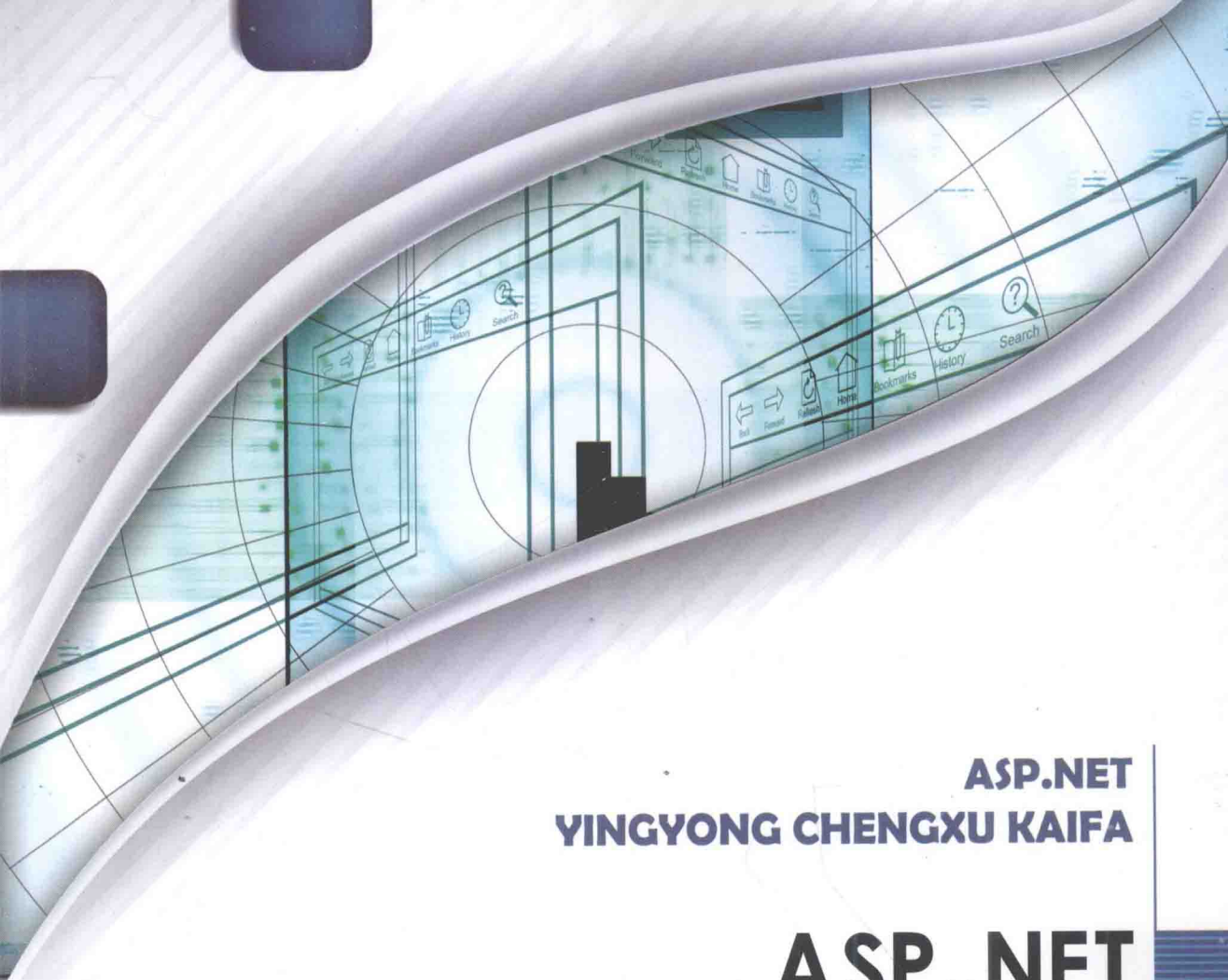




ASP.NET
YINGYONG CHENGXU KAIFA

ASP.NET 应用程序开发

库	波	余恒芳	主 编
江	骏	张克斌	副主编
		王路群	主 审

ASP.NET 应用程序开发

主 编 库 波 余恒芳
副主编 江 骏 张克斌
参 编 秦培煜 郭 俐 袁晓曦
主 审 王路群



北京理工大学出版社

BEIJING INSTITUTE OF TECHNOLOGY PRESS

ISBN 978-7-302-47000-9
定价：39.00元

未经许可，不得以任何方式复制或抄袭本书内容。

内 容 简 介

“ASP.NET 应用程序开发”课程的主要任务是培养学生使用 Visual Studio 开发平台,利用 .NET 技术来创建适合企业和公司应用的商业网站、建立维护网站的能力,使学生了解 ASP.NET 技术应用领域及与其他网站开发技术的结合,掌握运用 ASP.NET 技术开发动态网页及网站的技术,达到高素质劳动者和商务级专门人才必须具备的动态网页制作核心知识和网站建立维护的基本技能,并为就业和继续学习打下良好的基础。

本书突出以实训与理论相结合,以“实用、好用、够用”为原则,结构合理,通俗易懂,实用性强。全书共 9 章,内容包括: XHTML CSS 基础知识、C#程序设计基础、ASP.NET 基础、ASP.NET 进阶、ASP.NET 数据库编程、ASP.NET 数据服务控件、数据验证控件、ASP.NET 配置和优化、网站开发典型案例。在内容上,将 ASP.NET 应用技术与具体案例相结合,通过项目实训联系和综合实训,注重学生的编程技能的培养;在板块设计上,分阶段、分层次、循序渐进、深入浅出地培养学生的动手能力。

本书可作为高等院校学生的 ASP.NET 应用程序开发课程的教材以及 ASP.NET 培训班的教材,也可作为广大 ASP.NET 爱好者的自学参考书。

版权专有 侵权必究

图书在版编目(CIP)数据

ASP.NET 应用程序开发/库波,余恒芳主编. —北京:北京理工大学出版社, 2013. 5

ISBN 978-7-5640-7700-6

I. ①A… II. ①库…②余… III. ①网页制作工具—程序设计 IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2013)第 106699 号

出版发行 / 北京理工大学出版社

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010)68914775 (办公室) 68944990 (批销中心) 68911084 (读者服务部)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 三河市天利华印刷装订有限公司

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 14

字 数 / 327 千字

版 次 / 2013 年 5 月第 1 版 2013 年 5 月第 1 次印刷

印 数 / 1 ~ 1 500 册

定 价 / 39.00 元

责任编辑 / 王玲玲

责任校对 / 周瑞红

责任印制 / 王美丽

图书出现印装质量问题,本社负责调换

ASP.NET 应用程序开发是微软公司 2000 年推出的一种 Internet 编程技术,是 .NET 框架的组成部分,编程效率很高,采用面向对象的方法来创建动态 WEB 应用程序。

本书的目标是将它作为大学生学习 ASP.NET 的入门教程,而且设计成适合计算机相关各个专业的学生学习。学习本课程之前必须有一定的计算机基础、C#程序设计基础和数据库原理等课程基础,并且有一定的 Windows 使用基础。

本书突出以知识点“实用、好用、够用”为原则,结构合理、通俗易懂、实用性强,以任务驱动贯穿全书,再辅以实践任务,让学生在实践中掌握 ASP.NET 编程的开发方法。在内容上,将 ASP.NET 应用技术与具体案例相结合,通过项目实训联系和综合实训,注重学生的编程技能的培养;在板块设计上,分阶段、分层次、循序渐进、深入浅出地培养学生的具体动手能力。

本书是作者在多年的教学实践和科学研究的基础上,阅览了大量国内外相关教材资料后,编写而成的。主要特点如下:

1. 知识点全面,语言简洁

本书由浅入深地展示了 ASP.NET 应用程序开发的特点和方法,较为系统、全面地讲解了 ASP.NET 应用程序开发的基础知识和基本方法。

结合高等院校学生的特点,在本教材中,没有大段的理论介绍,每个章节先讲解各个知识点,然后辅以实践任务,更多的是对精心设计的实例及其操作步骤的讲解,引导读者加深对各个知识点的理解和应用。

2. 内容翔实、融会贯通

本书教学重点和教学内容都是在 C#初级程序设计、C#高级程序设计和网页设计、数据库原理等前序课程的基础上建立起来的,教材中涉及了这些前序课程中的要点内容,学生可以循序渐进地学习,不断提升职业能力。

3. 结构合理、内容新颖

本书结构合理,内容重难点突出。本书按内容划分章节,每个章节中知识点和实践任务有机结合,教学内容明确,通过实践任务中的案例,将学生动手实践进行到底。

4. 提供实训资源和电子教案

本教程案例丰富,教材中的实训资源、电子教案等教学资源会在出版社网站(www.bitpress.com.cn)中提供给读者。后期会申请建立精品资源共享课,除包含教材的资源库、案例库外,还会在校级精品资源共享课中设立论坛,方便读者交流和学习。

本书分为 9 章,内容包括:XHTML CSS 基础知识、C#程序设计基础、ASP.NET 基础、ASP.NET 进阶、ASP.NET 数据库编程、ASP.NET 数据服务控件、数据验证控件、ASP.NET 配置和优化、网站

开发典型案例。每章后均附有实践任务,便于读者活学活用书中的知识,迅速熟练使用 ASP.NET 进行 WEB 编程。

本书由库波、余恒芳担任主编,江骏、张克斌担任副主编,秦培煜、郭俐、袁晓曦参加编写。全书由库波、余恒芳统稿,王路群主审。

由于时间仓促,书中不妥或错误之处敬请广大读者批评指正。同时,恳请读者一旦发现错误,于百忙之中及时与编者联系,编者将不胜感激。

编者

目 录

第1篇 基础技术

第1章 XHTML CSS 基础知识	003
本章导读	003
知识点储备	003
知识点1 XHTML 基本语法	003
知识点2 CSS 样式表	006
知识点3 CSS Sprites 技术	014
实践任务	015
本章小结	018

第2章 C#程序设计基础	019
本章导读	019
知识点储备	019
知识点1 标识符	019
知识点2 变量与常量	020
知识点3 数据类型	023
知识点4 运算符与表达式	025
知识点5 强制转换	030
知识点6 字符串	032
知识点7 流程控制语句	038
知识点8 数组	046
实践任务	047
任务1 创建一个控制台应用程序	047
任务2 创建一个 Windows 应用程序	050
本章小结	053

第3章 ASP.NET 基础	054
本章导读	054
知识点储备	054
知识点1 ASP.NET 简介	054
知识点2 配置 ASP.NET 运行环境	056
知识点3 简单控件	063

实践任务	067
任务1 第一个 ASP.NET 应用程序	067
任务2 控件综合应用	068
本章小结	072

第2篇 核心技术

第4章 ASP.NET 进阶	075
----------------------	-----

本章导读	075
知识点储备	075
知识点1 Web 对象模型	075
知识点2 ASP.NET 内置对象	080
实践任务	084
任务1 设置 Session 时效	084
任务2 网站访问计数器	085
任务3 访问 HTML 基于表单的数据	085
任务4 Cookie 的存储与读取	086
本章小结	088

第5章 ASP.NET 数据库编程	089
-------------------------	-----

本章导读	089
知识点储备	089
知识点1 ADO.NET 简介	089
知识点2 ADO.NET 组成部件	090
知识点3 使用 Connection 对象连接数据库	092
知识点4 使用 Command 对象	096
知识点5 使用 DataReader 对象读取数据库	097
知识点6 使用 DataAdapter 对象	098
知识点7 使用 DataSet 对象访问数据库	099
知识点8 LINQ 新技术	100
实践任务	110
任务1 以表格形式显示数据表中的数据	110
任务2 对数据表进行插入操作	115
任务3 分页显示数据表中的记录	117
本章小结	118

第6章 ASP.NET 数据服务控件 119

本章导读 119

知识点储备 119

知识点1 GridView 控件绑定数据 119

知识点2 DataList 控件 129

知识点3 DetailsView 控件 138

知识点4 简单服务器控件的数据绑定 144

实践任务 144

任务1 数据绑定与 GridView 控件的应用 144

任务2 DataView 控件的应用 146

本章小结 149

第7章 数据验证控件 150

本章导读 150

知识点储备 150

知识点1 初识验证控件 150

知识点2 ASP.NET 验证服务器控件 151

知识点3 使用图片或声音标识错误摘要 166

知识点4 使用验证组 168

知识点5 禁用 ASP.NET 服务器控件验证 170

实践任务 172

任务1 表单验证 172

任务2 邮件地址验证 174

本章小结 176

第8章 ASP.NET 配置和优化 177

本章导读 177

知识点储备 177

知识点1 使用 Web.config 进行配置 177

知识点2 使用 Global.asax 进行配置 178

知识点3 主题和皮肤 180

知识点4 母版页 181

实践任务 182

任务1 创建主题和皮肤 182

任务2 动态切换皮肤 183

任务3 使用模板页和内容页	185
本章小结	189

第3篇 典型应用

第9章 网站开发典型案例	193
--------------------	-----

本章导读	193
------------	-----

实践任务	193
------------	-----

任务1 文件上传	193
----------------	-----

任务2 文件加密	196
----------------	-----

任务3 留言板	200
---------------	-----

任务4 图文验证码	209
-----------------	-----

任务5 QQ留言	214
----------------	-----

参考文献	216
------------	-----

第1篇 基础技术

XHTML CSS 基础知识



本章导读

本章主要介绍 XHTML 的基本语法,然后介绍 CSS 样式表的基本概念及应用,最后介绍 CSS Sprites 技术。重点介绍 CSS 横向导航菜单、下拉及多级弹出式菜单和使用 CSS 构造块(div)。读者学习完本章后应能掌握 CSS 样式表导航技术和 div + CSS 网页标准布局。



知识点储备

知识点1 XHTML 基本语法

HTML(Hyper Text Markup Language),是超文本链接标记语言,是在互联网发布超文本文件的通用语言,是将所有的数据都嵌入到 HTML 文档中,不能与别人进行数据交换。

XML(Extensible Markup Language),能精确地描述数据,能使数据从文档中分离出来,便于数据交换。

XHTML(Extensible Hyper Text Markup Language),是一种能兼容 XML 的变体,是 HTML 的升级版,X 代表可扩展的,XHTML 将以前版本的 HTML 能够实现的一些功能交给了 CSS,可以将内容和表示分离开,大大节省了网站维护的时间和费用。

1. 文档类型

XHTML 文档包括三个部分:DOCTYPE、Head、Body,在文档中,<! DOCTYPE >是强制使用的。基本的结构是:

```
<! DOCTYPE...> <html> <head> <title>... </title> </head> <body>...  
</body> </html>
```

在 XHTML 文档中,文档类型声明总是位于首行的。

例,如下一个 XHTML 文档的实例:

```
<! DOCTYPEhtmlPUBLIC " -//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> <html> <head> <title> simpledocu-  
ment </title> </head> <body> <p> a simple paragraph </p> </body> </html>
```

例中,文档类型声明定义文档的类型是:

```
<! DOCTYPEhtmlPUBLIC " -//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

其余的部分跟 HTML 相似:

```
<html> <head> <title> simpledocument </title> </head> <body> <p>
```

```
asimpleparagraph </p> </body> </html>
```

DTD (Document Type Definition): 文档类型解析, 规定了网页使用的标准通用标记语言 (SGML, Standard Generalized Markup Language) 的语法。SGML 是很精确的被计算机读取的语言。DTD 文件是一个 ASCII 的文本文件, 后缀名为 .dtd。利用 DOCTYPE 声明中的 URL 可以访问指定类型的 DTD 详情。

XHTML 有三种文档类型, 分别是 STRICT (严格类型)、TRANSITIONAL (过渡类型) 和 FRAMESET (框架类型)。

XHTML 1.0 中规定了三种 XML 文档类型, 对应三种 DTD。

(1) XHTML 1.0 Strict

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

这种情况下, 使用简洁的标记, 避免表现上的混乱。

(2) XHTML 1.0 Transitional

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

这种情况下, 利用 HTML 表现特性时, 需要为那些不支持层叠样式表的浏览器编写 XHTML。

(3) XHTML 1.0 Frameset

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

这种情况, 利用 HTML 框架将浏览器框窗口分割成两部分或更多框架。

2. 语言编码

在 XHTML 中定义语言编码, 采用类似于这样的语句:

```
<meta http-equiv="content-type" content="text/HTML; charset=gb2312"/>
```

为了能使浏览器正确解释和通过 W3C 代码校验, 所有的 XHTML 文档都必须声明它们所使用的编码语言, 在国内一般使用 gb2312 (简体中文)。设计多国语言页面, 可以采用 unicode、iso-8859-1 等, 根据需求定义。

注: XML 与 XHTML 文档定义语言编码不同, XML 文档定义语言编码方式:

```
<? XML version="1.0" encoding="gb2312"? >
```

3. XHTML 基本语法

XHTML 基本语法:

- ① 在文档的开头必须定义文档类型;
- ② 在根元素中应声明命名空间, 即设置 xmlns 属性;
- ③ 所有标签都必须是闭合的;
- ④ 所有元素和属性都必须小写;
- ⑤ 所有的属性必须用引号“”括起来;
- ⑥ 所有标签都必须合理嵌套;
- ⑦ 所有属性都必须被赋值, 没有值的属性就用自身来赋值。

4. HTML 标签

例, 先设计一个 HTML 网页。网页源代码如下:

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

```
<head>
```

```
<title>网页标题页</title>
```

```
</head>
```

```
<body>
```

```
<p><b>这是我的第一个 HTML 网页!</b></p>
```

```
</body>
```

```
</html>
```

在这个网页源代码中,网页的标题在一对标签<title></title>中,<title>标签表示标题,</title>标符为标题结束符。标签会影响到被它包含的内容。这就是 HTML 标签的作用,被标签包含的内容就是元素(element)。

HTML 的常用标签和标签的属性如下:

<a>:超文本链接。如,创建超文本链接,其中的 URL 为链接目标地址; ,创建自动发送电子邮件的链接; ,创建位于文档内部的书签; ,创建指向位于文档内部书签的链接。

:粗体字。如湖北。

<big></big>:字体加大。如<big>武汉</big>。

<body></body>:文件本体。如<body bgcolor="">,设置背景颜色。

<body background="">:设置背景图片。如<body bgsound="">,设置背景音乐; <body bgproperties="fixed">,固定背景图片(IE 适用); <body text="">,设置文本颜色; <body link="">,设置链接颜色; <body vlink="">,设置已使用的链接的颜色; <body alink="">,设置正在被击中的链接的颜色; <body topmargin="">,设置页面上边距; <body leftmargin="">,设置页面的左边距。

:换行。

<button></button>:创建一个按钮。

<caption></caption>:表格的标题。如<caption>商品表</caption>。

<center></center>:水平居中。如<caption>商品编号</caption>。

<cite></cite>:引用,通常是斜体。如<cite>商品类型</cite>。

:强调文本(通常是斜体加黑体)。如商品名称。

:设置字体。如,设置字体大小从 1 到 7,颜色使用名字或 RGB 的十六进制值。

<form></form>:创建表单。

<frameset></frameset>:帧标签。如<frameset rows="value,value">,定义一个帧内的行数,可以使用绝对像素值或高度的百分比; <frameset cols="value,value">,定义一个帧内的列数,可以使用绝对像素值或宽度的百分比。

<h1></h1>到<h6></h6>:标题。如<h1>到<h6>字号顺序减小,<h1>文章标题</h1>,<h2>一级标题</h2>,<h3>二级标题</h3>,<h4>三级标题</h4>等。

<head></head>:文头。如<head>商品信息管理</head>。

<hr />:水平线。如<hr size="1" width="80%" color="ff0000" />。

<html></html>:创建一个 HTML 文档,分别放在 html 文件的开头和最后。如<html xmlns="http://www.w3.org/1999/xhtml"></html>。

`<label> </label>`: 为一个控件提供标签。如 `<label> 商品销售量 </label>`。

`<marquee> </marquee>`: 普通转动。如 `<marquee behavior = slide> ... </marquee>` 滑动; `<marquee behavior = scroll> ... </marquee>` 预设卷动; `<marquee behavior = alternate> ... </marquee>` 来回卷动; `<marquee direction = down> ... </marquee>` 向下卷动; `<marquee direction = up> ... </marquee>` 向上卷动; `<marquee direction = right> </marquee>` 向右卷动; `<marquee direction = 'left'> </marquee>` 向左卷动; `<marquee loop = 2> ... </marquee>` 卷动次数; `<marquee width = 180> ... </marquee>` 设定宽度; `<marquee height = 30> ... </marquee>` 设定高度; `<marquee bgcolor = FF0000> ... </marquee>` 设定背景颜色; `<marquee scrollamount = 30> ... </marquee>` 设定卷动距离; `<marquee scrolldelay = 300> ... </marquee>` 设定卷动时间。

`<p> </p>`: 段落。如 `<p> <b> 这是我的第一个 HTML 网页! </b> </p>`。

`<select> </select>`: 创建下拉菜单。如 `<select name = "name"> </select>`。

`<table> </table>`: 创建一个表格。

`<tr> </tr>`: 表格中的每一行。如 `<tr align = "">`, 设置表格单元格的水平对齐方式 (left, center, right, justify); `<tr valign = "">`, 设置表格单元格的垂直对齐方式 (baseline, bottom, middle, top)。

`<td> </td>`: 表格中一行中的每一个格子。如 `<td colspan = "">`, 设置一个表格单元格跨占的列数 (缺省值为 1); `<td rowspan = "">`, 设置一个表格单元格跨占的行数 (缺省值为 1); `<td nowrap>`, 禁止表格单元格内的内容自动断行。

`<th> </th>`: 设置表格头。如 `<th> 商品信息 </th>`。

`<title> </title>`: 文档的标题。如 `<title> 商品管理页 </title>`。

`<!-- -->`: 注释。如 `<!-- 注释内容 -->`。

知识点 2 CSS 样式表

1. CSS 优先级

(1) CSS 语法规则

CSS (Cascading Style Sheet), 层叠样式表, 也称级联样式表, 是用来进行网页风格设计的, 能够将网页表现与内容分离的一种样式设计语言。CSS 能够对网页中的对象的位置进行像素级的精确控制, 支持几乎所有的字体、字号样式, 拥有网页对象盒模型的能力, 能实现初步的交互设计。

CSS 由三个部分组成: 选择符 (selector)、属性 (properties) 和属性的取值 (value)。基本格式如下:

```
selector {property: value}
```

(选择符 { 属性: 值})

选择符有多种形式。

第一种: 定义样式的 HTML 标记, 可以定义它的属性和值。如:

```
body {color:red}
```

选择符 body 是网页主题部分, color 是控制文章颜色的属性, black 是颜色的值。

第二种: 属性的值由多个单词组成, 必须在值上加引号。如:

```
p {font - family: "arial"}
```

(定义段落字体为 arial)

第三种:需要对一个选择符指定多个属性时,使用分号把所有的属性和值分开。如:

```
p {text-align:left;color:red}
```

(段落左对齐排列;并且段落中的文字为红色)

注:为了定义样式表以方便阅读,可以采用分行的方式书写。如:

```
p
{
text-align:right;
color:yellow;
font-family:meiryo
}
```

(段落右对齐排列,段落中文字为黄色,字体是 meiryo)

选择器的三种基本类型:

- 标签名选择器,如: `p{}`,即直接使用 HTML 标签作为选择器。
- 类选择器,如: `.polaris{}`。
- ID 选择器,如 `#polaris{}`。

注意,ID 选择器跟类选择器有很大的不同:一个页面内不能出现相同的 ID;ID 是后台开发人员经常使用的,前端开发人员应该尽量少用。当然,跟后台人员的工作配合默契之后,这些都不会成为限制。

(2) CSS 选择符优先级

CSS 优先级是 CSS 样式在浏览器中被解析的先后顺序。

既然 CSS 有优先级,就会有规则来约束,基本规则是:

- 统计选择符中的 ID 属性个数。
- 统计选择符中的 CLASS 属性个数。
- 统计选择符中的 HTML 标记名个数。

按正确的顺序写出三个统计的数字,不加空格,得到一个三位数(css2.1 是用 4 位数表示)。

注意,你需要把数字转换成一个以三个数字结尾的更大的数。根据选择符的最终数字列表,可以很容易确定较高数字特性凌驾于较低数字的特性。如:

- 每个 ID 选择符(`#someid`)加 0,1,0,0。
- 每个 class 选择符(`.someclass`)、每个属性选择符(形如 `[attr = value]` 等)、每个伪类(形如: `hover` 等)加 0,0,1,0。
- 每个元素或伪元素(`:firstchild`)等加 0,0,0,1。
- 其他选择符,包括全局选择符 `*`,加 0,0,0,0。相当于没加,不过这也是一种 specificity,后面会解释。

表 1-1 所示为按照特性分类的选择符列。

表 1-1 特性分类的选择符列表

选择符	特性值
<code>h1 { color:blue; }</code>	1
<code>pem { color:purple; }</code>	2
<code>.apple { color:red; }</code>	10

续表

选择符	特性值
p. bright{ color: yellow; }	11
p. brightem. dark{ color: brown; }	22
#id316{ color: yellow }	100

从表 1-1 看出,HTML 标记的权重是 1,CLASS 的权重是 10,ID 的权重是 100,继承的权重为 0。将这些数字串逐位相加,得到最终的权重,取舍时按从左到右的顺序逐位比较。CSS 的优先级可以解决冲突,当同一标记被 CSS 选择符选中时,就要用到 CSS 优先级来取舍。

(3) CSS 样式优先级

在网页中插入 CSS,有链入外部样式表、内部样式表、导入样式表和内嵌样式表等方法。CSS 样式优先级最高的是内嵌样式,内部样式表高于导入外部样式表,链入的外部样式表和内部样式表之间是最后定义的优先级高的。

① 链入外部样式表,是将样式表保存为一个样式表文件,然后在页面中用 <link> 标记链接到这个样式表文件,<link> 标记放到页面的 <head> 里面。如:

```
<head>
...
<link href = "mystyle.css" rel = "stylesheet" type = "text/css" media = "all">
...
</head>
```

例中,是从 mystyle.css 文件中以文档格式读出定义的样式表,rel 属性是指外部的样式表,type 是指文件类型,href 是文件所在位置,media 可选择媒体类型,如纸张、屏幕、语音等。

mystyle.css 文件的内容如下:

```
hr {color: sienna}
p {margin-left:18px}
body {background-image: url("images/back40.gif")}
/* 定义水平线的颜色为土黄;段落左边的空白边距为 20 像素;页面的背景图片为 images 目录下的 back40.gif 文件*/
```

② 内部样式表,把样式表放到网页的 <head> 中,用 <style> 标记插入,如:

```
<head>
...
<style type = "text/css">
hr {color: sienna}
p {margin-left:18px}
body {background-image: url("images/back40.gif")}
</style>
...
```

注:低版本浏览器不能识别 style 标记。在低版本浏览器中,为了避免这种情况,用注释(<!-- 注释 -->)。如:

```
<head>
```