



普通高等教育电气工程与自动化(应用型)“十二五”规划教材

Fundamentals of Computer
Software Technology

计算机 软件技术基础

◎ 杨建军 主编



机械工业出版社
CHINA MACHINE PRESS

普通高等教育电气工程与自动化（应用型）“十二五”规划教材

计算机软件技术基础

主 编 杨建军

参 编 曹射香 鲁 可 林海霞

白 浩 周德祥 陈 星 赵 玮

主 审 王 莉



机械工业出版社

本书介绍了计算机软件的相关技术及 Microsoft 公司推出的程序开发工具——Visual Studio 2008 中文版的使用技巧。全书共分 9 章,第 1~8 章分别介绍 Web 程序设计基础、C#语言基础、面向对象程序设计、算法与数据结构、多媒体与网络程序设计、软件的开发流程、操作系统、关系数据库系统等内容,第 9 章是项目与实践,以“创业投资网站”的开发为例,介绍一个 Web 项目的完整开发过程。

本书结构清晰,图文并茂,将面向对象的理论与 C#和 Visual Studio 2008 紧密结合,增加 Web 程序设计理论和技术,注重软件开发能力的训练。本书可作为大中专院校、职业院校及各类社会培训学校的教材,也可作为广大计算机用户的自学参考用书。

图书在版编目 (CIP) 数据

计算机软件技术基础/杨建军主编. —北京:机械工业出版社, 2011. 8
普通高等教育电气工程与自动化(应用型)“十二五”规划教材
ISBN 978-7-111-35294-5

I. ①计… II. ①杨… III. ①软件—高等学校—教材 IV. ①TP31

中国版本图书馆 CIP 数据核字 (2011) 第 152681 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)
策划编辑:于苏华 责任编辑:于苏华 马超 徐凡
版式设计:张世琴 责任校对:闫玥红
封面设计:张静 责任印制:乔宇
北京瑞德印刷有限公司印刷(三河市胜利装订厂装订)
2011 年 10 月第 1 版第 1 次印刷
184mm×260mm·19.5 印张·479 千字
标准书号:ISBN 978-7-111-35294-5
定价:38.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010) 88361066 门户网:<http://www.cmpbook.com>

销售一部:(010) 68326294 教材网:<http://www.cmpedu.com>

销售二部:(010) 88379649

读者购书热线:(010) 88379203 封面无防伪标均为盗版

普通高等教育电气工程与自动化（应用型）“十二五”规划教材

编审委员会委员名单

主任委员：刘国荣

副主任委员：

张德江 梁景凯 张 元 袁德成 焦 斌

吕 进 胡国文 刘启中 汤天浩 黄家善

钱 平 王保家

委 员（按姓氏笔画排序）：

丁元明 马修水 王再英 王 军 叶树江

孙晓云 朱一纶 张立臣 李先允 李秀娟

李海富 杨 宁 陈志新 周渊深 尚丽萍

罗文广 罗印升 罗 兵 范立南 娄国焕

赵巧娥 项新建 徐建英 郭 伟 高 亮

韩成浩 蔡子亮 樊立萍 穆向阳

前 言

随着计算机技术的发展,尤其是 Internet 在各个行业的广泛应用,计算机软件技术的基本理论与应用及计算机软件的开发方法已成为很多专业的学生必须掌握的基本知识。Visual Studio 2008 是 Microsoft 公司最新推出的集成软件开发工具,可广泛应用于各类应用程序的开发。

本书全面介绍计算机软件技术知识,力图反映当前软件技术发展的最新成果。本书语言精练,内容简洁,知识具有系统性、实用性和先进性,适合高等学校非计算机专业学生使用。

当代程序设计的基本思想是:面向对象,模块化,封装,抽象化与测试。本书将良好的编程思想同广泛应用的 C#一同介绍,着重介绍良好的编程风格和用 C#开发各类应用程序的方法。

本书既注重理论知识的介绍,又注重学生实际应用能力的培养与训练。

本书的主要特点如下:软件理论与实际应用相结合;采用最新的软件开发技术;将面向对象的理论与 C#和 Visual Studio 2008 紧密结合;增加 Web 程序设计理论和技术;通过每章的上机练习,加强学生软件开发能力的训练。

本书从教学实际需求出发,合理安排知识结构,从零开始、由浅入深、循序渐近地介绍计算机软件技术的基本理论与应用。本书共分为 9 章,主要内容如下:

第 1 章介绍 Web 程序设计基础,主要包括 HTML、XML 和 XHTML、CSS 基础、网页的布局方法、ASP.NET 3.5 等内容。

第 2 章介绍 C#基础,主要包括 C#基础知识、C#程序的流程控制、Visual Studio 2008 安装与使用、构建 C#应用程序等内容。

第 3 章介绍面向对象程序设计,主要包括面向对象的基本思想、类与对象、类的方法与重载、继承与多态、抽象类与接口、委托与事件等内容。

第 4 章介绍算法与数据结构,主要包括算法及其表示、常用算法的结构分析、数据结构、查找和排序、文件与文件操作等内容。

第 5 章介绍多媒体与网络编程,主要包括多媒体处理技术、多媒体编程、网络协议与编程等内容。

第 6 章介绍软件的开发流程,主要包括软件的需求定义、软件系统设计、软件系统实现、软件的测试、软件维护等内容。

第 7 章介绍操作系统,主要包括操作系统的基础知识与结构、操作系统的资源管理等内容。

第 8 章介绍关系数据库系统,主要包括数据模型、SQL、ADO.NET 3.5 数据库访问技术等内容。

第 9 章介绍项目与实践,主要包括软件的生存周期,以及创业投资网站系统的需求

分析、设计、实现及运行测试等内容。

本书在介绍每个知识点时都配有相应的实例，方便读者上机实践。同时，在难以理解和掌握的内容上给出相关提示，让读者能够快速提高操作技能。此外，本书配有大量综合实例和练习，让读者在不断的实际操作中更加牢固地掌握书中内容。

本书第1、9章由河南工业大学杨建军编写，第2章由林海霞编写，第3章由曹射香编写，第4章由鲁可编写，第5章由赵祎编写，第6章由周德祥编写，第7章由白浩编写，第8章由陈星编写，由杨建军进行统稿。王莉教授认真审阅了书稿。

由于作者水平有限，加之时间仓促，书中不足之处在所难免，欢迎广大读者批评指正。

编 者

2011年5月

目 录

前言

第 1 章 Web 程序设计基础	1	2.3.2 循环程序设计	67
1.1 Web 程序设计语言	1	2.3.3 跳转语句	69
1.2 XHTML 的语法规则	2	2.4 上机练习	70
1.3 XHTML 标记及其属性	5	2.5 习题	71
1.4 CSS 基础	11	第 3 章 面向对象程序设计	72
1.4.1 CSS 简介	12	3.1 面向对象的基本思想	72
1.4.2 CSS 属性简介	14	3.2 类与对象	73
1.5 网页的布局方法	18	3.2.1 类的定义	73
1.5.1 早期常用的网页布局方法	18	3.2.2 类的成员	74
1.5.2 新的 Web 标准	19	3.2.3 创建对象与构造函数	75
1.5.3 DIV + CSS 网页布局方案	19	3.3 类的方法与重载	76
1.6 ASP.NET 概述	23	3.3.1 方法的定义与调用	76
1.6.1 ASP.NET 发展概述	23	3.3.2 方法的参数类型	77
1.6.2 ASP.NET 工作原理	24	3.3.3 静态方法和实例方法	79
1.6.3 ASP.NET 中的文件类型	24	3.3.4 方法重载	80
1.6.4 ASP.NET 应用程序的组成	25	3.3.5 this 关键字	81
1.6.5 ASP.NET 的常用内置对象	26	3.4 继承与多态	82
1.7 上机练习	32	3.4.1 继承	82
1.8 习题	35	3.4.2 多态	84
第 2 章 C#语言基础	36	3.4.3 抽象与密封	88
2.1 C#语言概述	36	3.5 接口与属性	90
2.1.1 .NET Framework 简介	36	3.5.1 接口	90
2.1.2 C#语言简介	39	3.5.2 属性	93
2.1.3 Visual Studio 2008 IDE 介绍	39	3.6 委派与事件	97
2.1.4 命名空间	45	3.6.1 委派与匿名方法	97
2.1.5 控制台应用程序的设计	47	3.6.2 事件	99
2.1.6 Windows 窗体应用程序的设计	49	3.7 上机练习	102
2.2 C#语言基础	51	3.8 习题	105
2.2.1 基本编码规则	51	第 4 章 算法与数据结构	106
2.2.2 数据类型	53	4.1 算法	106
2.2.3 System.String 类	56	4.1.1 算法的概念	106
2.2.4 System.Console 类	58	4.1.2 算法的表示	108
2.2.5 System.Math 类	59	4.1.3 常用算法的结构分析	108
2.2.6 数据类型的转换	60	4.2 数据结构	111
2.3 C#程序的流程控制	61	4.2.1 数据的结构关系	111
2.3.1 分支程序设计	61	4.2.2 数据结构的研究方法	112
		4.2.3 线性表	114

4.2.4 树和二叉树	118	6.5.3 调试	203
4.2.5 图	124	6.6 软件维护	205
4.3 查找和排序	128	6.6.1 软件维护的定义和特点	205
4.3.1 查找	128	6.6.2 软件维护的实施	205
4.3.2 排序	131	6.6.3 软件的可维护性	206
4.4 文件与文件的操作	133	6.7 上机练习	208
4.4.1 文件	133	6.8 习题	208
4.4.2 文件的操作	135	第7章 操作系统	209
4.5 上机练习	135	7.1 操作系统的基础知识	209
4.6 习题	138	7.1.1 操作系统的基本概念	209
第5章 多媒体与网络编程	140	7.1.2 操作系统的功能	211
5.1 多媒体处理技术	140	7.1.3 操作系统的分类	213
5.1.1 音频信号处理的基本知识	140	7.2 操作系统的结构	214
5.1.2 图形图像处理的基本知识	141	7.3 操作系统的资源管理	216
5.2 多媒体开发过程	142	7.3.1 进程管理	216
5.3 多媒体编程	145	7.3.2 内存管理	218
5.4 网络协议及网络操作系统	149	7.3.3 设备管理	223
5.4.1 OSI 参考模型	149	7.3.4 文件管理	226
5.4.2 TCP/IP	150	7.4 常用操作系统介绍	230
5.4.3 网络操作系统——Windows	154	7.5 上机练习	231
5.5 上机练习	157	7.6 习题	232
5.5.1 IPAddress 和 IPEndPoint 类	158	第8章 关系数据库系统	233
5.5.2 套接字 (Socket)	158	8.1 数据管理技术的产生与发展	233
5.5.3 网络流 (NetworkStream)	160	8.1.1 人工管理阶段	233
5.5.4 面向连接的套接字	160	8.1.2 文件系统阶段	234
5.6 习题	166	8.1.3 数据库系统阶段	234
第6章 软件的开发流程	167	8.2 数据库系统概述	235
6.1 软件工程过程和软件的生存周期	167	8.2.1 与数据库有关的基本概念	235
6.2 软件的需求定义	168	8.2.2 数据库系统的特点	236
6.2.1 系统可行性分析	168	8.3 数据模型	237
6.2.2 软件开发计划	170	8.3.1 数据模型的组成要素	237
6.2.3 软件的需求分析	171	8.3.2 概念模型	238
6.3 软件系统设计	183	8.3.3 常用数据模型	239
6.3.1 软件概要设计概述	183	8.3.4 层次模型	239
6.3.2 软件设计的基本原则	185	8.3.5 网状模型	239
6.3.3 软件详细设计	188	8.3.6 关系模型	240
6.3.4 软件详细设计表示方法	189	8.4 关系术语及关系运算	240
6.4 软件的系统实现	191	8.4.1 关系术语	240
6.4.1 程序设计语言的特征和选择	191	8.4.2 关系的特点	241
6.4.2 程序设计风格	192	8.4.3 实际关系模型	241
6.5 软件测试	193	8.4.4 关系运算	241
6.5.1 软件测试技术分类	194	8.5 SQL	242
6.5.2 测试用例	195	8.5.1 SQL 概述	243

8.5.2 数据定义	243	8.8 上机练习	270
8.5.3 数据查询	244	8.9 习题	282
8.5.4 数据更新	246	第9章 项目与实践	283
8.5.5 事务处理	246	9.1 软件的生存周期	283
8.5.6 触发器	247	9.2 创业投资网站的功能分析	283
8.5.7 存储过程	248	9.2.1 选题的背景和意义	283
8.5.8 视图	248	9.2.2 系统可行性分析	284
8.6 ADO.NET 数据库访问技术	249	9.2.3 系统功能概述	284
8.6.1 .NET Framework 数据提供程序	250	9.3 创业投资网站系统的设计	284
8.6.2 Connection 对象	251	9.3.1 系统功能设计	285
8.6.3 Command 对象	255	9.3.2 系统数据库设计	285
8.6.4 DataAdapter 对象	258	9.4 创业投资网站系统的实现	287
8.6.5 DataReader 对象	259	9.4.1 连接数据库	288
8.6.6 DataSet 对象	259	9.4.2 用户未登录页面的设计与实现	288
8.7 创建数据库	263	9.4.3 注册用户功能的设计与实现	294
8.7.1 使用命令行方式创建数据库	264	9.4.4 管理员相关页面的设计与实现 ...	297
8.7.2 使用 Visual Studio 2008 创建 数据库	264	9.5 系统的运行测试	299
8.7.3 使用 DataReader 完成数据库 访问	266	9.5.1 系统测试	299
8.7.4 修改数据库	269	9.5.2 测试分析报告	300
		参考文献	301

第 1 章 Web 程序设计基础

学习目标

本章将介绍 Web 程序设计的相关内容，包括 Web 程序设计语言、XHTML 的语法规则、XHTML 标记及其属性、CSS 基础、网页的布局方法和 ASP.NET 概述等，这些都是 Web 程序设计的基础。通过本章的学习，读者应该理解 HTML、XML 和 XHTML 的概念，以及三者之间的关系；掌握 XHTML 的语法规则；熟悉并使用 XHTML 标记；掌握 CSS + DIV 布局的思想；掌握 ASP.NET 的有关内容。

本章重点

- Web 程序设计语言
- XHTML 的语法规则
- XHTML 标记及其属性
- CSS 基础
- 网页的布局方法
- ASP.NET 概述

1.1 Web 程序设计语言

Web 程序设计语言包括 HTML、XML 和 XHTML。它们都是用浏览器可以执行的程序设计语言，可以使用记事本来编辑的文本文件。

XML 能够让用户自定义文件结构和标记，同时可以让计算机软件“看懂”文件内容。ASP.NET 技术的很多设置文件都是采用 XML 文件格式，如 web.config、web.sitemap 文件等。

下面分别介绍 HTML、XML 和 XHTML。

1. HTML

HTML (HyperText Markup Language, 超文本标记语言) 是制作静态页面的主要编程语言。HTML 文件是一种纯文本文件，通常以 .htm 或 .html 作为文件扩展名。在用浏览器打开某网页时，通过选择相应的“源文件”命令，可查看该网页的 HTML 源代码。

可以用各种类型的工具来创建或者处理 HTML 页面，如记事本、FrontPage、Dreamweaver、Visual Studio 2005 或 2008、Visual Web Developer 2008 等。

HTML 的语法有 3 种表达方式，具体格式如下：

- <元素名>内容</元素名>
- <元素名 属性名 1 = 属性值 1 属性名 2 = 属性值 2...>内容</元素名>
- <元素名>

一个典型的元素由 3 部分组成：一个开始标记、内容、一个结束标记。

HTML 的语法表达方式中的 `<元素名>` 是该元素的开始标记, `</元素名>` 是该元素的结束标记。例如, `<html>` 是 HTML 元素的开始标记, `</html>` 是该 HTML 元素的结束标记。

无论在何种操作系统下,只要有浏览器就可以运行 HTML 页面文档。

HTML 只是建议 Web 浏览器应该如何显示和排列信息,并不能精确定义格式,因此在不同的浏览器中显示的 HTML 文件效果可能会不同。

HTML 的主要缺点如下:

- HTML 的标记是固定的。HTML 不允许用户创建自己的标记,因此 HTML 很难做更复杂的事情(如它无法描述矢量图形、科技符号和一些其他的特殊显示效果)。
- HTML 中标记的作用只是建议浏览器用何种方式显示数据。HTML 无法解释数据之间的关系,以及相关结构方面的信息,因此不能适应日益增多的信息检索要求和存档要求。

2. XML

XML (eXtensible Markup Language, 可扩展标记语言) 可以将网络上的文档规范化,并赋予标记一定的含义。XML 1.0 是在 1998 年 2 月正式推出的,目前 XML 的相关技术仍在持续发展和制定中。

XML 的目的并不是编排内容,而是用来描述数据,它并没有像 HTML 那样的默认标记,事实上,用户需要自定义描述数据所需的各种标记。

XML 已经在文件配置、数据存储、基于 Web 的 B2B 交易、存储矢量图形和描述分子结构等众多方面得到了广泛的应用。但由于目前的浏览器对 XML 的支持还不够完善,XML 在互联网上完全替代 HTML 还需要很长一段时间。因此,掌握 XHTML 就显得尤为重要。

3. XHTML

XHTML (eXtensible HyperText Markup Language, 可扩展超文本标记语言) 是为了使 HTML 向 XML 顺利过渡而定义的标记语言,它以 HTML 为基础,采用 XML 严谨的语法结构。

XHTML 结合了部分 XML 的强大功能及大多数 HTML 的简单特性,是一种增强的 HTML,它的可扩展性和灵活性将适应未来网络应用的需求。

目前在网站设计中流行的 Web 标准就是基于 XHTML 的应用(即通常所说的 CSS + DIV)。大部分的浏览器都可以正确地解析 XHTML,即使老版本的浏览器,也将 XHTML 作为 HTML 的一个子集。

1.2 XHTML 的语法规则

1. 使用 UTF-8 之外的编码,文档必须具有 XML 声明

当文档的字符编码不是默认的 UTF-8 时,编程人员必须在 XHTML 页面中添加一个 XML 声明并指定字符编码。

例如,

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

2. HTML 标记之前必须使用 DOCTYPE 声明

文档类型声明 DOCTYPE 指定本文档遵从的 DTD (Document Type Definition, 文档类型定义) 标准,同时指定了文档中的 XHTML 版本,以及可以和哪些验证工具一起使用等信

息，以保证此文档与 Web 标准一致。

文档类型声明是每个网页文档必需的，XHTML 1.0 提供了 3 种 DTD 声明供选择，DOCTYPE 声明必须引用以下一种类型。

- Transitional（过渡型）：可以使用符合 HTML 4.0 标准的标记，但是必须符合 XHTML 的语法。Transitional 是 ASP.NET 默认文档类型定义。
- Strict（严格型）：XHTML 1.0 Strict 与 XHTML 1.0 Transitional 的不同之处在于，它在文档结构和表示形式之间实施了一种更为明显的分离。与 XHTML 1.0 Transitional 不同，XHTML 1.0 Strict 强迫使用层叠样式表（CSS）来控制页的外观。
- Frameset（框架型）：XHTML 1.0 Frameset 文档意在成为使用 <frameset> 标记将浏览器划分为多个框架的文档（注意：XHTML 1.0 Transitional 和 Strict 页不能包含 <frameset> 标记）。

例如，

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

- DOCTYPE 是 document type（文档类型）的缩写。
- “-//W3C//DTD XHTML 1.0 Transitional”说明此文档符合 W3C 制定的 XHTML 1.0 规范，即声明此文档应该按照 XML 文档规范来配对所有标记。
- “xhtml1-transitional.dtd”中的 DTD 是文档类型定义，包含了文档的规则，浏览器根据页面所定义的 DTD 来解释页面内的标识，并将其显示出来。

3. 页面的 html 标记必须指定命名空间

html 标记必须指定 XHTML 命名空间，也就是将 namespace 属性添加到 html 标记中。

例如，

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

- 其中的 xmlns 是 XHTML namespace 的缩写，即 XHTML 命名空间，用来声明网页内所用到的标记是属于哪个名称空间的。
- 本例中，指定 HTML 的标记名称空间为 http://www.w3.org/1999/xhtml，这属于 XML 1.0 的写法，说明整个网页标记应符合 XHTML 规范。

4. 文档必须包含完整的结构标记

文档必须包含 head、title 和 body 结构标记；框架集文档必须包含 head、title 和 frameset 结构标记。

例如，一个最基本的 XHTML 网页结构由 3 个部分构成。

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>标题内容</title>
  </head>
  <body>
    主要内容
  </body>
</html>
```

其中,

- `<html>...</html>`: 整个 HTML 文件的起止标记。
- `<head>...</head>`: HTML 头部文件。头部文件中包含页面传递给浏览器的信息, 这些信息作为一个单独的部分, 不是网页的主体内容; 在头部文件中可以设置页面的标题、关键字、外部链接和脚本语言等内容。例如, 用 `<title>...</title>` 标记来设置网页的标题, 用 `<script>...</script>` 标记来插入脚本等。
- `<body>...</body>`: 文档内容部分。`<body>...</body>` 标记之间为页面文档的主体, 用来放置页面的内容, 是在浏览器中需要显示的内容; 对一个最简单的网页来说, `<body>...</body>` 标记是必须使用的。

5. 标记必须成对使用

若是单独不成对的标记, 需在标记最后加“/”结束。例如, `<br />`、`<hr />` 是正确的。

6. 标记必须正确嵌套

XHTML 要求有严谨的结构, 文档中的所有标记必须按顺序正确嵌套, 不得交叉。

7. 所有标记名称和属性的名称都必须使用小写

与 HTML 不同, XHTML 对大小写是敏感的, XHTML 要求所有的标记和属性的名称都必须使用小写。例如, `<title>` 和 `<TITLE>` 在 XHTML 是不同的标记。

8. 属性值必须用引号括起来

在 HTML 中, 不要求给属性值加引号, 但是在 XHTML 中, 属性值必须被加引号 (双引号或单引号都可以)。例如, `<height = 80>` 必须修改为 `<height = " 80" >`。

在特殊情况下, 若用户需要在属性值里使用双引号, 可以使用“'”表示, 例如, `<alt = " say' hello'" >`。

9. 使用 id 替代 name 属性

10. 属性不允许简写, 每个属性必须赋值

XHTML 规定所有属性都必须有一个值, 没有值的就重复本身。例如,

```
<input id = "Checkbox1" type = "checkbox" value = "篮球" checked />
```

必须修改为

```
<input id = "Checkbox1" type = "checkbox" value = "篮球" checked =  
"checked" />
```

11. 图片必须有说明文字

每个图片标记必须有 alt 说明文字, 即必须对 img 和 area 标记应用文字说明, 格式为 `alt = "说明" 属性`。例如, `<img src = " fish. jpg" alt = " 美丽鱼" />`。

12. 不要在注释内容中使用 “--”

“--”只能发生在 XHTML 注释的开头和结束, 也就是说, 在内容中它们不再有效。例如, 下面的代码是无效的。

```
<!-- 这里是注释 ----- 这里是注释 -->
```

可以用等号或者空格替换内部的虚线, 例如,

```
<!-- 这里是注释 ==== 这里是注释 --> 是正确的。
```

1.3 XHTML 标记及其属性

XHTML 标记有很多, 标记 (Tags) 是指定界符 (一对尖括号) 和由定界符括起来的文本, 用来控制数据在网页中的编排方式, 告诉应用程序 (如浏览器) 以何种格式表现标记之间的文字。当需要对网页某处内容的格式进行编排时, 只要把相应的标记放置在该内容之前, 浏览器就会以标记定义的方式显示网页的内容。标记控制文字显示的语法为

<标记名称>需进行格式控制的文字</标记名称>

在 XHTML 标记中, 可以通过设定一些属性, 来描述标记的外观和行为方式以及内在表现, 以便对文字编排进行更细微的控制。几乎所有的标记都有自己的属性。例如,

```
style="text-align:center"
```

其中, style 就是标记的属性, style 的值设置文本格式为居中对齐。

使用标记时有如下一些注意事项。

- 任何标记都用“<”和“>”括起来, 一般情况下, 标记是成对出现的。
- 标记名与“<”之间不能有空格。
- 某些标记要加上属性, 而属性只能加于起始标记中。格式为

<标记名 属性名=属性值 属性名=属性值...>网页内容</标记名>

下面介绍常用的标记及属性。

1. 主体标记 <body>...</body>

在 XHTML 文档中, 使用 body 元素的标记可以设置网页页面背景与文本的颜色。具体来说, 在 body 元素的开始标记 <body> 中, 使用某些关系到页面全局的属性, 可以控制页面的背景与文本颜色。

<body> 标记中不带任何属性时, 表示全使用默认的属性值。网页默认的显示格式为白色背景和 12 像素黑色 Times New Roman 字体。

在 XHTML 中, <body> 标记用属性 style 来设置样式, 如设置字体的大小、颜色、页面的背景色和背景图等。格式为

<标记 style="样式1: 值1; 样式2: 值2; ...">

其中, 样式与值用冒号分隔, 如果 style 属性中包含多个样式, 各个样式之间用分号隔开。style 属性常用的样式如下。

- background-color: 设置网页的背景颜色, 默认为白色背景。
- color: 设置网页中字体的颜色, 默认颜色为黑色。
- font-family: 设置网页中字体的名称, 如宋体、楷体、黑体等。
- font-size: 设置网页中字体的大小。
- text-align: 设置网页中文本的对齐方式。常用有 3 种不同的取值, 分别为 left (左对齐, 默认对齐方式)、right (右对齐)、center (居中对齐)。

例如, 设置网页字体为宋体, 字体的颜色为蓝色, 方法如下。

```
<body style="font-family:宋体;color:blue">
```

```
...
```

```
</body>
```

2. 分层标记 <div>...</div>

分层标记为 <div>...</div>。其主要作用如下。

- 分层标记用来排版大块的 XHTML 段落，为 XHTML 页面内大块（block-level）的内容提供结构和背景的标志。
- 可用 style 属性，在其中加入许多其他样式，以实现对其中包含元素的版面设置。
- div 标记除了可以作为文本编辑功能外，还可以用做容器标记，将按钮、图片、文本框等各种标记放在 div 里面作为它的子对象元素处理。

3. 文本格式化设置标记

文本格式化设置标记包括标题字体大小设置、字体风格设置、段落设置等内容。

(1) 标题字体大小设置标记 <h1>...</h1>

标题字体大小分为 6 级。由大至小，有 6 种设置标题字体格式的标记：<h1>、<h2>、<h3>、<h4>、<h5> 和 <h6>。标题字体的基本标记格式为

<h1 style = "...">标题内容</h1>

其中，n=1, 2, 3, 4, 5, 6，即有 h1、h2、h3、h4、h5 和 h6 六级。

例如，

<h1 style = " color:Red; text-align:right">h1 标题</h1>

<h2 style = "text-align:center">h2 标题</h2>

(2) 设置字体风格的标记

为了使文本突出，以引起浏览者的注意，对文本设置适当的字体风格是一个有效的途径。在 XHTML 中，提供有多种字体风格标记供选用。

设置字体风格的标记主要有如下几种。

- ... 标记：以加粗字的形式输出文本。
- <i>...</i> 标记：以斜体字的形式输出文本。
- <big>...</big> 标记：以较大字的形式输出文本。
- <small>...</small> 标记：以较小字的形式输出文本。

例如，

<div>

设置为粗体字

<i>设置为斜体字</i>

^{上标字}

_{下标字}

<big>设置为较大字</big>

<small>设置为较小字</small>

</div>

(3) 换行标记

用于添加一个 <Enter> 键换行，该标记没有结束标记，故在 XHTML 中以 </> 结束。在编写 XHTML 时，如果在文件中用 <Enter> 键分开了某一段文字，当在浏览器中显示时，浏览器会忽略源代码中的换行，而并不会显示换行的效果。若要显示网页中的文字换行效果，必须在文件中使用
 标记。

(4) 段落标记

- 段落标记为 `<p>...</p>`

段落标记 `<p>...</p>` 的作用是将标记之间的文本内容自动组成一个完整的段落。

- 预格式化标记 `<pre>...</pre>`

预格式化标记 `<pre>...</pre>` 使标记之间的文本信息能够在浏览器中按照原格式毫无变化地输出。它可以使浏览器中显示的内容与代码中输入的文本信息格式完全一样。

(5) 画线标记 `<hr/>`

画线标记 `<hr/>` 单独使用, 可以实现段落的换行, 并绘制一条水平直线, 并在直线的上下两端留出一定的空间。可以使用 `style` 属性进行设置。其中:

- `width`: 设置画线的长度, 取值可以是以像素为单位的具体数值, 也可以使用相对于其父标记宽度的百分比数值。
- `height`: 设置画线的粗细, 单位是像素。

(6) 空格标记 ` `

在 XHTML 中, 直接输入多个空格, 仅仅会被视为一个空格, 而多个回车换行符也仅仅被浏览器解读为一个空格。

为了能够显示多个空格, XHTML 保留了 HTML 中的空格标记 ` `。一个 ` ` 代表一个空格; 多个 ` ` 则代表相应的空格数。

(7) 文本居中标记 `<center>...</center>`

文本居中标记 `<center>...</center>` 用来将网页中 `center` 标记内的元素居中显示。

(8) 列表标记

列表标记包括无序列表标记 `<ul>...</ul>` 和有序列表标记 `<ol>...</ol>`。

1) 无序列表标记为 `<ul>...</ul>`。

无序列表标记为 `<ul>...</ul>`, 列表项标记为 `<li>...</li>`。语法格式为

```
<ul style="list-style-type">
<li>列表项 1
<li>列表项 2
...
<li>列表项 n
</ul>
```

`list-style-type` 可以设置为 `disc` (实心圆)、`circle` (空心圆) 和 `square` (实心方块), 默认形式为实心圆●。

`<li>` 有自动换行的作用, 每个条目自动为一行。

2) 有序列表标记 `<ol>...</ol>`。

有序列表标记 `<ol>...</ol>` 和列表项标记 `<li>...</li>` 语法格式为

```
<ol style="list-style-type">
<li>列表项 1
<li>列表项 2
...
<li>列表项 n
```


list-style-type 可以设置为 upper-alpha（大写英文）、lower-alpha（小写英文）、upper-roman（大写罗马数字）、lower-roman（小写罗马数字）和 decimal（十进制数字）等。

(9) 注释标记 <!-- 注释内容 -->

注释标记常用在比较复杂或多人合作设计的页面中，为代码部分加上说明，方便日后修改，增加页面的可读性和可维护性。浏览器会自动忽略注释标记中的文字（可以是单行也可以是多行）而不显示。

4. 表格标记

表格标记为 <table>...</table>。表格由行与列组成，每一个基本表格单位称为单元格。单元格在表格中可以包含文本、图像、表单及其他页面元素。表格标记常用属性如下。

- align: 设置表格在网页中的水平对齐方式，可选值为 left、right、center。
- background: 为表格指定背景图片。
- bgcolor: 为表格设定背景色。
- border: 设置表格边框厚度，如果此参数为 0，那么表格不显示边界。
- cellpadding: 设置单元格中的数据与表格边线之间的间距，以像素为单位。
- cellspacing: 设置各单元格之间的间距，以像素为单位。
- valign: 设置表格在网页中的垂直对齐方式，可选值为 top、middle、bottom。
- width: 设置整个表格宽度。

行起止标记为 <tr>...</tr>。此标记表明了表格一行的开始和结束，主要有以下属性。

- align: 设置行中文本在单元格中的水平对齐方式，可选值为 left、right、center。
- background: 为这一行单元格指定背景图片。
- bgcolor: 为这一行单元格设定背景色。

单元格起始标记为 <td>...</td>。单元格起始标记用于设置表行中某个单元格的开始和结束。

5. 图像标记

图像标记语法格式为

```
<img src = "图像文件名" [border = "边框宽度"] [align = "对齐方式"] [width = "宽度像素大小"] [height = "高度像素大小"] alt = "说明文字" />
```

主要属性说明如下。

- src: 这个属性是必需的，用来链接图像的来源。
- align: 设置图像旁边文字的位置。可以控制文字出现在图片的上方、中间、底端、左侧和右侧。可选值为 top、middle、bottom、left 和 right，默认值为 bottom。
- alt: 区别于 HTML，每个图片标记必须有 ALT 说明文字。

6. 超链接标记 <a>...

超链接是通过文字、图像等载体对文件进行链接，引导文件的阅读。

超链接标记的格式为

```
<a href = "URL" id = "设置锚点" target = "链接目标网页打开的窗口">锚点 </a>
```

锚点是实现链接的源点，浏览者通过在锚标上单击鼠标就可以到达链接目标点。