

单元 7 运用 DIV + CSS 制作网站页面

励志:每一个成功者都有一个开始。勇于开始,才能找到成功的路。

任务:熟悉网页的 CSS 样式,学会把用 ps 生成的 psd 文件切分成 html 文件。

能力锻炼:自学能力、分析问题的能力、解决问题的能力、动手能力。

关键词:CSS 样式表、DIV 层、路径、命名、切图、站点。

DIV+CSS 是网站标准(或称 Web 标准)中常用的术语之一,DIV+CSS 是一种网页的布局方法,这一种网页布局方法有别于传统的 HTML 网页设计语言中的表格(table)定位方式,真正达到了 w3c 内容与表现相分离。HTML 语言自 HTML4.01 面世以来,不再发布新版本,原因就在于 HTML 语言正变得越来越复杂化、专用化。XHTML 语言可以将 HTML 语言标准化,用 XHTML 语言重写后的 HTML 页面可以应用许多 XML 技术,使得网页更加容易扩展,适合自动数据交换,并且更加规整。在 XHTML 网站设计标准中,不再使用表格定位技术,而是采用 DIV+CSS 的方式实现各种定位。

7.1 背景知识

CSS 关键就在于其与脚本语言(如 JavaScript)及 XML 技术的融合,即 CSS+JavaScript+XML。即可扩展样式表语言相较于 CSS 过于复杂,不太容易上手。自从 CSS 出现之后,HTML 开始将页面内容与样式分离。

DIV 元素是用来为 HTML 文档内大块(block-level)内容提供结构和背景的元素。DIV 的起始标签和结束标签之间的所有内容都是用来构成这个块的,其中所包含元素的特性由 DIV 标签的属性来控制,或者是通过使用样式表格式化这个块来进行控制。

7.1.1 CSS 样式表

CSS(Cascading Style Sheet,可译为层叠样式表或级联样式表)是一组格式设置规则,用于控制 Web 页面的外观。通过使用 CSS 样式设置页面的格式,可将页面的内容与表现形式分离。页面内容存放在 HTML 文档中,而用于定义表现形式的 CSS 规则则存放在另一个文件中或 HTML 文档的某一部分,通常为文件头部分。将内容与表现形式分离,不仅可使维护站点外观更加容易,而且还可以使 HTML 文档代码更加简练,缩短浏览器的加载时间。

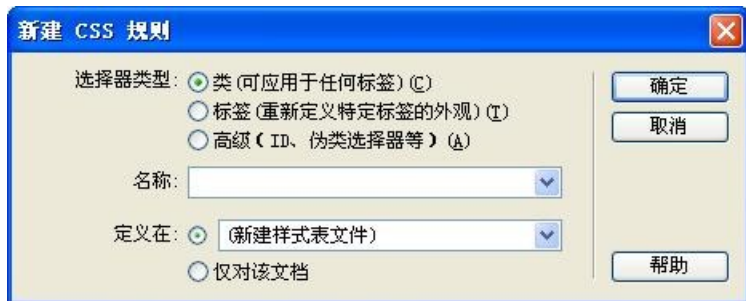


图 7-1 CSS 样式

1. CSS 样式基础 - 基本语法

CSS 的定义由三个部分构成:选择符(selector)、属性(properties)和属性的取值(value)。

语法: selector {property: value} (选择符 {属性:值})。

在以上的语法中,选择符用来关联调用样式的页面元素。每个选择符中,CSS 定义的具体内容使用“{}”包含。在每个属性的后面使用英文“:”区分属性与属性值。在每个属性值后面,使用英文的“;”分隔选择符中定义的其他属性。如果选择符中包含很多属性,则可以依次定义各种属性和属性值。

在 CSS 中包含各种选择符,每种选择符的优先级也有所差别。这些选择符包括标记选择符、类选择符、ID 选择符等。同时还可以使用子选择符进行精确定位,或者使用选择符分组的方法,同时定义几个选择符中的样式。

(1) 标记选择器。

标记必须是 HTML 中的标准标记。图 7-2 是使用标记选择符的例子。

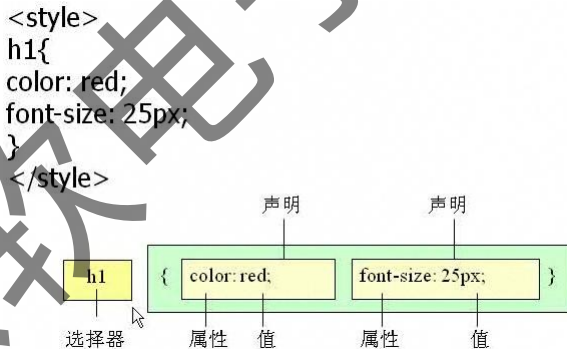


图 7-2 标记选择器

(2) 类别选择器,如图 7-3 所示。

类选择符用来定义页面中可以重复使用的 CSS 属性,其优先级低于 ID 选择符,其语法结构为: . 选择符名称。

在定义选择符的时候,要使用英文的“.”加选择符的名称。在页面元素中,使用相应的 class 属性值来调用类选择符中定义的 CSS 样式。

如: <h1 class="center">这个标题是居中排列的</h1>;

<p class="center">这个段落也是居中排列的</p>。

注意:这种省略 HTML 标记的类选择符是我们最常用的 CSS 方法,使用这种方法,我们可以很方便地在任意元素上套用预先定义好的类样式。

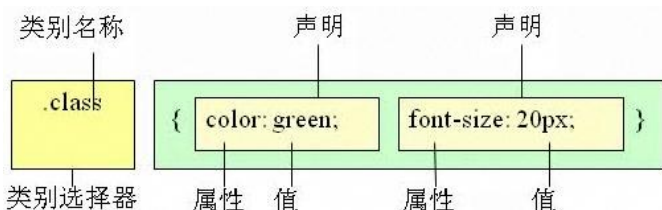


图 7-3 类别选择器

(3) ID 选择器,如图 7-4 所示。

ID 选择符是所有选择符中优先级最高的,其语法结构为“# 选择符名称”。

在定义 ID 选择符的时候,要使用英文的“#”加选择符的名称。在页面元素中,使用相应的 ID 属性值来调用 ID 选择符中定义的 CSS 样式。

```
# intro
{
font-size:110%;
font-weight:bold;
color:#0000ff;
background-color:transparent
}
```

注意:ID 选择符局限性很大,只能单独定义某个元素的样式,一般只在特殊情况下使用。

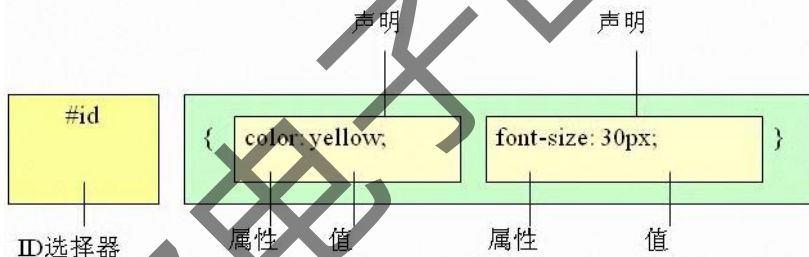


图 7-4 ID 选择器

(4) 注释:/* ... */。

在 CSS 中插入注释来说明代码的意思,注释有利于你或别人在以后编辑和更改代码时理解代码的含义。在浏览器中,注释是不显示的。CSS 注释以“/*”开头,以“*/”结尾,如下:

```
/* 定义段落样式表 */
p
{
text-align: center; /* 文本居中排列 */
color: black; /* 文字为黑色 */
font-family: arial /* 字体
为 arial */
}
```

(5) 选择符的声明。

CSS 声明是由属性、冒号、属性值和分号组成的,如 font-size:12px;/*, font-size 代表字体大小,12px 表示字体大小的值,*/表示选择符中定义的最后一个声明,声明后的分号可以省略。

多个选择符可以集体声明,就是将所有属性相同的 CSS 标记并同时进行声明,选择器之间使用英文“.”隔开。

(6)选择符的嵌套。

子选择符和选择符之间使用空格隔开。

2. CSS 样式进阶 - 使用 CSS 控制页面

(1)行内样式。

```
<p style="color=#0000FF;font-size:20px;text-decoration:underline;">
```

(2)内嵌式,如图 7-5 所示。

```
<html>
<head>
<title>页面标题</title>
<style type="text/css">
<!--
p{
    color:#FF00FF;
    text-decoration:underline;
    font-weight:bold;
    font-size:25px;
}
-->
</style>
</head>
<body>
<p>紫色、粗体、下划线、25px的效果1</p>
<p>紫色、粗体、下划线、25px的效果2</p>
<p>紫色、粗体、下划线、25px的效果3</p>
</body>
</html>
```

图 7-5 内嵌式

(3)链接式,如图 7-6 所示。

```
<html>
<head>
<title>页面标题</title>
<link href="1.css" type="text/css" rel="stylesheet">
</head>
<body>
<h2>CSS标题1</h2>
<p>紫色、粗体、下划线、25px的效果1</p>
<h2>CSS标题2</h2>
<p>紫色、粗体、下划线、25px的效果2</p>
</body>
</html>
```

```

color:#0000FF;
}
p{
color:#FF00FF;
text-decoration:underline;
font-weight:bold;
font-size:20px;
}

```

图 7-6 链接式

(4)导入样式,如图 7-7 所示。

```

<html>
<head>
<title>页面标题</title>
<style type="text/css">
<!--
@import url(l.css);
-->
</style>
</head>
<body>
<h2>CSS标题1</h2>
<p>紫色、粗体、下划线、25px的效果1</p>
<h2>CSS标题2</h2>
<p>紫色、粗体、下划线、25px的效果2</p>
<h3>CSS标题3</h3>
<p>紫色、粗体、下划线、25px的效果3</p>
</body>
</html>

```

图 7-7 导入式

7.1.2 网页的层 DIV

DIV 元素是用来为 HTML 文档内大块 (block-level) 内容提供结构和背景的元素。DIV 的起始标签和结束标签之间的所有内容都是用来构成这个块的,其中所包含元素的特性由 DIV 标签的属性来控制,或者是通过使用样式表格式化这个块来进行控制。

1. 盒子模型

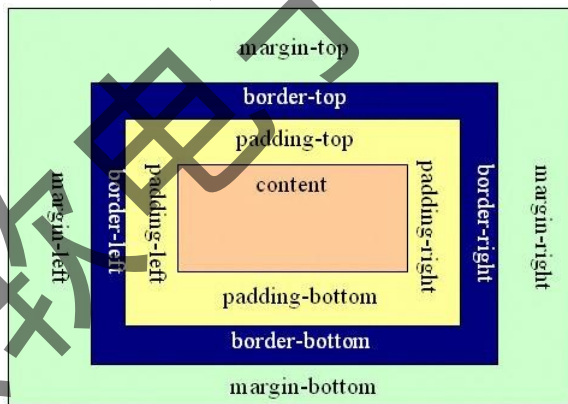


图 7-8 盒子模型

2. 元素定位

(1) Float 定位,如图 7-9 所示。

```

<html>
<head>
<title>float 属性</title>
<style type="text/css">
<!--
body{
margin:15px;

```

```

font-family:Arial;
font-size:12px;
}
.father{
background-color:#fffea6;
border:1px solid #111111;
padding:25px;/* 父块的 padding */
}
.son1{
padding:10px;/* 子块 son1 的 padding */
margin:5px;/* 子块 son1 的 margin */
background-color:#70baff;
border:1px dashed #111111;
float:left;/* 块 son1 左浮动 */
}
.son2{
padding:5px;
margin:0px;
background-color:#ffd270;
border:1px dashed #111111;
}
-->
</style>
</head>
<body>
<div class="father">
<div class="son1">float1</div>
<div class="son2">float2</div>
</div>
</body>
</html>

```

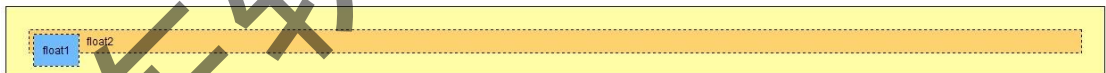


图 7-9 Float 定位

(2) Position 定位, 如图 7-10 所示。

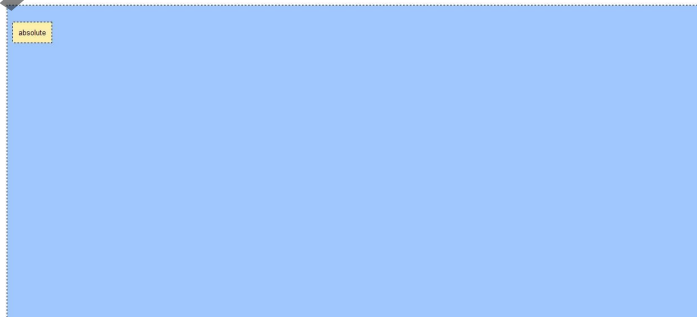


图 7-10 Position 定位

```
<html>
<head>
<title>position 属性</title>
<style type="text/css">
<!--
body{
    margin:10px;
    font-family:Arial;
    font-size:13px;
}
# father{
    background-color:# a0c8ff;
    border:1px dashed #000000;
    width:100%;
    height:100%;
}
# block{
    background-color:# fff0ac;
    border:1px dashed #000000;
    padding:10px;
    position:absolute; /* absolute 绝对定位 */
    left:20px; /* 块的左边框离页面左边界 20px */
    top:40px; /* 块的上边框离页面上边界 40px */
}
-->
</style>
</head>
<body>
<div id="father">
<div id="block">absolute</div>
</div>
</body>
</html>
```

(3) Z-index 定位,如图 7-11 所示。

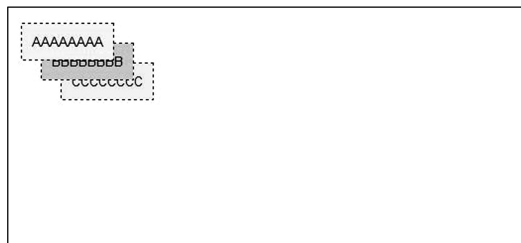


图 7-11 Z-index 定位

```
<html>
<head>
<title>z-index 属性</title>
<style type="text/css">
<! --
body{
    margin:10px;
    font-family:Arial;
    font-size:13px;
}
#block1{
    background-color:#fff0ac;
    border:1px dashed #000000;
    padding:10px;
    position:absolute;
    left:20px;
    top:30px;
    z-index:1;/* 高低值 1 */
}
#block2{
    background-color:#ffc24c;
    border:1px dashed #000000;
    padding:10px;
    position:absolute;
    left:40px;
    top:50px;
    z-index:0;/* 高低值 0 */
}
#block3{
    background-color:#c7ff9d;
    border:1px dashed #000000;
    padding:10px;
    position:absolute;
    left:60px;
    top:70px;
    z-index:-1;/* 高低值 -1 */
}
-->
</style>
</head>
<body>
<div id="block1">AAAAAAA</div>
```



```
<div id="block2">BBBBBBBB</div>
<div id="block3">CCCCCCCC</div>
</body>
</html>
```

7.2 项目实践

使用 dreamweaver 网页设计工具生成同 PS 设计的首页一样的 html 文件。

7.2.1 建立站点

步骤 1: 用 PS 设计制作页面, 效果图如图 7-12 所示。

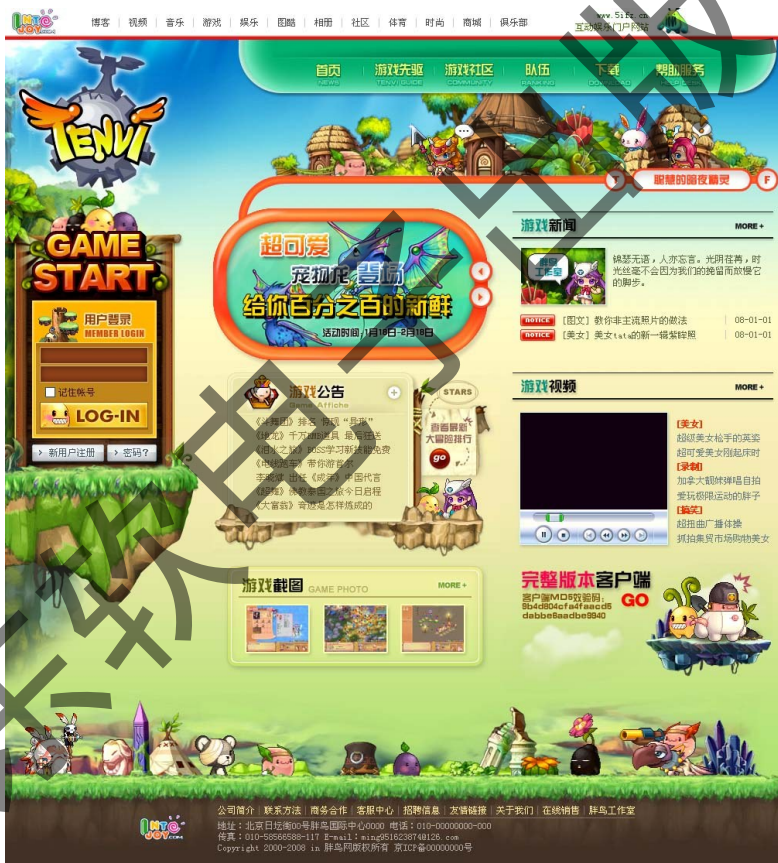


图 7-12 页面效果图

步骤 2: 打开 dreamweaver, 新建相应页面, 如图 7-13 所示。



图 7-13 新建页面

步骤 3: 新建相应样式, 如图 7-14 所示。



图 7-14 新建样式

步骤 4: 页面与样式做链接, 如图 7-15 所示。



图 7-15 页面与样式链接

7.2.2 切图

步骤 5: 打开之前用 PS 设计的页面, 如图 7-16 所示。

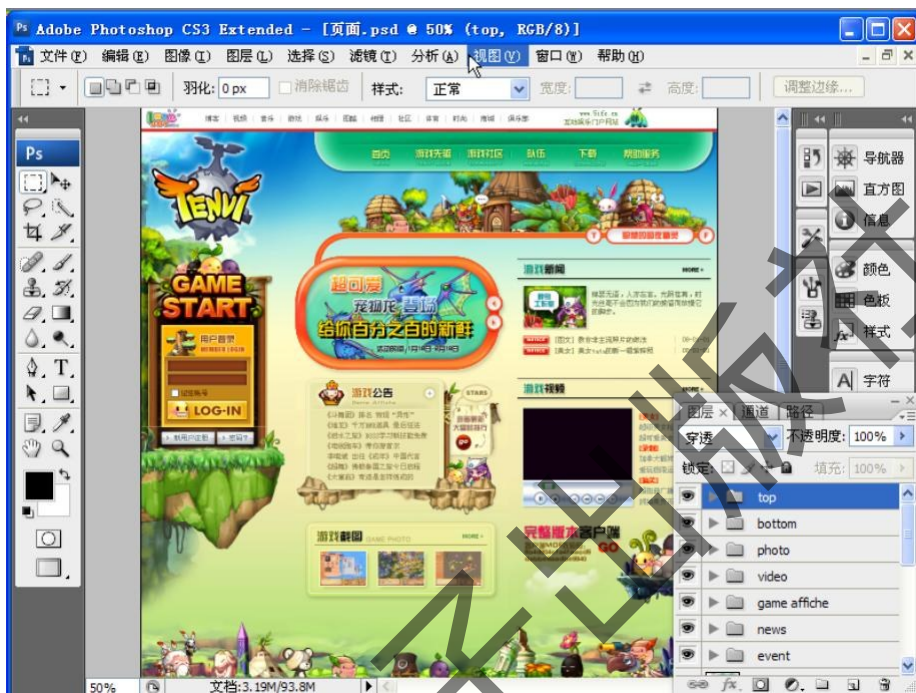


图 7-16 psd 格式的页面效果图

步骤 6: 使用标尺做参考线, 如图 7-17 所示。



图 7-17 做参考线

步骤 7: 隐藏暂时不使用的图层和组, 如图 7-18 所示。

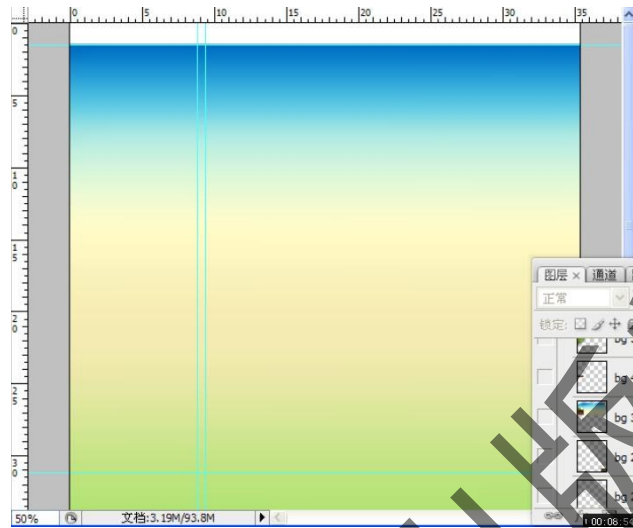


图 7-18 隐藏某些图层

步骤 8: 在需要的图层上选择矩形选框, 如图 7-19 所示。

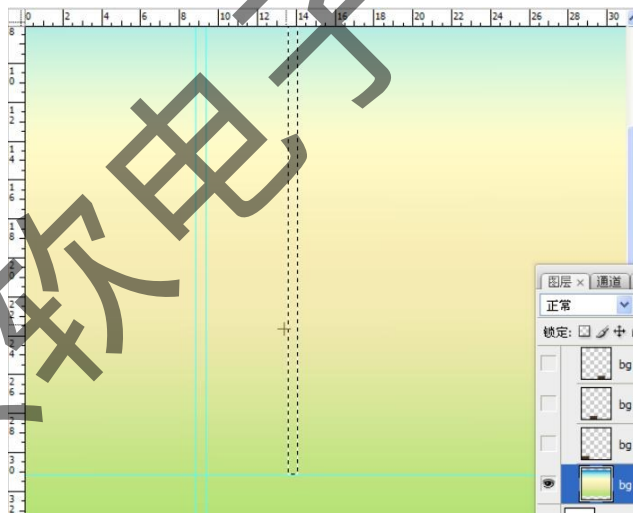


图 7-19 矩形选框

步骤 9: 生成相应的图片, 如图 7-20 所示。

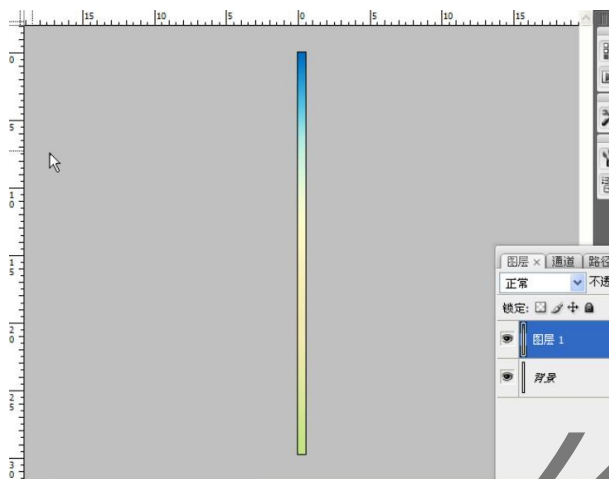


图 7-20 生成图片

步骤 10: 使用同样的方法切割并生成其他图片, 如图 7-21 所示。

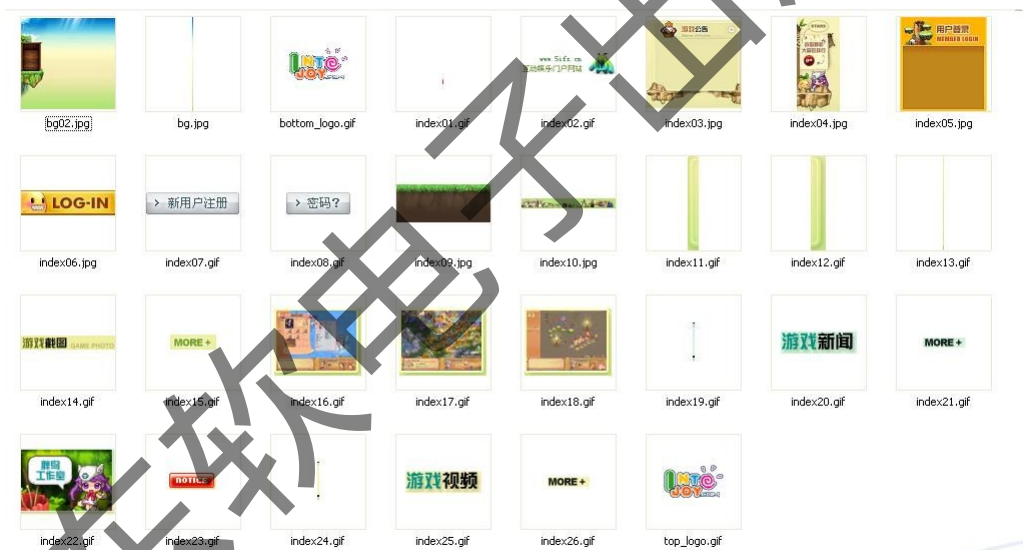


图 7-21 切割后的图片

7.2.3 生成 html 页面

步骤 11: 定义页面基本样式, 如图 7-22 所示。

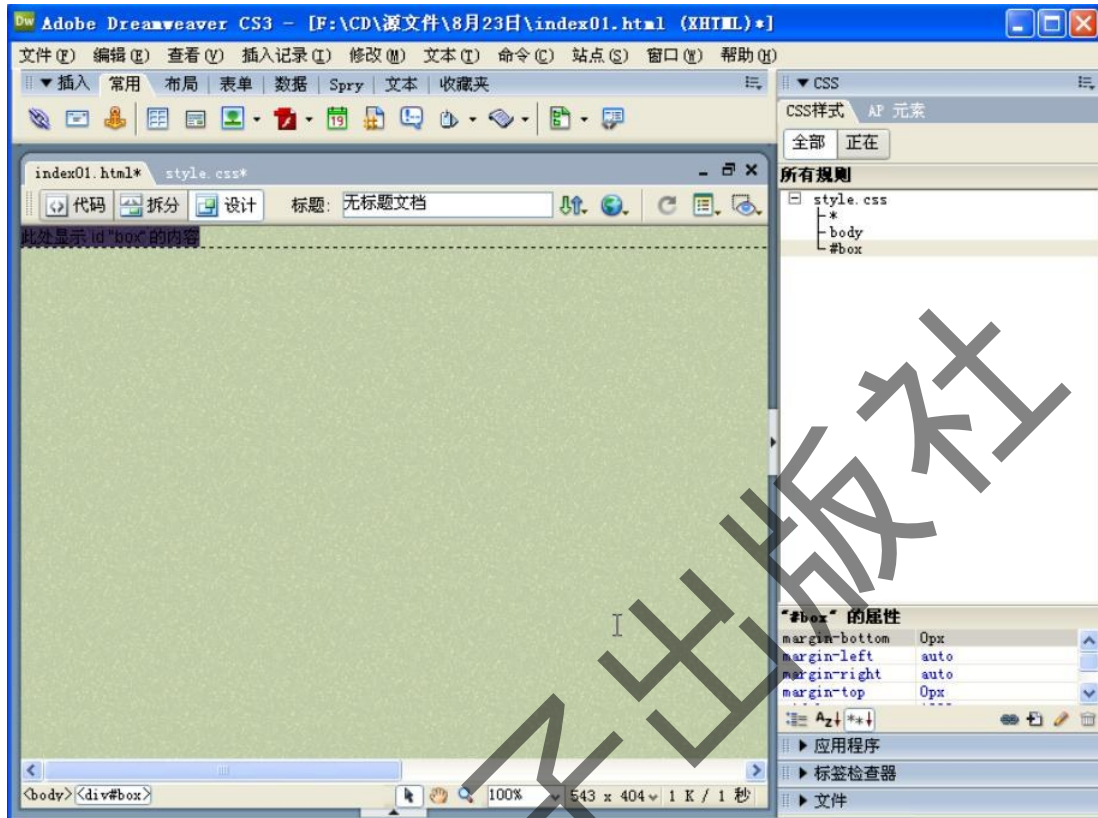


图 7-22 页面基本样式

步骤 12: 导入切割后的图片——以制作页面导航部分为例。

步骤 13: 新建一个层, 如图 7-23 所示。



图 7-23 新建层

步骤 14: 在样式表中写入相应的设置, 如图 7-24 所示。

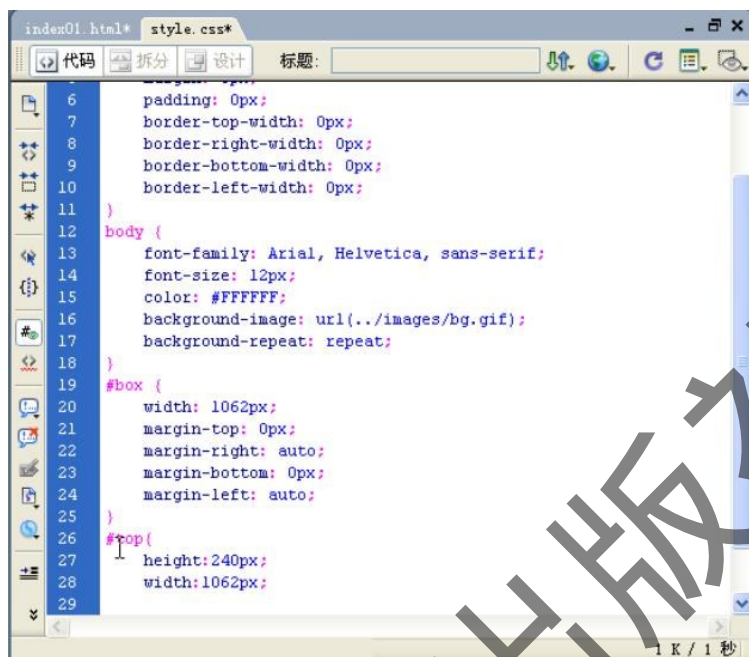


图 7-24 层的设置

步骤 15: 在此层中导入刚才切割的图片或已有的 flash, 如图 7-25、图 7-26 所示。



图 7-25 导入图片或 flash



图 7-26 导入效果

步骤 16: 重复以上步骤生成页面导航, 如图 7-27 所示。



图 7-27 页面导航

步骤 17: 重复以上步骤生成页面主体, 如图 7-28 所示。



图 7-28 页面主体

步骤 18: 重复以上步骤生成页面版底, 如图 7-29 所示。

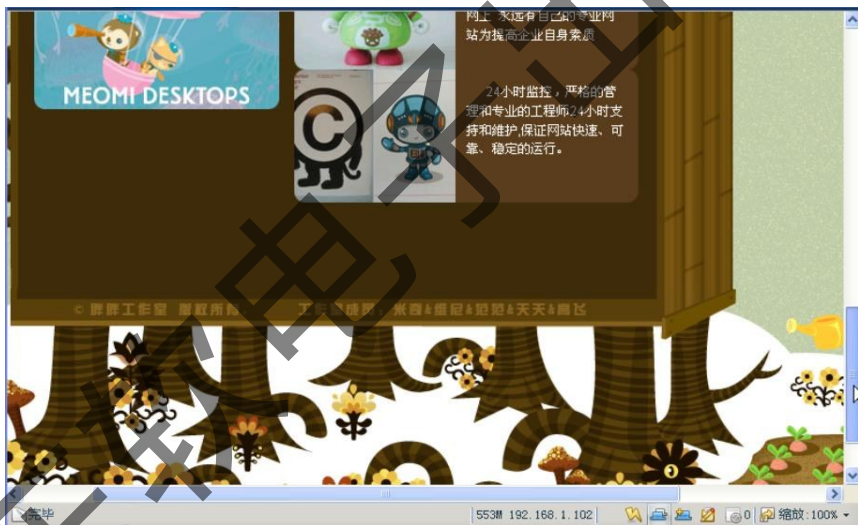


图 7-29 页面版底

7.3 扩展知识

7.3.1 路径

路径主要分为绝对路径和相对路径两部分。

1. 绝对路径

绝对路径指把路径写全, 比如 C 盘下的 windows 文件夹中有 notepad.exe 这个文件, 那么

要写出它的绝对路径就是 `c:\windows\notepad.exe`。

优点:可以在任何位置使用这个文件。

缺点:如果这个文件被移动位置,那么将无法正确访问。

2. 相对路径

相对路径一般是由 `[..]`、`[.]` 以及 `/` 等分隔符组成的,如果想访问 `C:\windows\notepad.exe` 这个文件,那么就可以用 `windows/notepad.exe` 这样的路径来访问。

说明:

可以访问上一级目录中文件;可以没有任何分隔符而直接使用文件名做链接,比如 `首页`; `../..` 访问多级目录中的文件, `../` 表示一个目录。

优点:可以随时访问文件,就算整个站点文件夹被移动了,还是可以正常访问,不会出现无法访问的问题。

缺点:很容易搞错文件的位置或目录的位置。

3. 总结

一般的存放路径为:建一个大的文件夹(项目的英文或者中文拼音名字),然后在这个文件夹中分别建立 `img` 文件夹(放图片)、`js` 文件夹(放 `js` 脚本)、`CSS` 文件夹(放 `CSS` 样式文件)和 `html` 文件夹。

7.3.2 文件及图片命名

1. 文件命名

文件命名要以最简短的体现清晰的含义。文件名尽量以英文单词为主,单个单词文件名称全部小写。如文件名由两个或两个以上单词组成,文件名称第二个单词起的第一个字母要大写,例如:友情链接(单个)——`link`、公司简介(两个)——`aboutUs`。所有命名如果利用数字编号来区分文件,第一个文件命名中的 `1` 必须忽略,例如:`aboutUs`、`aboutUs2`、`aboutUs3`。

命名单词组合顺序为:划分文件项目或类型单词+功能性单词+用途类单词或编号,例如 `productAdd`。`product` 表示产品项目的名称,后面加上 `Add` 表示这个页面的功能为添加(增加)。如有多个添加产品的页面就可以写成 `productAdd`、`productAdd2`、`productAdd3`。

2. 文件夹命名

(1)网页文件按照板块、项目来进行文件夹的划分,例如前台产品类文件被归纳在 `product` 文件夹内,如果每个项目页面不多,可以同时存放在一个目录下,但是要确保以后增加的量小,管理维护方便,否则应一律按照文件夹进行划分。后台举例:卖方后台的产品添加页面为 `buyerManage/productAdd`。

(2)每个板块或项目的图像文件要在该目录内命名为 `images`。

(3)`swf` 文件建立在该目录下的 `images` 文件夹内的 `swf` 文件夹。

(4)所有 `CSS` 文件要统一放在根目录下的 `style` 文件内,命名规则详见 `CSS` 命名。

(5)`js` 文件放在该网页目录下的 `js` 文件夹内。

3. 网页命名

`HTML` 代码中的语义性标签可以借鉴使用,但不可以单独使用。例如注册表单页面不可以定义为 `form`,正确的写法是 `regForm`。

4. 图片命名

图片命名与文件命名雷同,区别在于图片文件命名必须指定它所用的版块名或功能名称。例如页首 banner 不能直接命名 banner,正确的命名为 headerBanner,公司简介的 banner 就是 aboutUsBanner。

标志要有区分,如大标志为 LogoBg、小标志为 LogoSm。

注册小标志,例如小箭头、对号之类的,可以这样命名:regIco、regIco2。

导航背景图片:navBg。

举例:一个网页中有三个导航,那么我们可以按照数字编号来命名。

数字编号:nav、nav2、nav3。

背景图片命名:navBg、nav2Bg、nav3Bg。

位置命名:navTop、navCenter、NavBottom。

背景图片命名:navTop-Bg、navCenter-Bg、NavBottomBg。

如图片应用到多个页面或位置可用通用的名称,例如网页背景可命名为 bodyBg。

5. CSS 命名规则

CSS 文件名称要围绕所应用的范围,例如买方管理中心头部的 CSS 文件为 sellerManageHeader.css。注意:如果该 CSS 样式只针对一个页面应用时,则不允许建立 CSS 文件,一律写在该文件的 head 标签内。

在对 CSS 样式进行定义时,要尽量应用 class 定义。定义全部标签要谨慎考虑后期的可延展性,不能影响后期升级开发。

在定义局部样式时尽可能利用包含关系,例如定义 Class="loginInput" 中的文本域时不要为了方便书写,就写成了 input{...},正确的书写方式为.loginInput input{...}。

7.4 工程实训

7.4.1 建立站点

完成 index 文件及 image、style 文件夹的创建。

7.4.2 切图

把 psd 文件转成 DIV+CSS 组成的 HTML 文件。

7.4.3 成绩计算

成绩按表 7-1 来计算,供参考。

表 7-1

成绩计算表

阶段	百分比	评分标准	评语
生成 ps 设计的页面	20%		
切图	20%		
生成 HTML 页面	40%		
路径和文件、图片的设置及命名	20%		
总评			

东软电子出版社