何种样式? </h3>
</body>
</html>

```

demo.css 文件中的代码如下所示：

```

h3
{
    color:Yellow;
    background-color:Silver;
    font-size:10pt;
}

```

## 4. CSS 选择器 (1)

- 问题

使用元素选择器为 `<h4>` 元素定义样式；使用类选择器为 `<p>` 和 `<div>` 元素定义相同的样式；并结合元素和类选择器为按钮和文本框定义不同的样式。页面效果如图 - 7 所示。



图 - 7

图 - 7 中各元素及其样式的详细描述如下：

- 第一行文本为 `<h4>` 元素，使用元素选择器将其定义为 20pt 的红色文本；
- 第二行文本为 `<div>` 元素，第三行文本为 `<p>` 元素，使用类选择器将这两行文本均定义为绿色文本，且有宽度为 1px 的灰色实线边框；
- 第四行的按钮为 `<input>` 元素，使用分类选择器定义该按钮的宽度为 50px，且具有宽度为 1px 的灰色实线边框；
- 第四行的文本框为 `<input>` 元素，使用分类选择器定义该文本框的宽度为 150px，且具有宽度为 1px 的黑色实线边框。

## • 步骤

实现此案例需要按照如下步骤进行。

### 步骤一：创建页面文件

首先创建一个纯文本文件，并修改名称为 `04_demo_v03.html`，然后使用文本编辑工具，打开所创建的文件。

在此文件中，添加 HTML 代码，以创建一个标准结构的 HTML 文档（使用过渡型文档类型声明），并为页面定义内部样式表。代码如下所示：

```
<!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>
    <title>CSS 选择器 (1)</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <style type="text/css">
        </style>
</head>
<body>
</body>
</html>
```

### 步骤二：使用元素选择器

首先在内部样式表中，使用元素选择器为 `<h4>` 元素定义样式（`20pt` 的红色文本）；然后，在`<body>`元素中添加`<h4>`元素。代码如下所示：

```
<!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>
    <title>CSS 选择器 (1)</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <style type="text/css">

        h4 {color:Red;font-size:20pt;}

    </style>
</head>
<body>

    <!--选择器 1-->
    <h4>h4 : 元素选择器</h4>

</body>
</html>
```

### 步骤三：使用类选择器

继续在内部样式表中，定义名为 `style` 的样式类（绿色文本，且有宽度为 `1px` 的灰色实线边框），并在 `<body>` 元素中添加 `<div>` 和 `<p>` 元素，并使用样式 `style`。代码

如下所示：

```
<!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>
    <title>CSS 选择器 (1)</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <style type="text/css">
        h4 {color:Red;font-size:20pt;}
        .style {border:1px solid gray;color:Green;}
    </style>
</head>
<body>
    <!--选择器 1-->
    <h4>h4 : 元素选择器</h4>

    <div class="style">div : 类选择器</div>
    <p class="style">p : 类选择器</p>

</body>
</html>
```

#### 步骤四：结合元素使用类选择器

在内部样式表中，结合元素使用类选择器，首先为按钮定义样式（宽度为 50px，且具有宽度为 1px 的灰色实线边框），然后为文本框定义样式（宽度为 150px，且具有宽度为 1px 的黑色实线边框），最后在 `<body>` 元素中添加两个 `<input>` 元素，并分别使用所定义的样式。代码如下所示：

```
<!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>
    <title>CSS 选择器 (1)</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <style type="text/css">
        h4 {color:Red;font-size:20pt;}
        .style {border:1px solid gray;color:Green;}

        input.btn {border:1px solid gray;width:50px;}
        input.txt {border:1px solid black;width:150px;}
    </style>
</head>
<body>
    <!--选择器 1-->
    <h4>h4 : 元素选择器</h4>
    <div class="style">div : 类选择器</div>
    <p class="style">p : 类选择器</p>
```

```
<input type="button" class="btn" value="按钮" />
<input type="text" class="txt" />

</body>
</html>
```

- **完整代码**

本案例的完整代码如下所示：

```
<!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>
    <title>CSS 选择器 (1) </title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <style>
        h4 {color:Red;font-size:20pt;}
        .style {border:1px solid gray;color:Green;}
        input.btn {border:1px solid gray;width:50px;}
        input.txt {border:1px solid black;width:150px;}
    </style>
</head>
<body>
    <!--选择器 1-->
    <h4>h4: 元素选择器</h4>
    <div class="style">div: 类选择器</div>
    <p class="style">p: 类选择器</p>
    <input type="button" class="btn" value="按钮" />
    <input type="text" class="txt" />
</body>
</html>
```

## 5. CSS 选择器 (2)

- **问题**

使用 `id` 选择器为 `<p>` 元素定义样式；使用选择器分组为 `<label>` 和 `<span>` 元素统一定义样式；使用派生选择器定义文本“一周排行榜”中“榜”字的样式；并使用伪类选择器为超级链接文本定义样式，使得链接文本为 `12pt` 的灰色文本，页面效果如图 - 8 所示。



图 - 8

图 - 8 中的元素及其样式的详细描述如下：

- 第一行文本为 `<p>` 元素，使用 id 选择器将其定义为 30pt 的文本，且背景色为银灰色；
- 第二行文本分别为 `<label>` 元素和 `<span>` 元素，使用选择器分组的方式为二者定义相同的样式：文本的大小为 15 pt，且有宽度为 1px 的黑色实线边框；
- 第三行文本为 `<p>` 元素，其中的“榜”字为 `<span>` 元素，使用派生选择器定义“榜”字为 15pt 的红色文本，并有宽度为 1px 的红色实线边框；
- 第四行文本为 `<a>` 元素，链接文本为 12pt 的灰色文本。

当鼠标移入链接文本“使用伪类选择器”时，该文本变为 15pt 的红色文本，页面效果如图 - 9 所示。



图 - 9

• 步骤

实现此案例需要按照如下步骤进行。

### 步骤一：创建页面文件

首先创建一个纯文本文件，并修改名称为 `05_demo_v04.html`，然后使用文本编辑工具，打开所创建的文件。

在此文件中，添加 HTML 代码，以创建一个标准结构的 HTML 文档（使用过渡型文档类型声明），并为页面定义内部样式表。代码如下所示：

```
<!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>
<title>CSS 选择器 (2)</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<style type="text/css">
</style>
</head>
<body>
</body>
</html>
```

### 步骤二：使用 id 选择器

在内部样式表中，使用 `id` 选择器为 `<p>` 元素定义样式（`30pt` 的文本，且背景色为银灰色）；并在 `<body>` 元素中添加 `<p>` 元素，并设置其 `id` 属性的值。代码如下所示：

```
<!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>
<title>CSS 选择器 (2)</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<style type="text/css">

#title {font-size:30pt;background-color:Silver; }

</style>
<link type="text/css" rel="stylesheet" href="demo.css" />
</head>
<body>

<!--选择器 2--&gt;
&lt;p id="title"&gt;id 选择器&lt;/p&gt;

&lt;/body&gt;
&lt;/html&gt;</pre>
```

### 步骤三：选择器分组

使用选择器分组的方式为 `<span>` 元素和 `<label>` 元素定义相同的样式（文本的大小为 `15 pt`，且有宽度为 `1px` 的黑色实线边框），并在 `<body>` 元素中添加 `<span>` 和 `<label>` 元素。代码如下所示：

```

<!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>
    <title>CSS 选择器 (2)</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <style type="text/css">
        #title {font-size:30pt;background-color:Silver; }

        label,span {font-size:15pt;border:1px solid black;}

    </style>
    <link type="text/css" rel="stylesheet" href="demo.css" />
</head>
<body>
    <!--选择器2-->
    <p id="title">id 选择器</p>

    <label>label 文本</label>
    <span>span 文本</span>

</body>
</html>

```

#### 步骤四：使用派生选择器

使用派生选择器为 `<p>` 元素中的 `<span>` 元素定义样式 (15pt 的红色文本，并有宽度为 1px 的红色实线边框)，并在 `<body>` 元素中添加相应的标记。代码如下所示：

```

<!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>
    <title>CSS 选择器 (2)</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <style type="text/css">
        #title {font-size:30pt;background-color:Silver; }
        label,span {font-size:15pt;border:1px solid black; }

        p span
        {
            font-size:15pt;
            color:Red;
            border:1px solid red;
        }

    </style>
    <link type="text/css" rel="stylesheet" href="demo.css" />
</head>
<body>
    <!--选择器2-->
    <p id="title">id 选择器</p>
    <label>label 文本</label>
    <span>span 文本</span>


```

```
<p>一周排行<span>榜</span></p>
```

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

## 步骤五：使用伪类选择器

最后使用伪类选择器为超级链接定义其不同状态下的样式：链接文本为 12pt 的灰色文本，鼠标移入链接文本时，文本变为 15pt 的红色文本；并在 `<body>` 元素中添加超级链接元素。代码如下所示：

```
<!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>
    <title>CSS 选择器 (2)</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <style type="text/css">
        #title {font-size:30pt;background-color:Silver;}
        label,span {font-size:15pt;border:1px solid black;}
        p span {
            font-size:15pt;
            color:Red;
            border:1px solid red;
        }

        a {
            color:Gray;
            font-size:12pt;
        }
        a:hover {
            color:Red;
            font-size:15pt;
        }
    </style>
    <link type="text/css" rel="stylesheet" href="demo.css" />
</head>
<body>
    <!--选择器2-->
    <p id="title">id 选择器</p>
    <label>label 文本</label>
    <span>span 文本</span>
    <p>一周排行<span>榜</span></p>

    <a href="#">使用伪类选择器</a>

    <br /><br /><br /><br />
</body>
</html>
```

- 完整代码

本案例的完整代码如下所示：

```
<!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>
    <title>CSS 选择器 (2) </title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <style type="text/css">
        #title {font-size:30pt;background-color:Silver;}
        label,span {font-size:15pt;border:1px solid black;}
        p span {
            font-size:15pt;
            color:Red;
            border:1px solid red;
        }
        a {
            color:Gray;
            font-size:12pt;
        }
        a:hover {
            color:Red;
            font-size:15pt;
        }
    </style>
    <link type="text/css" rel="stylesheet" href="demo.css" />
</head>
<body>
    <!--选择器 2-->
    <p id="title">id 选择器</p>
    <label>label 文本</label>
    <span>span 文本</span>
    <p>一周排行<span>榜</span></p>
    <a href="#">使用伪类选择器</a>
    <br /><br /><br /><br />
</body>
</html>
```

## 6. NetCTOSS--admin\_list 页面布局

### • 问题

创建管理员列表页面 ( admin\_list.html ) , 并使用 CSS 样式为其定义页面布局 , 页  
面效果如图 - 10 所示。

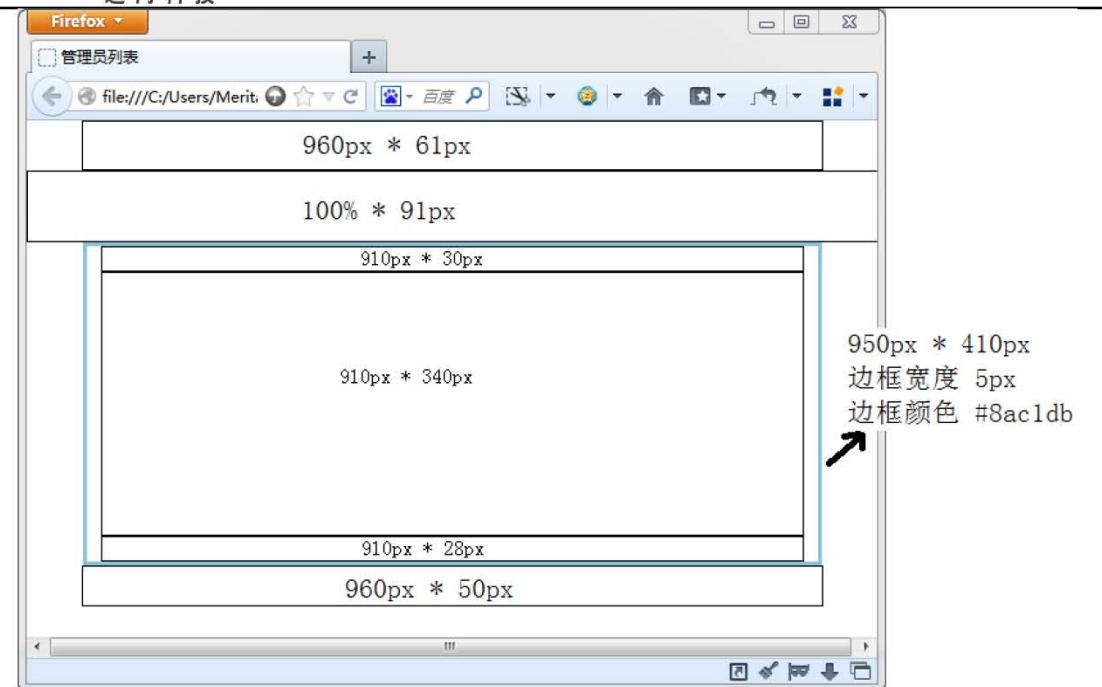


图 - 10

图 - 10 中已经标识出各组成部分的尺寸、颜色等信息。

### • 方案

分析图 - 10 所示的页面，可以发现，实现此案例时，首先需要在 HTML 代码中，使用 `<div>` 元素定义页面的各组成部分，然后在样式表中，分别为各组成部分定义样式。

此页面主要由四个部分组成：最上方的页头部分，随后的导航部分，中间的页面主要内容区域，以及页面最下方的页脚区域。其中，中间的主要内容区域，又分为三个组成部分。

### • 步骤

实现此案例需要按照如下步骤进行。

#### 步骤一：创建 HTML 文档，准备 CSS 样式文件

创建文件 `admin_list.html`，并为其准备一个名为 `admin_style.css` 的样式文件，HTML 页面的代码如下所示：

```

<!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>
    <title>管理员列表</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <link type="text/css" rel="stylesheet" href="admin_style.css" />
</head>
<body>
</body>
</html>

```

## 步骤二：定义页面的各组成部分

在 HTML 文件的 <body> 元素中添加 <div> 元素，用来定义页面的各组成部分。

每个 <div> 元素表示页面的一个独一无二的组成部分，因此，为各 <div> 元素定义 id 属性，便于后续的样式定义。

HTML 页面的代码如下所示：

```
<!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>
    <title>管理员列表</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <link type="text/css" rel="stylesheet" href="admin_style.css" />
</head>
<body>

    <!--Logo 区域-->
    <div id="header">
    </div>
    <!--导航区域-->
    <div id="navi">
    </div>
    <!--主要区域-->
    <div id="main">
        <!--操作区域-->
        <div id="operate">
        </div>
        <!--数据区域-->
        <div id="data">
        </div>
        <!--分页-->
        <div id="pages">
        </div>
    </div>
    <!--页脚区域-->
    <div id="footer">
    </div>

</body>
</html>
```

## 步骤三：为页面定义全局样式

在文件 admin\_style.css 中添加 CSS 代码。

首先为页面定义全局的统一样式。

- <body> 元素：取消内外的上下边距和边框，并设置左右外边距为 auto 以便于居中显示；
- <div> 元素：黑色边框、内边距为 0 以及页面居中显示。

admin\_style.css 文件中的代码如下所示：

```
/* 全局样式 */
body
{
    margin: 0px auto;
    padding: 0px;
    border: 0;
}
div
{
    border:1px solid black;
    margin: 0px auto;
    padding: 0px;
}
```

#### 步骤四：为页面定义整体布局样式

继续为 admin\_style.css 文件添加代码，定义其他部分的样式。

因为页头和页脚部分的宽度相同，因此，使用选择器分组的方式为其定义宽度，css 代码如下所示：

```
#header, #footer
{
    width: 960px;
}
```

然后为页头部分定义其高度，css 代码如下所示：

```
#header
{
    height: 61px;
}
```

分别为导航栏部分、页面主体部分以及页脚部分定义高度和宽度，css 代码如下所示：

```
#navi
{
    width: 100%;
    height: 91px;
}
#main
{
    border:5px solid #8ac1db;
    height: 410px;
    width: 950px;
}
#footer
{
    height: 50px;
}
```

#### 步骤五：为 id 为 main 的 <div> 元素里的各组成部分定义样式

继续在 admin\_style.css 文件中添加代码，为 id 为 main 的 <div> 元素里的各组成部分统一定义宽度，css 代码如下所示：

```
#operate, #data, #pages
{
```

```
width: 910px;
}
```

分别为操作部分、数据表格区域、分页的页码部分定义高度，css 代码如下所示：

```
#operate
{
    height: 30px;
}
#data
{
    height: 340px;
}
#pages
{
    height: 28px;
}
```

### • 完整代码

`admin_list.html` 的文件的代码如下所示：

```
<!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>
    <title>管理员列表</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <link type="text/css" rel="stylesheet" href="admin_style.css" />
</head>
<body>
    <!--Logo 区域-->
    <div id="header">
    </div>
    <!--导航区域-->
    <div id="navi">
    </div>
    <!--主要区域-->
    <div id="main">
        <!--操作区域-->
        <div id="operate">
        </div>
        <!--数据区域-->
        <div id="data">
        </div>
        <!--分页-->
        <div id="pages">
        </div>
    </div>
    <!--页脚区域-->
    <div id="footer">
    </div>
    </body>
</html>
```

`admin_style.css` 文件的代码如下所示：

```
/* 全局样式 */
body
{
```

```
margin: 0px auto;
padding: 0px;
border: 0;
}
div
{
    border:1px solid black;
    margin: 0px auto;
    padding: 0px;
}
/* 整体布局样式 */
#header, #footer
{
    width: 960px;
}
#header
{
    height: 61px;
}
#navi
{
    width: 100%;
    height: 91px;
}
#main
{
    border:5px solid #8ac1db;
    height: 410px;
    width: 950px;
}
#footer
{
    height: 50px;
}
/*正文部分*/
#operate,#data,#pages
{
    width:910px;
}
#operate
{
    height: 30px;
}
#data
{
    height: 340px;
}
#pages
{
    height: 28px;
}
```

## 7. NetCTOSS--admin\_list 页面背景

- 问题

继续修改管理员列表页面( admin\_list.html ) ,为页面的各组成部分定义背景样式 ,  
页面效果如图 - 11 所示。

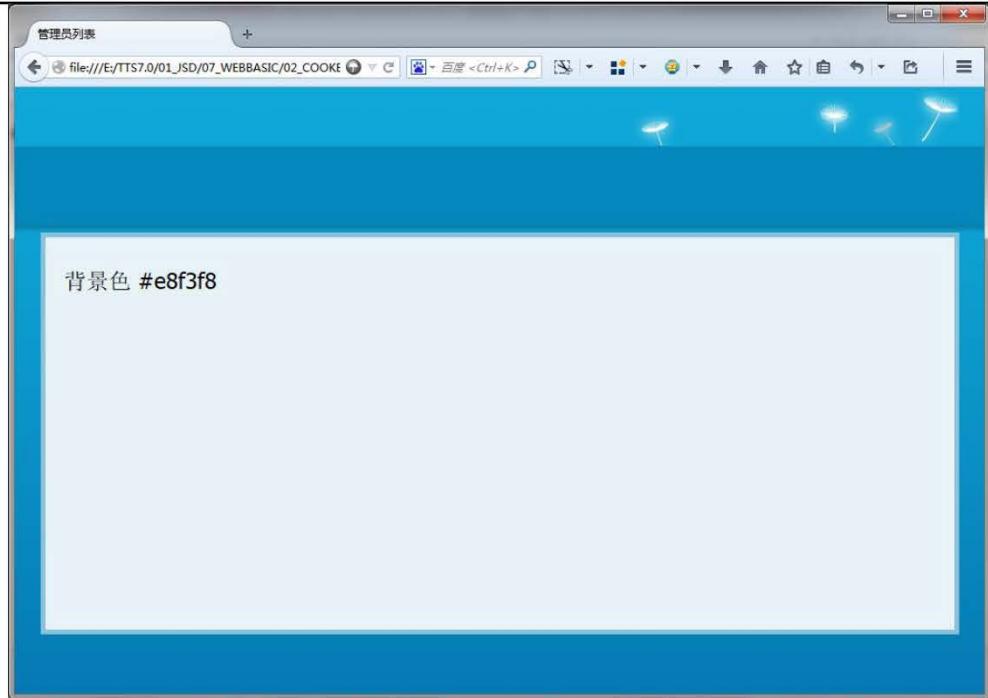


图 - 11

- 方案

分析图 - 11 所示的页面，可以发现，实现此案例时，只需要为各组成部分定义背景样式即可。其中：

- <body> 元素使用了渐变色的背景图片，需要实现背景图片的平铺；
- 页头部分使用了特定的背景图片，不需要平铺；
- 导航部分使用了渐变色的背景图片，需要实现背景图片的平铺；
- 正文部分使用了背景色#e8f3f8。

- 步骤

实现此案例需要按照如下步骤进行。

**步骤一：为 <body> 元素定义背景**

修改 admin\_style.css 文件中的代码。

首先为 <body> 元素定义背景等样式，并去掉 <div> 元素的边框样式。修改后的 CSS 代码如下所示：

```
body
{
    margin: 0px auto;
    padding: 0px;
    border: 0;

    background-repeat: repeat-x;
    background-image: url(images/body_bg.png);
```

```
}
```

```
div
{
    margin: 0px auto;
    padding: 0px;
}
```

## 步骤二：为页头部分定义背景

修改页头部分的样式定义，为其添加背景。修改后的 CSS 代码如下所示：

```
#header
{
    height: 61px;

    background: url(images/top_bg.png) no-repeat;

}
```

## 步骤三：为导航栏部分定义背景

修改导航栏部分的样式，为其定义背景。修改后的 CSS 代码如下所示：

```
#navi
{
    width: 100%;
    height: 91px;

    background: url(images/navigation.png) repeat-x;

}
```

## 步骤四：为页面主体部分定义背景

修改页面主体部分的样式，为其定义背景。修改后的 CSS 代码如下所示：

```
#main
{
    border: 5px solid #8ac1db;
    height: 410px;
    width: 950px;

    background-color: #e8f3f8;

}
```

### • 完整代码

本案例中，admin\_list.html 文件没有变化。

admin\_style.css 文件中的代码如下所示：

```
/* 全局样式 */
body
{
```

```
margin: 0px auto;
padding: 0px;
border: 0;
background-repeat: repeat-x;
background-image: url(images/body_bg.png);
}
div
{
    margin: 0px auto;
    padding: 0px;
}
/* 整体布局样式 */
#header, #footer
{
    width: 960px;
}
#header
{
    height: 61px;
    background: url(images/top_bg.png) no-repeat;
}
#navi
{
    width: 100%;
    height: 91px;
    background: url(images/navigation.png) repeat-x;
}
#main
{
    border:5px solid #8ac1db;
    height: 410px;
    width: 950px;
    background-color: #e8f3f8;
}
#footer
{
    height: 50px;
}

/*正文部分*/
#operate, #data, #pages
{
    width:910px;
}
#operate
{
    height: 30px;
}
#data
{
    height: 340px;
}
#pages
{
    height: 28px;
}
```

## 课后作业

### 1. 创建如下图所示的页面

---->>求职申请<<-----

用户名:

密码:

籍贯:

性别:  男  女  不透露

头像:  未选择文件。

愿意到以下城市工作:

北京  
上海  
南京  
深圳

留言:

不要公开我的信息

图 - 1

### 2. 简述 CSS 样式表的使用方式

### 3. 如何理解 CSS 样式表的层叠性

### 4. CSS 选择器中，元素选择器和类选择器的区别是什么？

### 5. 完成如下图所示的页面

在课堂案例的基础上，为 admin\_list.html 页面添加代码，完成图-2 中黑色圈出部分。

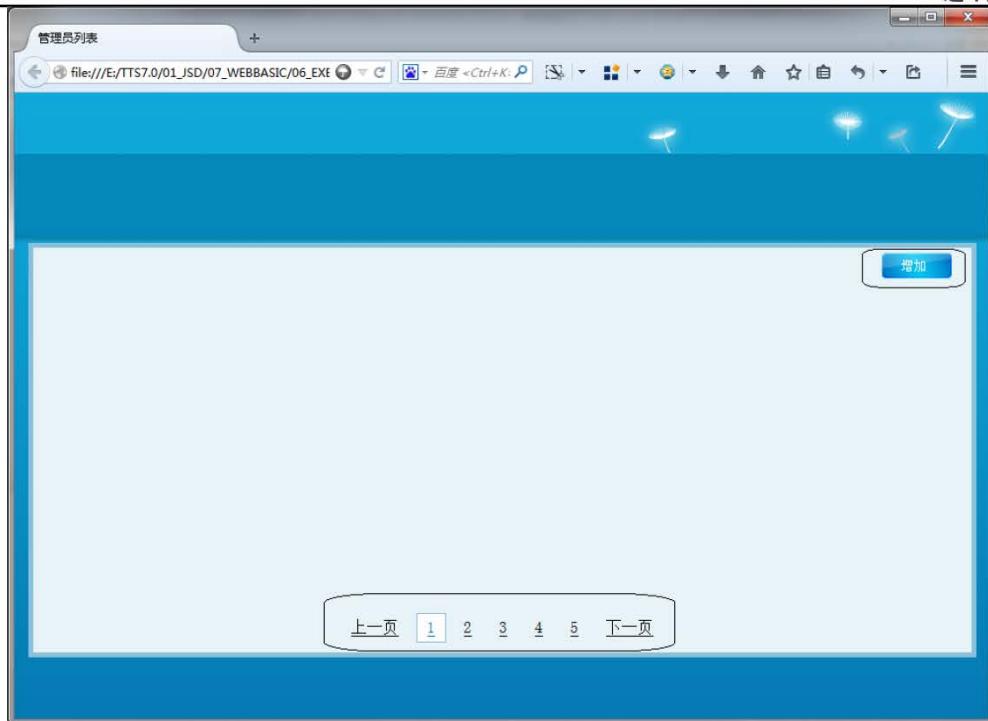


图 - 2

其中，第一个黑色圈出部分为操作部分的“增加”按钮，想想效果如图 - 3 所示。



图 - 3

第二个黑色圈出部分为页码部分。当鼠标移入页码时，页面效果如图 - 4 所示。



图 - 4

图 - 4 中，当前页码（比如页码 1）的背景色为白色，边框颜色为#89bdd8，文本颜色为#067db5；鼠标移入某个页码（比如页码 3）后，背景色为#cddde4，边框颜色为#97b9c9，文本颜色为#067db5。

# HTML 和 CSS

## Unit03

知识体系.....Page 110

文本格式化	文本格式化	控制字体
		控制文本格式
表格样式	表格常用样式属性	表格常用样式属性
		垂直方向对齐
	表格特有样式属性	边框合并
		边框边距
浮动定位	定位概述	定位概述
		普通流定位
	浮动定位	浮动概述
		浮动定位
		float 属性
		clear 属性
显示	显示方式	显示方式
		display 属性
	光标	光标
		cursor 属性
列表样式	列表样式	列表项标志 list-style-type
		列表项图像 list-style-image
定位	定位机制	什么是定位
		定位属性
		相对定位
		绝对定位
		堆叠顺序
		固定定位

经典案例.....Page 124

NetCTOSS--admin_list 页面文本	控制字体
	控制文本格式
NetCTOSS--admin_list 页面表格	垂直方向对齐

	边框合并
	边框间距
NetCTOSS--admin_list logo 图片	浮动定位
	float 属性
	clear 属性
NetCTOSS--admin_list 导航栏	显示方式
	display 属性
	光标
	cursor 属性
	列表项标志 list-style-type
NetCTOSS--admin_list 操作提示框	定位属性
	相对定位
	绝对定位
	堆叠顺序

课后作业.....Page 159

## 1. 文本格式化

### 1.1. 文本格式化

#### 1.1.1. 【文本格式化】控制字体

**控制字体**

知识讲解

- 指定字体
  - font-family : value1,value2;
- 字体大小
  - font-size : value;
- 字体加粗
  - font-weight : normal/bold;

```
p
{
  font-family : Times, 'New York', serif ;
  font-size : 14pt ;
  font-weight : bold ;
}
```

+

#### 1.1.2. 【文本格式化】控制文本格式

**控制文本格式**

知识讲解

- 文本颜色
  - color : value;
- 文本排列
  - text-align : left/right/center;
- 文字修饰
  - text-decoration : none/underline;
- 行高
  - line-height : value;
- 首行文本缩进
  - text-indent : value;

+

**控制文本格式（续1）**

知识讲解

```
div {
  color : #FF0000 ;
  text-align : left ;
  text-decoration : underline ;
  height:50px;
  text-indent : 20px ;
  border:1px solid black;
}

div {
  color : #FF0000 ;
  text-align : left ;
  text-decoration : underline ;
  height:50px;
line-height:50px;
  text-indent : 20px ;
  border:1px solid black;
}
```

+




<div>div中的文本</div>

## 2. 表格样式

### 2.1. 表格常用样式属性

#### 2.1.1. 【表格常用样式属性】表格常用样式属性

知识讲解

**表格常用样式属性**

- 边距属性 : padding
- 尺寸属性 : width、height
  - 设置表格或者单元格的尺寸
- 文本格式化属性
- 背景属性 : 设置表格或者单元格的背景色或者背景图像
- border 属性 : 设置表格的边框

+

知识讲解

**表格常用样式属性 (续1)**

**CSS 样式**

```
table { width:200px; border:1px solid black; }
td { height:50px; border:3px solid red; }
```

**html 文档**

```
<table>
<tr>
<td>aa</td>
<td>bb</td>
</tr>
<tr>
<td>cc</td>
<td>dd</td>
</tr>
</table>
```

+

#### 2.1.2. 【表格常用样式属性】垂直方向对齐

知识讲解

**垂直方向对齐**

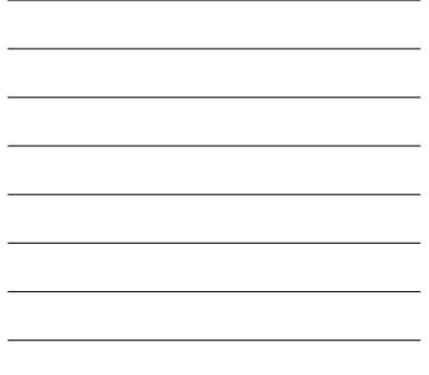
- vertical-align 属性
  - 在表单元格中，设置单元格框中的单元格内容的对齐方式
- 取值
  - vertical-align : top/middle/bottom;

+

111

## 2.2. 表格特有样式属性

### 2.2.1. 【表格特有样式属性】边框合并



**边框合并**

如果设置了单元格的边框，相邻单元格的边框会单独显示，类似于双线边框

- border-collapse 属性：合并相邻的边框
  - 设置是否将表格边框合并为单一边框
- border-collapse : separate/collapse;

```
table {border: 2px dotted black;}  
td {  
    width: 200px;  
    height: 50px;  
    border: 3px solid gray;  
}  
table.separate {border-collapse: separate;}  
table.collapse {border-collapse: collapse;}
```



**边框合并 (续1)**

html 文档

第1行第1列	第1行第2列
--------	--------

知识讲解

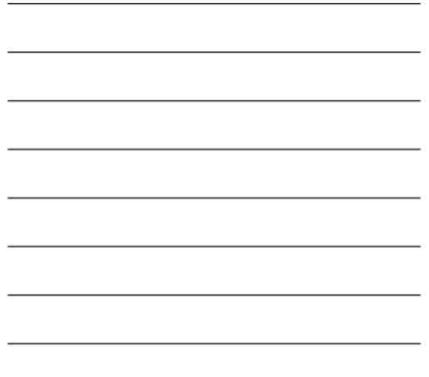
```
<table class="separate">  
<tr>  
    <td class="solid">第1行第1列</td>  
    <td class="dashed">第1行第2列</td>  
</tr>  
</table>
```

第1行第1列	第1行第2列
--------	--------

knowledge

```
<table class="collapse">  
<tr>  
    <td class="solid">第1行第1列</td>  
    <td class="dashed">第1行第2列</td>  
</tr>  
</table>
```

### 2.2.2. 【表格特有样式属性】边框间距



**边框间距**

border-spacing 属性

- 设置相邻单元格的边框间的距离
  - 仅限于分隔单元格边框，即border-collapse 属性为 separate 值的情况下，也称为边框分离模式
- 取值为长度值，可以为该属性指定一个或者两个值
  - 指定一个值：该值同时应用于水平和垂直间距
  - 指定两个值：第一个值指定水平间距，第二个值指定垂直间距，且两个值之间用空格隔开

知识讲解

### 边框边距 (续1)

```
table
{
    border:2px dotted gray;
    border-spacing:10px 20px;
}
td
{
    background-color:#f0f0f0;
    width:200px;
    height:50px;
    border-collapse:separate;
    border:1px solid black;
}
```

**+**

## 3. 浮动定位

### 3.1. 定位概述

#### 3.1.1. 【定位概述】定位概述

知识讲解

### 定位概述

- 定位：定义元素框相对于其正常位置应该出现的位置，或者相对于父元素、另一个元素甚至浏览器窗口本身的位置
- 普通流定位
- 浮动定位
- 相对定位
- 绝对定位

**+**

#### 3.1.2. 【定位概述】普通流定位

知识讲解

### 普通流定位

- 页面中的块级元素框从上到下一个接一个地排列
  - 每一个块级元素都会出现在一个新行中（比如 <p> 元素、<div> 元素）
  - 元素框之间的垂直距离是由框的垂直外边距计算出来的
- 内联元素将在一行中从左到右排列水平布置
  - 不需要从新行开始
  - 可以使用水平内边距、边框和外边距调整它们的间距

**+**

## 3.2. 浮动定位

### 3.2.1. 【浮动定位】浮动概述

知识讲解

**浮动概述**

- 浮动定位是指
  - 将元素排除在普通流之外
  - 将浮动元素放置在包含框的左边或者右边
  - 浮动元素依旧位于包含框之内
- 浮动的框可以向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止
  - 经常使用它来实现特殊的定位效果

+

### 3.2.2. 【浮动定位】浮动定位

知识讲解

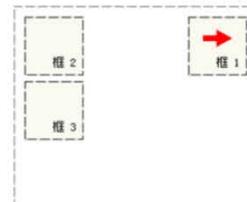
**浮动定位**

- 包含框中有三个元素框，如果把框 1 向右浮动，则它脱离文档流并且向右移动，直到它的右边缘碰到包含框的右边缘

不浮动的框



框 1 向右浮动



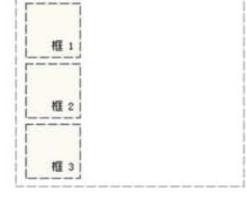
+

知识讲解

**浮动定位（续1）**

- 当框 1 向左浮动时，它脱离文档流并且向左移动，直到它的左边缘碰到包含框的左边缘
- 因为框 1 不再处于文档流中，所以它不占据空间，实际上覆盖住了框 2，使框 2 从视图中消失

不浮动的框



框 1 向左浮动



+

知识讲解

### 浮动定位 (续2)

- 如果把所有三个框都向左移动，那么框1向左浮动直到碰到包含框，另外两个框向左浮动直到碰到前一个浮动框：三个框在同一行上显示

在需要设置多个块级元素同行排列的情况下会非常有用

不浮动的框

框 1  
框 2  
框 3

所有三个框向左浮动

框 1  
框 2  
框 3

+

知识讲解

### 浮动定位 (续3)

- 如果包含框太窄，那么其它浮动块会自动向下移动，直到有足够的空间（如左图所示）
- 如果浮动元素的高度不同，那么当它们向下移动时可能被其它浮动元素“卡住”（如右图所示）

框 1 向左浮动

框 3 下降

所有三个框向左浮动

框 3 被框 1 “卡住了”

+

### 3.2.3. 【浮动定位】float 属性

知识讲解

#### float 属性

- 如果需要设置框浮动在包含框的左边或者右边，可以通过 float 属性来实现
- float 属性定义元素在哪个方向浮动
  - 在 CSS 中，任何元素都可以浮动
- float : none/left/right;

**CSS 样式：**

```
div {  
    border: 1px solid black;  
    width: 100px;  
    height: 50px;  
    float: left;  
}
```

+

**HTML 文档：**

```
<div>1</div>  
<div>2</div>  
<div>3</div>
```

1 2 3

### 3.2.4. 【浮动定位】clear 属性

**clear 属性**

- clear 属性用于清除浮动所带来的影响
- clear : none/left/right/both ;
  - 定义了元素的哪边上不允许出现浮动元素

```
<div style="float:left; border:1px solid black;">div text</div>
<p>p text</p>
```

div text
p text

```
<div style="float:left; border:1px solid black;">div text</div>
<p style="clear:left;">p text</p>
```

## 4. 显示

### 4.1. 显示方式

#### 4.1.1. 【显示方式】显示方式

**显示方式**

- “一切皆为框”：页面上所有的元素都可以显示为框
- 块级元素
  - <div>、<h1> 或 <p> 元素等
  - 这些元素显示为一块内容，即“块框”
- 内联元素 / 行内元素
  - <span>、<a> 元素等
  - 内容显示在行中，即“行内框”

除了默认的显示效果之外，可以使用 display 属性来修改元素框的显示方式，即改变生成的框的类型

#### 4.1.2. 【显示方式】display 属性

**display 属性**

- 取值
  - display : none/block/inline;
- none
  - 让生成的元素根本没有框
  - 该框及其所有内容就不再显示，不占用文档中的空间
- block
  - 让行内元素（比如 <a> 元素）表现得像块级元素一样
- inline
  - 让块级元素（比如 <p> 元素）表现得像内联元素一样

知识讲解

### display 属性 ( 续1 )

**CSS 样式 :**

```
a
{
    border:1px solid black;
    width:70px;
    height:30px;
    background-color:silver;
}
```

**HTML 文档 :**

```
<a href="#">链接1</a>
<a href="#" style="display:block;">链接2</a>
<a href="#" style="display:none;">链接3</a>
<a href="#">链接4</a>
```

+ 链接1  
链接2  
链接3  
链接4

## 4.2. 光标

### 4.2.1. 【光标】光标

知识讲解

### 光标

- 默认情况下，光标会根据用户的操作发生改变
  - 当鼠标悬停在一个链接上时，光标将从指针形状变为手状形状
  - 当鼠标悬停在文本区域时，会显示 I 形状
  - 而当鼠标悬停在一个按钮上时，光标会显示为箭头
- 可以使用 cursor 属性指定显示给用户的鼠标光标类型（形状）
  - 可以为用户提供一种可视化的暗示，提示可以进行的操作

+ \*

### 4.2.2. 【光标】cursor 属性

知识讲解

### cursor 属性

- cursor 属性定义了鼠标指针放在一个元素边界范围内时所用的光标形状
- 可取值
  - default
  - pointer
  - crosshair
  - text
  - wait
  - help 等

+ \*

## 5. 列表样式

### 5.1. 列表样式

#### 5.1.1. 【列表样式】列表项标志 list-style-type

**列表项标志 list-style-type**

**Tarena**  
达内科技

- list-style-type 属性用于控制列表中列表项标志的样式
  - 无序列表：出现在各列表项旁边的圆点
  - 有序列表：可能是字母、数字或另外某种计数体系中的一个符号
- 无序列表取值
  - none : 无标记
  - disc : 实心圆，为默认值
  - circle : 空心圆
  - square : 实心方块

+

**列表项标志 list-style-type ( 续1 )**

**Tarena**  
达内科技

- 有序列表取值
  - none : 无标记
  - decimal : 数字 ( 如 1,2,3,4,5 ) , 为默认值
  - lower-roman : 小写罗马数字 ( 如 i, ii, iii, iv, v )
  - upper-roman : 大写罗马数字 ( 如 I, II, III, IV, V )
  - 等

+

#### 5.1.2. 【列表样式】列表项图像 list-style-image

**列表项图像 list-style-image**

**Tarena**  
达内科技

- list-style-image 属性使用图像来替换列表项的标记
  - 取值为： url() , 指定图像作为有序或无序列表项的标志

HTML 文档：

```
<ul class="parent">
<li>账户
    <ul>
        <li><a href="#">基本信息</a></li>
        <li><a href="#">修改密码</a></li>
    </ul>
</li>
<li>相册
    <ul>
        <li><a href="#">上传照片</a></li>
        <li><a href="#">我的相册</a></li>
    </ul>
</li>
</ul>
```

+

## 列表项图像 list-style-image ( 续1 )

CSS 样式 :

```
li {list-style-type:none;}
li a {text-decoration:none;}
```

```
ul.parent li
{
    list-style-image:url(dh.gif);
}
ul.parent li ul li
{
    list-style-image:url(ph.gif);
}
```

- ✓ 账户
  - \* 基本信息
  - \* 修改密码
- ✓ 相册
  - \* 上传照片
  - \* 我的相册

# 6. 定位

## 6.1. 定位机制

### 6.1.1. 【定位机制】什么是定位

#### 什么是定位

- 普通流定位
  - 页面中的块级元素框从上到下一个接一个地排列
  - 每一个块级元素都会出现在一个新行中
  - 内联元素将在一行中从左到右排列水平布置
- 浮动
- 相对定位
- 绝对定位
- 固定定位

### 6.1.2. 【定位机制】定位属性

#### 定位属性

属性	说明
position	规定元素的定位类型，可取值： static/relative/absolute/fixed
偏移属性	top、bottom、left、right属性，用于定义元素框的偏移位置
z-index	设置元素的堆叠顺序
float/clear	浮动定位属性

#### 使用position 属性和偏移属性

实现普通流定位、相对定位、绝对定位和固定定位

使用float 属性实现浮动定位

其他属性为辅助属性

## 定位属性 (续1)

- position 属性：更改定位模式为相对定位、绝对定位或者固定定位
  - position : static/relative/absolute/fixed;
- 偏移属性：实现元素框位置的偏移
  - top/bottom/right/left: value;
- 堆叠顺序
  - z-index : value;



### 6.1.3. 【定位机制】相对定位

## 相对定位

- 元素仍保持其未定位前的形状
- 元素原本所占的空间仍保留
- 元素框会相对于它原来的位置偏移某个距离
  - 设置垂直或水平位置，让元素相对于它的起点进行移动
- 设置元素为相对定位
  - 首先需要设置position 属性的值为 relative
  - 然后使用left 属性或者right 属性设置水平方向的偏移量
  - 也可以使用top 属性或者bottom 属性设置垂直方向的偏移量



## 相对定位 (续1)

CSS 样式 :

```
div
{
    width: 100px;
    height: 50px;
    border: 1px solid gray;
}
```

HTML 文档 :

```
<body>
<div></div>
<div style="position: relative; left: 20px; top: 10px;"></div>
<p>后续段落</p>
</body>
```

元素原本所占的空间仍保留

后续段落



### 6.1.4. 【定位机制】绝对定位

知识讲解

### 绝对定位

- 将元素的内容从普通流中完全移除，不占据空间
- 并使用偏移属性来固定该元素的位置
  - 相对于最近的已定位祖先元素
  - 如果元素没有已定位的祖先元素，那么它的位置相对于最初的包含块，比如 body 元素
- 设置元素为绝对定位
  - 首先需要设置position 属性的值为 absolute
  - 然后使用left 属性或者right 属性设置元素的水平位置
  - 也可以使用top 属性或者bottom 属性设置元素的垂直位置

知识讲解

### 绝对定位（续1）

CSS 样式 :

```
div.child { width:80px; height:30px; border:1px solid black; background-color:silver; } div.parent { width:200px; height:150px; border:1px solid gray; }
```

HTML 文档 :

```
<body> <div class="parent"> <div class="child">1</div> <div class="child">2</div> </div> </body>
```

没有设置定位属性时

知识讲解

### 绝对定位（续2）

HTML 文档 :

```
<body> <div class="parent"> <div class="child">1</div> <div class="child" style="position : absolute ; top : 15px ; left : 30px ;">2</div> </div> </body>
```

设置为绝对定位

元素框不再占用原有位置，会产生重叠  
相对于最近的已定位祖先元素定位  
如果没有已定位的祖先元素，那么它的位置相对于最初的包含块，比如 body 元素

+ [ ]

121

知识讲解

### 绝对定位 (续3)

CSS 样式 :

```
div.parent
{
    width : 200px;
    height : 150px;
    border : 1px solid gray;
    position : relative;
}
```

会根据父元素 <div> 的位置进行定位  
因为父元素设置了定位属性

后续段落

#### 6.1.5. 【定位机制】堆叠顺序

知识讲解

### 堆叠顺序

- 一旦修改了元素的定位方式，则元素可能会发生堆叠
- 可以使用z-index 属性来控制元素框出现的重叠顺序
- z-index 属性
  - 值为数值：数值越大表示堆叠顺序更高，即离用户更近
  - 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，即元素的显示会接近页面表面
  - 可以设置为负值：表示离用户更远

知识讲解

### 堆叠顺序 (续1)

```
div {
    z-index:100;
    width:200px;
    height:100px;
    border:1px solid gray;
    background-color:White;}
div.first {
    position:relative;
    top:-10px;
    left:30px;
    z-index: 1;
    background-color:Silver; }
div.second {
    position:absolute;
    top:60px;
    left:150px;
    z-index:2;
    background-color:gray; }
```

### 6.1.6. 【定位机制】固定定位

固定定位

Technology **Tarena**  
达内科技

• 将元素的内容固定在页面的某个位置

- 元素从普通流中完全移除，不占用页面空间
- 当用户向下滚动页面时元素框并不随着移动

• 设置固定定位

- 首先需要设置position 属性的值为 fixed
- 通过 left、top、right 以及bottom 这些偏移属性来定义元素的位置

达内IT培训

## 经典案例

### 1. NetCTOSS--admin\_list 页面文本

- 问题

继续为 admin\_list.html 页面添加内容：为页面添加文本并为其定义样式，页面效果如图 - 1 所示。

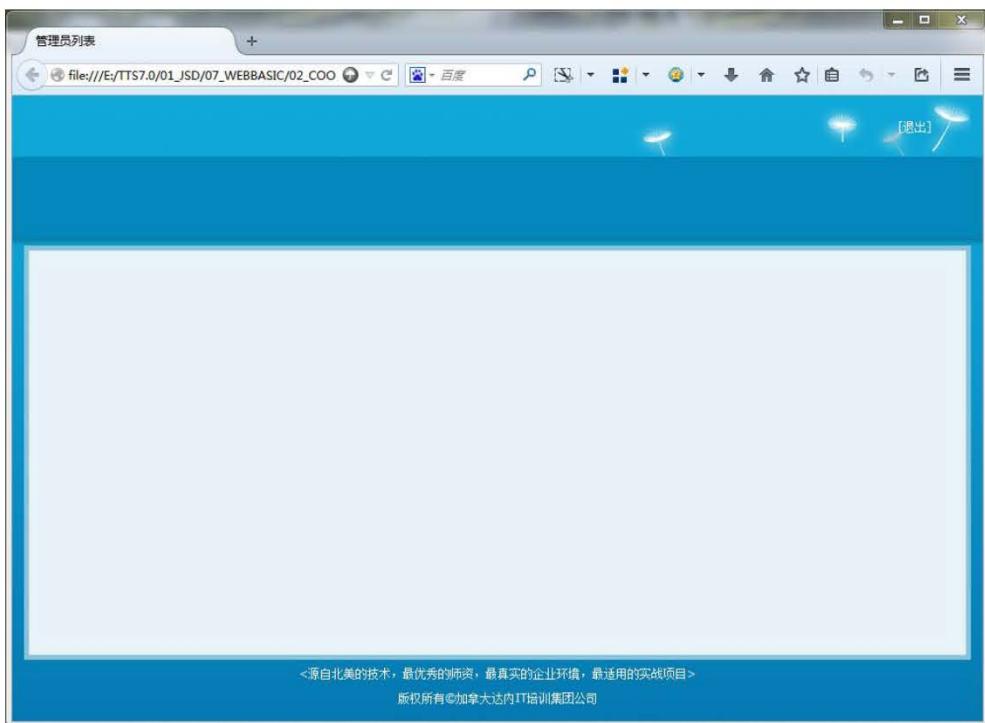


图 - 1

由图 - 1 可以看出，需要为页面添加如下内容：

- 1、页头部分添加超级链接“【退出】”：位于页头部分右侧，垂直方向居中显示；链接文本为白色，没有下划线；当鼠标移入链接文本时，文本加粗且显示下划线；
- 2、页脚部分添加两行文本：水平方向居中，垂直方向居中。

- 方案

实现此案例时，首先需要为 HTML 页面添加新的元素内容，然后在 CSS 文件中添加样式代码。

- 步骤

实现此案例需要按照如下步骤进行。

#### 步骤一：为页面添加文本标记

在页头部分添加超级链接元素，并在页脚部分添加两个段落元素。HTML 页面的代码如下所示：

```
<!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>
<title>管理员列表</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link type="text/css" rel="stylesheet" href="admin_style.css" />
</head>
<body>

    <!--Logo 区域-->
    <div id="header">

        <a href="#">[退出]</a>

    </div>
    <!--导航区域-->
    <div id="navi">
    </div>
    <!--主要区域-->
    <div id="main">
        <form>
            <!--操作区域-->
            <div id="operate">
            </div>
            <!--数据区域-->
            <div id="data">
            </div>
            <!--分页-->
            <div id="pages">
            </div>
        </form>
    </div>
    <!--页脚区域-->
    <div id="footer">

        <p>&lt;源自北美的技术，最优秀的师资，最真实的企业环境，最适用的实战项目
&gt;</p>
        <p>版权所有©加拿大达内 IT 培训集团公司 </p>

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

## 步骤二：为页面定义字体

修改 admin\_style.css 文件中的代码。

首先需要为页面定义整体的字体样式，因此，修改 `<body>` 元素的样式定义：定义字体大小及字体样式。元素选择器 `body` 修改后的 CSS 代码如下所示：

```
body
{
```

```
margin: 0px auto;
padding: 0px;
border: 0;
background-repeat: repeat-x;
background-image: url(images/body_bg.png);

font-size: 12px;
font-family: Tahoma, Geneva, sans-serif;

}
```

### 步骤三：修改页头部分的样式

为实现超级链接在页头部分居右，修改页头部分的样式定义，设置文本居右对齐。修改后的 CSS 代码如下所示：

```
#header
{
    height: 61px;
    background: url(images/top_bg.png) no-repeat;

    text-align:right;

}
```

### 步骤四：为超级链接定义样式

定义超级链接的文本样式：白色字体，没有下划线。为实现如图 - 1 所示的效果，需要设置超级链接的右外边距以及行高。

然后，使用伪类定义鼠标移入超级链接后的样式：字体加粗、有下划线。

增加的 CSS 代码如下所示：

```
#header a
{
    margin-right: 40px;
    line-height: 61px;
    color: #fff;
    text-decoration: none;
}
#header a:hover
{
    font-weight: bold;
    text-decoration: underline;
}
```

### 步骤五：为页脚中的段落文本定义样式

为页脚部分的文本定义样式：为使得两行段落在垂直方向上完全填充整个 div，设置每行段落文本的行高为 div 高度的一半（25px）。添加的 CSS 代码如下所示：

```
/* 页脚文本 */
#footer p
{
    margin:0px auto;
    color:#fff;
    line-height:25px;
```

```
    text-align:center;
}
```

- 完整代码

admin\_list.html 的文件的代码如下所示：

```
<!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>
    <title>管理员列表</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <link type="text/css" rel="stylesheet" href="admin style.css" />
</head>
<body>
    <!--Logo 区域-->
    <div id="header">
        <a href="#">[退出]</a>
    </div>
    <!--导航区域-->
    <div id="navi">
    </div>
    <!--主要区域-->
    <div id="main">
        <form>
            <!--操作区域-->
            <div id="operate">
            </div>
            <!--数据区域-->
            <div id="data">
            </div>
            <!--分页-->
            <div id="pages">
            </div>
        </form>
    </div>
    <!--页脚区域-->
    <div id="footer">
        <p>&lt;源自北美的技术，最优秀的师资，最真实的企业环境，最适用的实战项目
&gt;</p>
        <p>版权所有©加拿大达内 IT 培训集团公司 </p>
    </div>
</body>
</html>
```

admin\_style.css 文件的代码如下所示：

```
/* 全局样式 */
body
{
    margin: 0px auto;
    padding: 0px;
    border: 0;
    background-repeat: repeat-x;
    background-image: url(images/body_bg.png);
    font-size: 12px;
    font-family: Tahoma, Geneva, sans-serif;
}
div
{
```

```
margin: 0px auto;
padding: 0px;
}
/* 整体布局样式 */
#header, #footer
{
    width: 960px;
}
#header
{
    height: 61px;
    background: url(images/top_bg.png) no-repeat;
    text-align:right;
}
#navi
{
    width: 100%;
    height: 91px;
    background: url(images/navigation.png) repeat-x;
}
#main
{
    border:5px solid #8ac1db;
    height: 410px;
    width: 950px;
    background-color: #e8f3f8;
}
#footer
{
    height: 50px;
}

/*正文部分*/
#operate, #data, #pages
{
    width:910px;
}
#operate
{
    height: 30px;
}
#data
{
    height: 340px;
}

#pages
{
    height: 28px;
}

/*页头的超级链接*/
#header a
{
    margin-right: 40px;
    line-height: 61px;
    color: #fff;
    text-decoration: none;
}
#header a:hover
{
    font-weight: bold;
    text-decoration: underline;
}
/* 页脚文本 */
#footer p
{
```

```

margin:0px auto;
color:#fff;
line-height:25px;
text-align:center;
}

```

## 2. NetCTOSS--admin\_list 页面表格

- 问题

继续修改管理员列表页面（ admin\_list.html ），为页面添加表格数据并为表格定义样式，然后为页面各组成部分定义背景，页面效果如图 - 2 所示。



<input type="checkbox"/> 全选	管理员ID	姓名	拥有角色	
<input type="checkbox"/>	1	贾强	超级管理员、角色管理员、账单管理员	修改  删除
<input type="checkbox"/>	2	贾强	超级管理员、帐务账号管理员	修改  删除
<input type="checkbox"/>	3	贾强	超级管理员、帐务账号管理员	修改  删除
<input type="checkbox"/>	4	贾强	超级管理员、帐务账号管理员	修改  删除

图 - 2

- 方案

分析图 - 2 所示的页面，可以发现，实现此案例时，首先需要为页面添加表格，然后需要为表格定义样式。

表格所需要的样式如下：

- 表格：宽 910px，白色背景，文本居中，表格无边框，且边框合并，上下外边距为 5px，水平方向居中显示；
- 单元格：高 32px，文本垂直居中，边框颜色 #CCC；
- 表头单元格：高 35px，文本加粗显示，背景色 #fbbedce；
- 鼠标移入表行：背景色为 #f7f9fd。

- 步骤

实现此案例需要按照如下步骤进行。

### 步骤一：为页面添加表格

修改 admin\_list.html 文件的代码，为页面添加表格。首先在 id 值为 data 的 `<div>` 元素中定义表格。

为便于表格的样式定义，需要为 `<table>` 定义 `id` 属性，并为表头行定义 `class` 属性。因为表格中的按钮也需要样式定义，因此，需要分别为“修改”按钮和“删除”按钮定义 `class` 属性。

所添加的表格的 `html` 代码如下所示：

```
<!--其他部分代码，略-->
<!--数据区域-->
<div id="data">

    <table id="datalist">
        <tr class="header">
            <td>
                <input type="checkbox" />
                <span>全选</span>
            </td>
            <td>管理员 ID</td>
            <td>姓名</td>
            <td>拥有角色</td>
            <td></td>
        </tr>
        <tr>
            <td><input type="checkbox" /></td>
            <td>1</td>
            <td>贾强</td>
            <td>超级管理员、角色管理员、账单管理员</td>
            <td>
                <input type="button" class="btn_modify" value="修改" />
                <input type="button" class="btn_delete" value="删除" />
            </td>
        </tr>
        <tr>
            <td><input type="checkbox" /></td>
            <td>2</td>
            <td>贾强</td>
            <td>超级管理员、帐务账号管理員</td>
            <td>
                <input type="button" class="btn_modify" value="修改" />
                <input type="button" class="btn_delete" value="删除" />
            </td>
        </tr>
        <tr>
            <td><input type="checkbox" /></td>
            <td>3</td>
            <td>贾强</td>
            <td>超级管理员、帐务账号管理員</td>
            <td>
                <input type="button" class="btn_modify" value="修改" />
                <input type="button" class="btn_delete" value="删除" />
            </td>
        </tr>
    </table>
</div>
```

```
        <input type="button" class="btn_modify" value="修改" />
        <input type="button" class="btn_delete" value="删除" />
    </td>
</tr>
<tr>
    <td><input type="checkbox" /></td>
    <td>4</td>
    <td>贾强</td>
    <td>超级管理员、帐务账号管理员</td>
    <td>
        <input type="button" class="btn_modify" value="修改" />
        <input type="button" class="btn_delete" value="删除" />
    </td>
</tr>
</table>

</div>
<!--其他部分代码，略-->
```

## 步骤二：为表格定义样式

HTML 代码修改完毕后，修改 admin\_style.css 文件中的样式代码。

首先为表格定义外观，设置其宽度、边距、背景色以及边框。添加的 CSS 代码如下所示：

```
#datalist
{
    width: 910px;
    margin: 5px auto;
    text-align: center;
    background-color:#ffff;
    border:0;
    border-collapse:collapse;
}
```

## 步骤三：定义单元格的样式

为表格中的单元格定义样式：高度、边框以及垂直方向上的对齐。添加的 CSS 代码如下所示：

```
#datalist td
{
    height: 32px;
    vertical-align:middle;
    border: 1px solid #CCC;
}
```

## 步骤四：定义鼠标移入表行的样式

当鼠标移入某行时，需要改变其背景色。因此，需要为表格中的表行定义 hover 状态下的样式。添加的 CSS 代码如下所示：

```
#datalist tr:hover
{
```

```
background-color:#f7f9fd;
}
```

### 步骤五：定义表格中表头行的样式

为表格中的表头行定义样式，设置单元格中的字体加粗、单元格的高度以及背景色。添加的 CSS 代码如下所示：

```
#datalist tr.header td
{
    font-weight: bold;
    height: 35px;
    background: #fbcedc;
}
```

### 步骤六：为表格中的按钮定义样式

首先为表格中的两种按钮定义相同的样式，如宽度、字体颜色、背景色等；然后分别为“修改”和“删除”按钮定义背景图片。添加的 CSS 代码如下所示：

```
/* 按钮样式 */
input.btn_modify, input.btn_delete
{
    margin-left: 3px;
    width: 58px;
    cursor: pointer;
    color: #000;
    border: 0;
    background-color: #fff;
    background-repeat: no-repeat;
    background-position: left center;
}
input.btn_modify
{
    background-image: url(images/modification.png);
}
input.btn_delete
{
    background-image: url(images/delete.png);
}
```

### • 完整代码

`admin_list.html` 的文件的代码如下所示：

```
<!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>
        <title>管理员列表</title>
        <meta http-equiv="content-type" content="text/html; charset=utf-8" />
        <link type="text/css" rel="stylesheet" href="admin_style.css" />
    </head>
    <body>
        <!--Logo 区域-->
        <div id="header">
            <a href="#">[退出]</a>
        </div>
```

```

<!--导航区域-->
<div id="navi">
</div>
<!--主要区域-->
<div id="main">
  <form>
    <!--操作区域-->
    <div id="operate">
    </div>
    <!--数据区域-->
    <div id="data">
      <table id="datalist">
        <tr class="header">
          <td>
            <input type="checkbox" />
            <span>全选</span>
          </td>
          <td>管理员 ID</td>
          <td>姓名</td>
          <td>拥有角色</td>
          <td></td>
        </tr>
        <tr>
          <td><input type="checkbox" /></td>
          <td>1</td>
          <td>贾强</td>
          <td>超级管理员、角色管理员、账单管理员</td>
          <td>
            <input type="button" class="btn_modify" value="修改" />
            <input type="button" class="btn_delete" value="删除" />
          </td>
        </tr>
        <tr>
          <td><input type="checkbox" /></td>
          <td>2</td>
          <td>贾强</td>
          <td>超级管理员、帐务账号管理员</td>
          <td>
            <input type="button" class="btn_modify" value="修改" />
            <input type="button" class="btn_delete" value="删除" />
          </td>
        </tr>
        <tr>
          <td><input type="checkbox" /></td>
          <td>3</td>
          <td>贾强</td>
          <td>超级管理员、帐务账号管理员</td>
          <td>
            <input type="button" class="btn_modify" value="修改" />
            <input type="button" class="btn_delete" value="删除" />
          </td>
        </tr>
        <tr>
          <td><input type="checkbox" /></td>
          <td>4</td>
          <td>贾强</td>
          <td>超级管理员、帐务账号管理员</td>
          <td>
        
```

```

        <input type="button" class="btn modify" value="修改" />
        <input type="button" class="btn delete" value="删除" />
    </td>
</tr>
</table>
</div>
<!--分页-->
<div id="pages">
</div>
</form>
</div>
<!--页脚区域-->
<div id="footer">
    <p>&lt;源自北美的技术，最优秀的师资，最真实的企业环境，最适用的实战项目
&gt;</p>
    <p>版权所有©加拿大达内 IT 培训集团公司 </p>
</div>
</body>
</html>

```

admin\_style.css 文件中的代码如下所示：

```

/* 全局样式 */
body
{
    margin: 0px auto;
    padding: 0px;
    border: 0;
    background-repeat: repeat-x;
    background-image: url(images/body_bg.png);
    font-size: 12px;
    font-family: Tahoma, Geneva, sans-serif;
}
div
{
    margin: 0px auto;
    padding: 0px;
}
/* 整体布局样式 */
#header, #footer
{
    width: 960px;
}
#header
{
    height: 61px;
    background: url(images/top_bg.png) no-repeat;
    text-align:right;
}
#navi
{
    width: 100%;
    height: 91px;
    background: url(images/navigation.png) repeat-x;
}
#main
{
    border:5px solid #8ac1db;
    height: 410px;
    width: 950px;
    background-color: #e8f3f8;
}
#footer

```

```
{  
    height: 50px;  
}  
#pages  
{  
    height: 28px;  
}  
  
/*正文部分*/  
#operate,#data,#pages  
{  
    width:910px;  
}  
#operate  
{  
    height: 30px;  
}  
#data  
{  
    height: 340px;  
}  
  
/*页头的超级链接*/  
#header a  
{  
    margin-right: 40px;  
    line-height: 61px;  
    color: #fff;  
    text-decoration: none;  
}  
#header a:hover  
{  
    font-weight: bold;  
    text-decoration: underline;  
}  
/* 页脚文本 */  
#footer p  
{  
    margin:0px auto;  
    color:#fff;  
    line-height:25px;  
    text-align:center;  
}  
  
/* 数据表格部分 */  
#datalist  
{  
    width: 910px;  
    margin: 5px auto;  
    text-align: center;  
    background-color:#fff;  
    border:0;  
    border-collapse:collapse;  
}  
#datalist td  
{  
    height: 32px;  
    vertical-align:middle;  
    border: 1px solid #CCC;  
}  
#datalist tr:hover  
{  
    background-color:#f7f9fd;  
}  
#datalist tr.header td  
{  
    font-weight: bold;
```

```

height: 35px;
background: #fbedce;
}

/* 按钮样式 */
input.btn_modify, input.btn_delete
{
    margin-left: 3px;
    width: 58px;
    cursor: pointer;
    color: #000;
    border: 0;
    background-color: #fff;
    background-repeat: no-repeat;
    background-position: left center;
}
input.btn_modify
{
    background-image: url(images/modification.png);
}
input.btn_delete
{
    background-image: url(images/delete.png);
}

```

### 3. NetCTOSS--admin\_list logo 图片

- 问题

继续修改管理员列表页面（ admin\_list.html ），为页头部分添加 logo 图片，页面效果如图 - 3 所示。



图 - 3

- 方案

分析图 - 3 所示的页面，可以发现，相对于上一个案例，此案例的页面只是在页头部分添加了 logo 图片。

因此，实现此案例时，首先需要为页面的页头部分添加图片，然后为该图片设置样式即可。因为页头部分已经设置了文本居右显示，如果需要其中的图片显示在左边，则需要设置该图片向左浮动。

### • 步骤

实现此案例需要按照如下步骤进行。

#### 步骤一：为页头添加图片

修改 `admin_list.html` 文件的代码，为页头添加图片。修改后的 `html` 代码如下所示：

```
<!--其他部分代码，略-->
<!--Logo 区域-->
<div id="header">

    <a href="#">[退出]</a>
</div>
<!--其他部分代码，略-->
```

#### 步骤二：为表格定义样式

`Html` 代码修改完毕后，修改 `admin_style.css` 文件中的样式代码。

设置图片向左浮动，所添加的 `CSS` 代码如下所示：

```
#header img
{
    float:left;
}
```

### • 完整代码

`admin_list.html` 的文件的代码如下所示：

```
<!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>
        <title>管理员列表</title>
        <meta http-equiv="content-type" content="text/html; charset=utf-8" />
        <link type="text/css" rel="stylesheet" href="admin_style.css" />
    </head>
    <body>
        <!--Logo 区域-->
        <div id="header">
            
        </div>
        <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;">
            <thead>
                <tr style="background-color: #cccccc; height: 30px;">
                    <th style="text-align: center; width: 10%;">序号用户名操作1admin<a href="#">退出</td>
                </tr>
                <tr>
                    <td style="text-align: center;">2tarena<a href="#">退出</td>
                </tr>
            </tbody>
        </table>
    </body>
</html>
```

```

        <a href="#">[退出]</a>
    </div>
    <!--导航区域-->
    <div id="navi">
    </div>
    <!--主要区域-->
    <div id="main">
        <form>
            <!--操作区域-->
            <div id="operate">
            </div>
            <!--数据区域-->
            <div id="data">
                <table id="datalist">
                    <tr class="header">
                        <td>
                            <input type="checkbox" />
                            <span>全选</span>
                        </td>
                        <td>管理员 ID</td>
                        <td>姓名</td>
                        <td>拥有角色</td>
                        <td></td>
                    </tr>
                    <tr>
                        <td><input type="checkbox" /></td>
                        <td>1</td>
                        <td>贾强</td>
                        <td>超级管理员、角色管理员、账单管理员</td>
                        <td>
                            <input type="button" class="btn_modify" value="修改" />
                            <input type="button" class="btn_delete" value="删除" />
                        </td>
                    </tr>
                    <tr>
                        <td><input type="checkbox" /></td>
                        <td>2</td>
                        <td>贾强</td>
                        <td>超级管理员、帐务账号管理员</td>
                        <td>
                            <input type="button" class="btn_modify" value="修改" />
                            <input type="button" class="btn_delete" value="删除" />
                        </td>
                    </tr>
                    <tr>
                        <td><input type="checkbox" /></td>
                        <td>3</td>
                        <td>贾强</td>
                        <td>超级管理员、帐务账号管理员</td>
                        <td>
                            <input type="button" class="btn_modify" value="修改" />
                            <input type="button" class="btn_delete" value="删除" />
                        </td>
                    </tr>
                    <tr>
                        <td><input type="checkbox" /></td>
                        <td>4</td>
                        <td>贾强</td>
                    </tr>
                </table>
            </div>
        </form>
    </div>

```

```

        <td>超级管理员、帐务账号管理员</td>
        <td>
            <input type="button" class="btn modify" value="修改" />
            <input type="button" class="btn delete" value="删除" />
        </td>
    </tr>
</table>
</div>
<!--分页-->
<div id="pages">
</div>
</form>
</div>
<!--页脚区域-->
<div id="footer">
    <p>&lt;源自北美的技术，最优秀的师资，最真实的企业环境，最适用的实战项目
&gt;</p>
    <p>版权所有©加拿大达内 IT 培训集团公司 </p>
</div>
</body>
</html>

```

`admin_style.css` 文件中的代码如下所示：

```

/* 全局样式 */
body
{
    margin: 0px auto;
    padding: 0px;
    border: 0;
    background-repeat: repeat-x;
    background-image: url(images/body_bg.png);
    font-size: 12px;
    font-family: Tahoma, Geneva, sans-serif;
}
div
{
    margin: 0px auto;
    padding: 0px;
}
/* 整体布局样式 */
#header, #footer
{
    width: 960px;
}
#header
{
    height: 61px;
    background: url(images/top_bg.png) no-repeat;
    text-align:right;
}
#navi
{
    width: 100%;
    height: 91px;
    background: url(images/navigation.png) repeat-x;
}
#main
{
    border:5px solid #8ac1db;
    height: 410px;
    width: 950px;
}

```

```
background-color: #e8f3f8;
}
#footer
{
    height: 50px;
}
#pages
{
    height: 28px;
}

/*正文部分*/
#operate,#data,#pages
{
    width:910px;
}
#operate
{
    height: 30px;
}
#data
{
    height: 340px;
}

/*页头的超级链接*/
#header a
{
    margin-right: 40px;
    line-height: 61px;
    color: #fff;
    text-decoration: none;
}
#header a:hover
{
    font-weight: bold;
    text-decoration: underline;
}
/* 页脚文本 */
#footer p
{
    margin:0px auto;
    color:#fff;
    line-height:25px;
    text-align:center;
}

/* 数据表格部分 */
#datalist
{
    width: 910px;
    margin: 5px auto;
    text-align: center;
    background-color:#fff;
    border:0;
    border-collapse:collapse;
}
#datalist td
{
    height: 32px;
    vertical-align:middle;
    border: 1px solid #CCC;
}
#datalist tr:hover
{
    background-color:#f7f9fd;
}
```

```
#datalist tr.header td
{
    font-weight: bold;
    height: 35px;
    background: #fbedce;
}

/* 按钮样式 */
input.btn_modify, input.btn_delete
{
    margin-left: 3px;
    width: 58px;
    cursor: pointer;
    color: #000;
    border: 0;
    background-color: #fff;
    background-repeat: no-repeat;
    background-position: left center;
}
input.btn_modify
{
    background-image: url(images/modification.png);
}
input.btn_delete
{
    background-image: url(images/delete.png);
}
/* 页头 logo 部分 */
#header img
{
    float: left;
}
```

#### 4. NetCTOSS--admin\_list 导航栏

- 问题

继续修改管理员列表页面（ admin\_list.html ），为页面添加导航栏信息并为其定义样式，页面效果如图 - 4 所示。



图 - 4

## • 方案

分析图 - 4 所示的页面，可以发现，实现此案例时，首先需要使用列表为页面添加导航菜单栏，然后需要为列表定义样式。

列表所需要的样式如下：

- 列表：宽度 960px，上下外边距 3px，内边距 5px，居中显示；
- 列表项：左右边距 14px，上下边距 0px；
- 列表项中的链接：宽 68px，高 77px。

## • 步骤

实现此案例需要按照如下步骤进行。

### 步骤一：为页面添加导航菜单栏

修改 admin\_list.html 文件的代码，为页面的导航部分添加导航链接。使用无序列表定义整个菜单，每个列表项为一个菜单项，列表项中包含超级链接，用于实现功能导航。

为便于样式的维护(比如图片的更换)，不将图片作为 `<img>` 元素加入到 `<a>` 元素中，而是将图片作为 `<a>` 元素的背景图片。因此，需要为各个链接定义不同的样式类。

修改后的 html 代码如下所示：

```
<!--其他部分代码，略-->
<!--导航区域-->
<div id="navi">
```

```

<ul id="menu">
    <li><a href="#" class="index_off"></a></li>
    <li><a href="#" class="role_off"></a></li>
    <li><a href="#" class="admin_off"></a></li>
    <li><a href="#" class="fee_off"></a></li>
    <li><a href="#" class="account_off"></a></li>
    <li><a href="#" class="service_off"></a></li>
    <li><a href="#" class="bill_off"></a></li>
    <li><a href="#" class="report_off"></a></li>
    <li><a href="#" class="information_off"></a></li>
    <li><a href="#" class="password_off"></a></li>
</ul>

</div>
<!--其他部分代码，略-->

```

## 步骤二：为列表定义样式

为列表定义宽度和边距，并设置为不显示列表项的标识符。添加的 CSS 代码如下所示：

```

#menu
{
    margin: 3px auto;
    width: 960px;
    padding: 5px;
    list-style-type: none;
}

```

## 步骤三：为列表项定义样式

列表项是块级元素，默认情况下，会分行显示。为了实现各列表项在同一行显示，需要设置列表项向左浮动，并设置适当的间距。添加的 CSS 代码如下所示：

```

#menu li
{
    float: left;
    margin: 0 14px;
}

```

## 步骤四：为列表项中的超级链接定义样式

每个列表项中包含一个超级链接，用于实现功能导航。

为了让超级链接显示整个背景图片，需要为其设置适当的高度和宽度。因此，需要将超级链接元素显示为块级元素。添加的 CSS 代码如下所示：

```

#menu a
{
    display: block;
    width: 68px;
    height: 77px;
}

```

## 步骤五：为每个导航链接定义不同的背景图片

为菜单中的每个超级链接定义样式，以设置背景图片。添加的 CSS 代码如下所示：

```
.index off
{
    background: url(images/index_out.png) no-repeat;
}
.role_off
{
    background: url(images/role_out.png) no-repeat;
}
.admin_off
{
    background: url(images/admin_out.png) no-repeat;
}
.fee off
{
    background: url(images/fee_out.png) no-repeat;
}
.account_off
{
    background: url(images/account_out.png) no-repeat;
}
.service off
{
    background: url(images/service_out.png) no-repeat;
}
.bill off
{
    background: url(images/bill_out.png) no-repeat;
}
.report_off
{
    background: url(images/report_out.png) no-repeat;
}
.information_off
{
    background: url(images/information_out.png) no-repeat;
}
.password off
{
    background: url(images/password_out.png) no-repeat;
}
```

### • 完整代码

admin\_list.html 的文件的代码如下所示：

```
<!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>
    <title>管理员列表</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <link type="text/css" rel="stylesheet" href="admin_style.css" />
</head>
<body>
    <!--Logo 区域-->
    <div id="header">
        
        <a href="#">[退出]</a>
    </div>
    <!--导航区域-->
    <div id="navi">
        <ul id="menu">
            <li><a href="#" class="index off"></a></li>
            <li><a href="#" class="role_off"></a></li>
            <li><a href="#" class="admin_off"></a></li>
```

```

<li><a href="#" class="fee_off"></a></li>
<li><a href="#" class="account_off"></a></li>
<li><a href="#" class="service_off"></a></li>
<li><a href="#" class="bill_off"></a></li>
<li><a href="#" class="report_off"></a></li>
<li><a href="#" class="information_off"></a></li>
<li><a href="#" class="password_off"></a></li>
</ul>
</div>
<!--主要区域-->
<div id="main">
  <form>
    <!--操作区域-->
    <div id="operate">
    </div>
    <!--数据区域-->
    <div id="data">
      <table id="datalist">
        <tr class="header">
          <td>
            <input type="checkbox" />
            <span>全选</span>
          </td>
          <td>管理员 ID</td>
          <td>姓名</td>
          <td>拥有角色</td>
          <td></td>
        </tr>
        <tr>
          <td><input type="checkbox" /></td>
          <td>1</td>
          <td>贾强</td>
          <td>超级管理员、角色管理员、账单管理员</td>
          <td>
            <input type="button" class="btn_modify" value="修改" />
            <input type="button" class="btn_delete" value="删除" />
          </td>
        </tr>
        <tr>
          <td><input type="checkbox" /></td>
          <td>2</td>
          <td>贾强</td>
          <td>超级管理员、帐务账号管理员</td>
          <td>
            <input type="button" class="btn_modify" value="修改" />
            <input type="button" class="btn_delete" value="删除" />
          </td>
        </tr>
        <tr>
          <td><input type="checkbox" /></td>
          <td>3</td>
          <td>贾强</td>
          <td>超级管理员、帐务账号管理员</td>
          <td>
            <input type="button" class="btn_modify" value="修改" />
            <input type="button" class="btn_delete" value="删除" />
          </td>
        </tr>
      </table>
    </div>
  </form>
</div>

```

```

        <td><input type="checkbox" /></td>
        <td>4</td>
        <td>贾强</td>
        <td>超级管理员、帐务账号管理员</td>
        <td>
            <input type="button" class="btn_modify" value="修改" />
            <input type="button" class="btn_delete" value="删除" />
        </td>
    </tr>
</table>
</div>
<!--分页--&gt;
&lt;div id="pages"&gt;
&lt;/div&gt;
&lt;/form&gt;
&lt;/div&gt;
<!--页脚区域--&gt;
&lt;div id="footer"&gt;
    &lt;p&gt;&amp;lt;源自北美的技术，最优秀的师资，最真实的企业环境，最适用的实战项目&amp;gt;&lt;/p&gt;
    &lt;p&gt;版权所有©加拿大达内 IT 培训集团公司 &lt;/p&gt;
&lt;/div&gt;
&lt;/body&gt;
&lt;/html&gt;
</pre>

```

`admin_style.css` 文件中的代码如下所示：

```

/* 全局样式 */
body
{
    margin: 0px auto;
    padding: 0px;
    border: 0;
    background-repeat: repeat-x;
    background-image: url(images/body_bg.png);
    font-size: 12px;
    font-family: Tahoma, Geneva, sans-serif;
}
div
{
    margin: 0px auto;
    padding: 0px;
}
/* 整体布局样式 */
#header, #footer
{
    width: 960px;
}
#header
{
    height: 61px;
    background: url(images/top_bg.png) no-repeat;
    text-align:right;
}
#navi
{
    width: 100%;
    height: 91px;
    background: url(images/navigation.png) repeat-x;
}
#main
{

```

```
border:5px solid #8ac1db;
height: 410px;
width: 950px;
background-color: #e8f3f8;
}
#footer
{
    height: 50px;
}
#pages
{
    height: 28px;
}
/*正文部分*/
#operate,#data,#pages
{
    width:910px;
}
#operate
{
    height: 30px;
}
#data
{
    height: 340px;
}

/*页头的超级链接*/
#header a
{
    margin-right: 40px;
    line-height: 61px;
    color: #fff;
    text-decoration: none;
}
#header a:hover
{
    font-weight: bold;
    text-decoration: underline;
}
/* 页脚文本 */
#footer p
{
    margin:0px auto;
    color:#fff;
    line-height:25px;
    text-align:center;
}

/* 数据表格部分 */
#datalist
{
    width: 910px;
    margin: 5px auto;
    text-align: center;
    background-color:#fff;
    border:0;
    border-collapse:collapse;
}
#datalist td
{
    height: 32px;
    vertical-align:middle;
    border: 1px solid #CCC;
}
#datalist tr:hover
{
```

```
background-color:#f7f9fd;
}
#datalist tr.header td
{
    font-weight: bold;
    height: 35px;
    background: #fbcedce;
}

/* 按钮样式 */
input.btn_modify, input.btn_delete
{
    margin-left: 3px;
    width: 58px;
    cursor: pointer;
    color: #000;
    border:0;
    background-color:#fff;
    background-repeat:no-repeat;
    background-position:left center;
}
input.btn modify
{
    background-image: url(images/modification.png);
}
input.btn delete
{
    background-image: url(images/delete.png);
}
/* 页头 logo 部分*/
#header img
{
    float:left;
}

/* 导航菜单栏 */
#menu
{
    margin: 3px auto;
    width:960px;
    padding:5px;
    list-style-type:none;
}
#menu li
{
    float: left;
    margin: 0 14px;
}
#menu a
{
    display: block;
    width: 68px;
    height: 77px;
}
.index off
{
    background: url(images/index_out.png) no-repeat;
}
.role off
{
    background: url(images/role out.png) no-repeat;
}
.admin_off
{
    background: url(images/admin out.png) no-repeat;
}
.fee_off
```

```

{
    background: url(images/fee_out.png) no-repeat;
}
.account_off
{
    background: url(images/account_out.png) no-repeat;
}
.service_off
{
    background: url(images/service_out.png) no-repeat;
}
.bill_off
{
    background: url(images/bill_out.png) no-repeat;
}
.report_off
{
    background: url(images/report_out.png) no-repeat;
}
.information_off
{
    background: url(images/information_out.png) no-repeat;
}
.password_off
{
    background: url(images/password_out.png) no-repeat;
}

```

## 5. NetCTOSS--admin\_list 操作提示框

- 问题

继续修改管理员列表页面( admin\_list.html ) ,为页面添加操作完成后的提示信息。如果操作成功，则提示信息的页面效果如图 - 5 所示。



图 - 5

如果操作失败，则页面效果如图 - 6 所示。



图 - 6

## • 方案

分析图 - 5 和图 - 6 所示的页面，可以发现，实现此案例时，首先需要使用 `<div>` 元素为页面添加操作信息提示框，用于显示操作提示的文本、图片以及关闭操作的图标。

图中的操作信息提示框位于页面其他元素的上方，且不占用页面空间，因此，需要设置其定位方式为绝对定位。

操作可能成功，也可能失败，两种操作结果的样式不相同。因此，需要为操作信息提示框定义两套样式，分别如图 - 5 和图 - 6 所示。

分析图 - 5 和图 - 6 可以看出，两套样式中的大部分内容相同，只是背景图片、字体颜色和文本内容不同。因此，首先定义操作信息提示框的公共样式：

- 框：宽 400px，高 70px，左内边距 40px，边框颜色为 #f57676，上偏移 90px，左偏移 30%；
- 框中文本：字体大小为 16px；
- 背景：背景色为 #fdecec，背景图片不平铺，水平偏移 10px，垂直居中。

操作成功的提示框的特有样式为：

- 文本：颜色为 #af6606
- 背景：背景图片为 ok.png

操作失败的提示框的特有样式为：

- 文本：颜色为 #f00
- 背景：背景图片为 warning.png

### • 步骤

实现此案例需要按照如下步骤进行。

#### 步骤一：为页面添加操作信息提示框

为 admin\_list.html 页面添加 html 代码，以定义操作信息提示框。

将操作信息提示框定义在 id 属性值为 main 的 <div> 元素中，这样，可以相对于该元素设置提示框的偏移位置。

操作信息提示框中需要使用 <img> 元素表示“关闭”的操作，并使用 <span> 元素定义信息文本。为便于对整个框定义样式，需要为其定义 id 属性。

修改后的 html 代码如下所示：

```
<!--其他部分代码，略-->
<!--主要区域-->
<div id="main">
    <form>
        <!--操作区域-->
        <div id="operate">
        </div>

        <!--作信息提示框-->
        <div id="operate_result_info">
            
            <span>操作成功！</span>
        </div>

        <!--数据区域-->
        <div id="data">
            <!--其他部分代码，略-->
        </div>
        <!--分页-->
        <div id="pages">
        </div>
    </form>
</div>
```

#### 步骤二：为操作信息提示框定义样式

定义信息提示框的样式，新添加的 CSS 代码如下所示：

```
#operate_result_info
{
    width: 400px;
    height: 70px;
    line-height: 70px;
    padding-left: 40px;
    font-size: 16px;
    position: absolute;
```

```
z-index: 100;
top: 90px;
left: 30%;
background-color: #fdececc;
border: 1px solid #f57676;
background-repeat: no-repeat;
background-position: 10px center;
}
```

注意：需要定义其为绝对定位，并相对于其父元素（id 值为 main 的 `<div>` 元素）设置偏移距离。

### 步骤三：修改其父元素的定位方式

为保证提示框相对于其父元素偏移，而不是相对于 `<body>` 偏移，需要修改其父元素的定位属性为相对定位。

修改后的 CSS 代码如下所示（只是添加了相对定位方式）：

```
#main
{
    background: #e8f3f8;
    border: 5px solid #8ac1db;
    min-height: 410px;
    width: 950px;
    max-height: 450px;
    padding: 5px;

    position: relative;
}
```

### 步骤四：定义操作信息提示框中图片的位置

设置信息提示框中的关闭图片向右浮动，并设置其边距，新添加的 CSS 代码如下所示：

```
#operate result info img
{
    float: right;
    margin-right: 10px;
    margin-top: 10px;
    cursor: pointer;
}
```

### 步骤五：定义操作成功的样式

定义样式类 `operate_success`，作为操作成功的样式，新添加的 CSS 代码如下所示：

```
div.operate success
{
    color: #af6606;
    background-image: url(images/ok.png);
}
```

样式准备完毕后，在页面上使用样式类，以测试效果。即，为操作框的 `<div>` 元素定义 `class` 属性值为 `operate_success`。HTML 代码如下所示：

```
<div id="operate_result_info" class="operate_success">
    
    <span>操作成功！</span>
</div>
```

## 步骤六：定义操作失败的样式

定义样式类 `operate_fail`，作为操作失败的样式，新添加的 CSS 代码如下所示：

```
div.operate_fail
{
    color: #f00;
    background-image: url(images/warning.png);
```

样式准备完毕后，在页面上使用样式类，以测试效果。即，为操作框的 `<div>` 元素定义 `class` 属性值为 `operate_fail`。HTML 代码如下所示：

```
<div id="operate_result_info" class="operate_fail">
    
    <span>操作失败！</span>
</div>
```

实际项目中，只需要根据操作结果选择相应的样式类即可。

### • 完整代码

`admin_list.html` 的文件的代码如下所示：

```
<!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>
        <title>管理员列表</title>
        <meta http-equiv="content-type" content="text/html; charset=utf-8" />
        <link type="text/css" rel="stylesheet" href="admin_style.css" />
    </head>
    <body>
        <!--Logo 区域-->
        <div id="header">
            
            <a href="#">[退出]</a>
        </div>
        <!--导航区域-->
        <div id="navi">
            <ul id="menu">
                <li><a href="#" class="index_off"></a></li>
                <li><a href="#" class="role_off"></a></li>
                <li><a href="#" class="admin_off"></a></li>
                <li><a href="#" class="fee_off"></a></li>
                <li><a href="#" class="account_off"></a></li>
                <li><a href="#" class="service_off"></a></li>
                <li><a href="#" class="bill_off"></a></li>
                <li><a href="#" class="report_off"></a></li>
                <li><a href="#" class="information_off"></a></li>
```

```

<li><a href="#" class="password off"></a></li>
</ul>
</div>
<!--主要区域-->
<div id="main">
<form>
<!--操作区域-->
<div id="operate">
</div>
<!--操作信息提示框-->
<div id="operate result info" class="operate success">
    
    <span>操作成功! </span>
</div>
<!--数据区域-->
<div id="data">
    <table id="datalist">
        <tr class="header">
            <td>
                <input type="checkbox" />
                <span>全选</span>
            </td>
            <td>管理员 ID</td>
            <td>姓名</td>
            <td>拥有角色</td>
            <td></td>
        </tr>
        <tr>
            <td><input type="checkbox" /></td>
            <td>1</td>
            <td>贾强</td>
            <td>超级管理员、角色管理员、账单管理员</td>
            <td>
                <input type="button" class="btn_modify" value="修改" />
                <input type="button" class="btn_delete" value="删除" />
            </td>
        </tr>
        <tr>
            <td><input type="checkbox" /></td>
            <td>2</td>
            <td>贾强</td>
            <td>超级管理员、帐务账号管理员</td>
            <td>
                <input type="button" class="btn_modify" value="修改" />
                <input type="button" class="btn_delete" value="删除" />
            </td>
        </tr>
        <tr>
            <td><input type="checkbox" /></td>
            <td>3</td>
            <td>贾强</td>
            <td>超级管理员、帐务账号管理员</td>
            <td>
                <input type="button" class="btn_modify" value="修改" />
                <input type="button" class="btn_delete" value="删除" />
            </td>
        </tr>
        <tr>
            <td><input type="checkbox" /></td>
        </tr>
    </table>
</div>

```

```

        <td>4</td>
        <td>贾强</td>
        <td>超级管理员、帐务账号管理员</td>
        <td>
            <input type="button" class="btn_modify" value="修改" />
            <input type="button" class="btn_delete" value="删除" />
        </td>
    </tr>
</table>
</div>
<!--分页-->
<div id="pages">
</div>
</form>
</div>
<!--页脚区域-->
<div id="footer">
    <p>&lt;源自北美的技术，最优秀的师资，最真实的企业环境，最适用的实战项目
&gt;</p>
    <p>版权所有©加拿大达内 IT 培训集团公司 </p>
</div>
</body>
</html>

```

`admin_style.css` 文件中的代码如下所示：

```

/* 全局样式 */
body
{
    margin: 0px auto;
    padding: 0px;
    border: 0;
    background-repeat: repeat-x;
    background-image: url(images/body_bg.png);
    font-size: 12px;
    font-family: Tahoma, Geneva, sans-serif;
}
div
{
    margin: 0px auto;
    padding: 0px;
}
/* 整体布局样式 */
#header, #footer
{
    width: 960px;
}
#header
{
    height: 61px;
    background: url(images/top_bg.png) no-repeat;
    text-align:right;
}
#navi
{
    width: 100%;
    height: 91px;
    background: url(images/navigation.png) repeat-x;
}
#main
{
    border:5px solid #8ac1db;
}

```

```
height: 410px;
width: 950px;
background-color: #e8f3f8;
}
#footer
{
    height: 50px;
}
#pages
{
    height: 28px;
}
/*正文部分*/
#operate, #data, #pages
{
    width: 910px;
}
#operate
{
    height: 30px;
}
#data
{
    height: 340px;
}

/*页头的超级链接*/
#header a
{
    margin-right: 40px;
    line-height: 61px;
    color: #fff;
    text-decoration: none;
}
#header a:hover
{
    font-weight: bold;
    text-decoration: underline;
}
/* 页脚文本 */
#footer p
{
    margin: 0px auto;
    color: #fff;
    line-height: 25px;
    text-align: center;
}

/* 数据表格部分 */
#datalist
{
    width: 910px;
    margin: 5px auto;
    text-align: center;
    background-color: #fff;
    border: 0;
    border-collapse: collapse;
}
#datalist td
{
    height: 32px;
    vertical-align: middle;
    border: 1px solid #CCC;
}
#datalist tr:hover
{
    background-color: #f7f9fd;
```

```
}

#datalist tr.header td
{
    font-weight: bold;
    height: 35px;
    background: #fbedce;
}

/* 按钮样式 */
input.btn_modify, input.btn_delete
{
    margin-left: 3px;
    width: 58px;
    cursor: pointer;
    color: #000;
    border: 0;
    background-color: #fff;
    background-repeat: no-repeat;
    background-position: left center;
}
input.btn_modify
{
    background-image: url(images/modification.png);
}
input.btn_delete
{
    background-image: url(images/delete.png);
}
/* 页头 logo 部分 */
#header img
{
    float: left;
}

/* 导航菜单栏 */
#menu
{
    margin: 3px auto;
    width: 960px;
    padding: 5px;
    list-style-type: none;
}
#menu li
{
    float: left;
    margin: 0 14px;
}
#menu a
{
    display: block;
    width: 68px;
    height: 77px;
}
.index_off
{
    background: url(images/index_out.png) no-repeat;
}
.role_off
{
    background: url(images/role_out.png) no-repeat;
}
.admin_off
{
    background: url(images/admin_out.png) no-repeat;
}
.fee_off
{
```

```
background: url(images/fee_out.png) no-repeat;
}
.account_off
{
    background: url(images/account_out.png) no-repeat;
}
.service off
{
    background: url(images/service_out.png) no-repeat;
}
.bill_off
{
    background: url(images/bill_out.png) no-repeat;
}
.report off
{
    background: url(images/report_out.png) no-repeat;
}
.information off
{
    background: url(images/information_out.png) no-repeat;
}
.password_off
{
    background: url(images/password_out.png) no-repeat;
}
/* 修改操作信息提示框的父元素的定位 */
#main
{
    position: relative;
}
/* 操作信息提示框 */
#operate_result_info
{
    width: 400px;
    height: 70px;
    line-height: 70px;
    padding-left: 40px;
    font-size: 16px;
    position: absolute;
    z-index: 100;
    top: 90px;
    left: 30%;
    background-color: #fdececc;
    border: 1px solid #f57676;
    background-repeat: no-repeat;
    background-position: 10px center;
}
#operate_result_info img
{
    float: right;
    margin-right: 10px;
    margin-top: 10px;
    cursor: pointer;
}
div.operate_success
{
    color: #af6606;
    background-image: url(images/ok.png);
}
div.operate_fail
{
    color: #f00;
    background-image: url(images/warning.png);
}
```

## 课后作业

1. 完成如下图所示的页面



图 - 1

2. 简要描述 CSS 中的定位机制
3. display 属性的作用是什么？
4. 完成如下图所示的页面（提高题，选做）

有增加管理员的页面 admin\_add.html，该页面的效果如图 - 2 所示。



图 - 2

其中，信息提示文本框（“至少选择一个”）的宽度为 190px，角色的包含框的高度为 87px，宽度为 360px。