

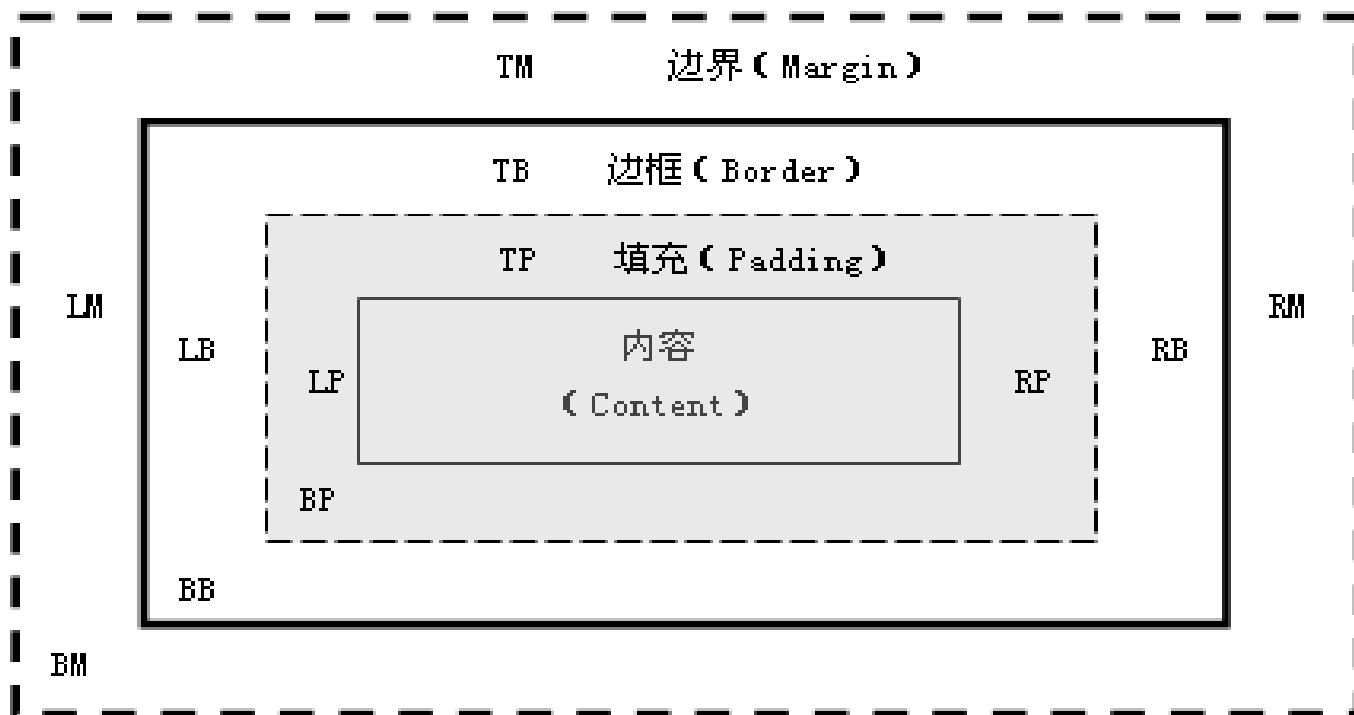
第7章 应用div+CSS布局网页

zhaozhl@lzu.edu.cn

- 通过CSS能够定制和控制文本、图像等内容在网页中的表现，尤其能够使文本在网页中“表现”为不同的字体、大小、颜色。
- 此外，使用div元素和CSS还能够对网页进行布局——控制元素及其内容在网页中的位置及所占平面空间。

- 盒子模型
- 与盒子模型有关的样式特性
- 元素及其在网页中的排列
- div+CSS网页布局
- Web标准网页设计举例
- div+CSS布局 vs. 表格布局

7.1 盒子模型



盒子模型

注意：在盒子模型中，填充、边框和边界还可以分解为上、右、下和左四个部分。如，填充又可以分解为**上填充 (top padding)**、**右填充 (right padding)**、**下填充 (bottom padding)**和**左填充 (left padding)**。

7.2 与盒子模型有关的样式特性



与盒子模型有关的样式特性主要分为方框和边框两大类。

7.2.1 方框特性



1. 宽度 (width)

该特性用于设置元素内容的宽度。width特性值通常是以px (像素) 和em (字体高度) 为单位的相对值。

2. 高度 (height)

该特性用于设置元素内容的高度。height特性值通常是以px (像素) 和em (字体高度) 为单位的相对值。

3. 浮动方式 (float)

该特性用于设置元素的浮动方式。float特性值可以是left、right或none，分别表示元素向左浮动、向右浮动或不浮动，其中none是默认值。

4. 填充宽度 (padding)

该特性用于设置元素的填充宽度。padding特性值通常是以px (像素) 和em (字体高度) 为单位的相对值，也可以使用padding-top、padding-right、padding-bottom和padding-left特性分别设置上、右、下和左填充的宽度。

5. 边界宽度 (margin)

该特性用于设置元素的边界宽度。margin特性值通常是以px (像素) 和em (字体高度) 为单位的相对值，也可以使用margin-top、margin-right、margin-bottom和margin-left特性分别设置上、右、下和左边界的宽度。

7.2.2 边框特性



1 . 边框宽度 (border-width)

该特性用于设置元素的边框宽度。border-width特性值通常是以px (像素) 和em (字体高度) 为单位的相对值 , 也可以使用border-top-width、border-right-width、border-bottom-width和border-left-width特性分别设置上、右、下和左边框的宽度。

2 . 边框样式 (border-style)

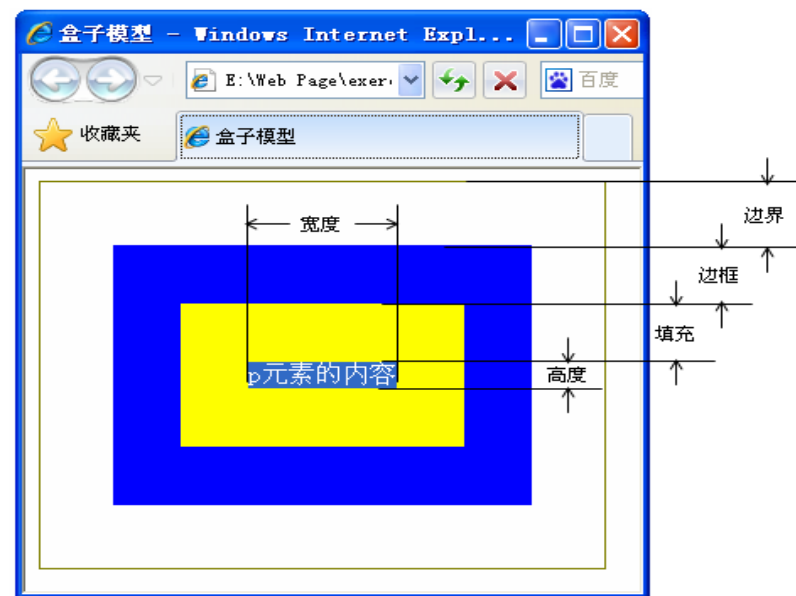
该特性用于设置元素的边框样式。border-style特性值可以none (无边框) 、dotted (点划线) 、dashed (虚线) 、solid (实线) 、double (双线) 、groove (槽状) 、ridge (脊状) 、inset (凹陷) 和outset (凸出) 。也可以使用border-top-style、border-right-style、border-bottom-style和border-left-style特性分别设置上、右、下和左边框的样式。

3 . 边框颜色 (border-color)

该特性用于设置元素的边框颜色。与color和background-color特性值类似 , border-color特性值可以是颜色名称 , 也可以是rgb代码 , 还可以是一个三位或六位的十六进制值数。也可以使用border-top-color、border-right-color、border-bottom-color和border-left-color特性分别设置上、右、下和左边框的颜色。

7.2.3 盒子模型演示

```
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type" content="text/html; charset=gb2312"/>
  <title>盒子模型</title>
  <style type="text/css">
    .box { color:red; background-color:yellow;
      padding:40px; border:40px solid blue; margin:40px }
  </style>
</head>
<body>
  <table style="border:1px solid olive">
    <tr><td><p class="box">p元素的内容</p></td></tr>
  </table>
</body>
</html>
```



在 IE 浏览器中演示盒子模型

7.3 元素及其盒子在网页中的排列



- 根据盒子模型及其原理，HTML文档主体中的每个元素都被Web浏览器看成一个矩形盒子。
- 在Web标准网页的设计与制作中，元素及其盒子的排列方法主要有三种：**正常流（Normal Flow）**、**浮动（Float）**和**定位（Positioning）**。

7.3.1 正常流中的元素盒子

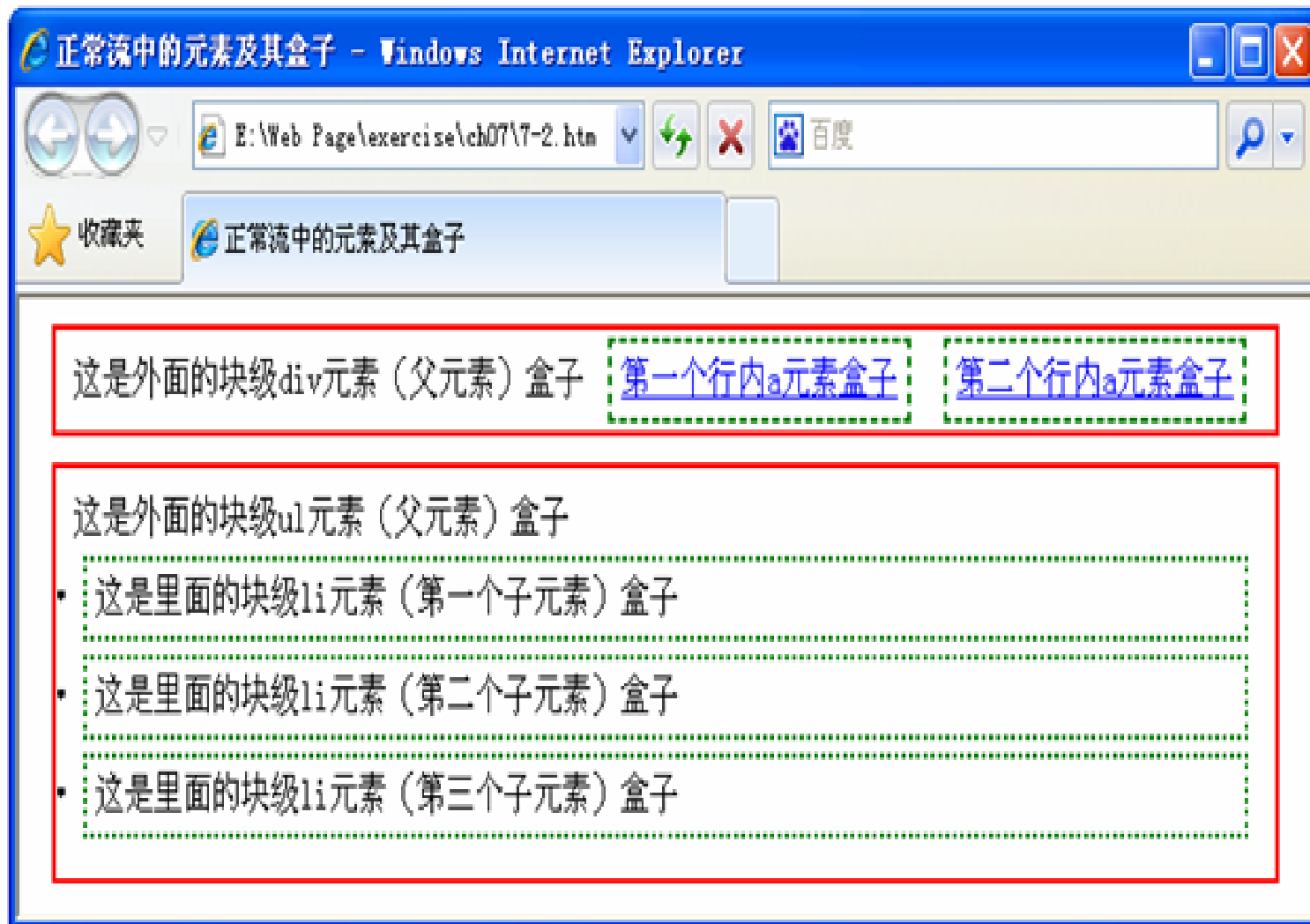


- 默认情况下，所有元素盒子都是在正常流中定位的。
- 在正常流中，元素盒子在Web浏览器网页中的位置是由元素在HTML文档中的位置决定的。
- 具体而言，**相邻的行内元素的盒子**在同一水平行中从左到右依次排列；
- **相邻的块级元素的盒子**从上到下依次排列，其中每个块级元素的盒子独占整个一行；
- 如果**两个块级元素具有父元素和子元素的嵌套关系**，那么父元素的块级元素盒子包含子元素的块级元素盒子。

7.3.1 正常流中的元素盒子

```
<head>
.....
<style type="text/css">
  div { padding:10px; border:2px solid red; margin:10px }
  a { padding:5px; border:2px dashed green; margin:5px }
  ul { padding:10px; border:2px solid red; margin:10px }
  li { padding:5px; border:2px dotted green; margin:5px }
</style>
</head>
<body>
  <div>
    这是外面的块级div元素（父元素）盒子
    <a href="#">第一个行内a元素盒子</a>
    <a href="#">第二个行内a元素盒子</a>
  </div>
  <ul>
    这是外面的块级ul元素（父元素）盒子
    <li>这是里面的块级li元素（第一个子元素）盒子</li>
    <li>这是里面的块级li元素（第二个子元素）盒子</li>
    <li>这是里面的块级li元素（第三个子元素）盒子</li>
  </ul>
</body>
```

7.3.1 正常流中的元素盒子



7.3.2 元素盒子的浮动



- 在通常和默认情况下，每个元素的float特性值都是none，也就是正常流情况。
- 但是，如果将float特性值设置为left或right，元素盒子就会向其父元素的左侧或右侧浮动。

7.3.2 元素盒子的浮动

```
<head>
  <meta http-equiv="content-type" content="text/html; charset=gb2312"/>
  <title>元素盒子的浮动</title>
  <style type="text/css">
    ul { list-style:none }
    li { float:left; width:180px; border:1px solid; }
  </style>
</head>
<body>
  <ul>
    <li>第一个li元素盒子</li>
    <li>第二个li元素盒子</li>
    <li>第三个li元素盒子</li>
  </ul>
</body>
```

第一个li元素盒子	第二个li元素盒子	第三个li元素盒子
-----------	-----------	-----------

7.3.3 应用CSS和盒子浮动制作水平导航条



```
<head>
.....
<style type="text/css">
  ul { width:450px; margin:0 auto; list-style:none; }
  li { float:left; width:150px; padding:10px 0; background-color:#FA1; text-align:center; }
  a { text-decoration:none; color:red; }
  a:hover { color:purple; }
</style>
</head>
<body>
  <ul>
    <li><a href="#">链接 1</a></li>
    <li><a href="#">链接 2</a></li>
    <li><a href="#">链接 2</a></li>
  </ul>
</body>
```

链接 1

链接 2

链接 2

7.4 div+CSS网页布局



- 随着CSS技术的发展、应用和规范，div+CSS网页布局已经成为Web网页设计与制作的标准方法，并已经取代传统的表格布局方法。
- 在div+CSS网页布局中，能够使用div元素和CSS样式表灵活地对网页的平面空间进行分区。

7.5 Web标准网页设计举例



- 为了使网页能够在不同Web浏览器中正常显示，Web标准网页设计应该遵循“**内容和结构与表现的分离**”的原则。
- 在Web标准网页设计与制作中，“**内容和结构**”保存在**XHTML文档**中，“**表现**”的定义则保存在**CSS文档**中。
- 在**XHTML文档的头部**，使用**link元素**可以链接CSS文档中的**外部样式表**。
- 在XHTML文档的主体中，如果**元素**与外部样式表中的**选择器**及其规则匹配，元素中的内容将按照规则中的样式声明“**表现**”在网页中。

7.5 Web标准网页设计举例



7.5 Web标准网页设计举例



(1) 首先，在**CSS文档**中创建外部样式表。CSS代码如下：

```
#wrapper { width:780px; margin:0 auto; }
#logo { width:760px; height:30px; padding:10px; background-color:#CCC; font-size:30px; }
#banner { width:780px; height:100px; background-image:url("../ch02/2-4.gif"); }

#naviBar { width:780px; height:2em; padding:0; background-color:#FA1; }
#naviBar ul { width:780px; padding:0.5em 0; margin:0 auto; list-style:none; }
#naviBar li { float:left; width:260px; text-align:center; }
#naviBar a { text-decoration:none; color:red; }
#naviBar a:hover { color:purple; }

#mainContent { width:760px; padding:10px; background-color:#CFF; }
    .para { text-indent:2em; }

#footer { width:760px; height:1em; padding:10px; background-color:#FF9; }
    address { text-align:center; }
```

7.5 Web标准网页设计举例

(2) 然后，在XHTML文档中使用如下代码（部分）：

```
<div id="wrapper">
  <div id="logo">Web标准网页设计原理与制作技术</div>
  <div id="banner"></div>

  <div id="naviBar">
    <ul>
      <li><a href="#">课程介绍</a></li>
      <li><a href="#">课堂教学</a></li>
      <li><a href="#">学习参考</a></li>
    </ul>
  </div>

  <div id="mainContent">
    <p class="para">本课程以HTML 4.01、XHTML 1.0和CSS 2.1技术规范为基础.....</p>
    <p class="para">.....</p>
    <p class="para">.....</p>
  </div>

  <div id="footer">
    <address>Copyright © 2010-2012 《网页设计与制作》课程建设小组</address>
  </div>
</div>
```

div+CSS vs. Table

- 常用的Web基本框架布局规范：
 - **Div+CSS**
 - Table

- **表现和内容相分离**

将设计部分剥离出来放在一个独立样式文件中，HTML文件中只存放文本信息。

- **提高搜索引擎对网页的索引效率**

用只包含结构化内容的HTML代替嵌套的标签，搜索引擎将更有效地搜索到你的网页内容，并可能给你一个较高的评价。

• 提高页面浏览速度

对于同一个页面视觉效果，采用CSS+DIV重构的页面容量要比TABLE编码的页面文件容量小得多，前者一般只有后者的1/2大小。

还有因为<table>标签是要等</table>下载好以后才可以显示的，所以整个网页是一下子跳出来的，这样让人感觉起来显示速度非常的慢。而使用css+div,<div> 标签不用等</div>下载好就可以显示里面的内容的,所以让人感觉打开网页的速度非常之快。

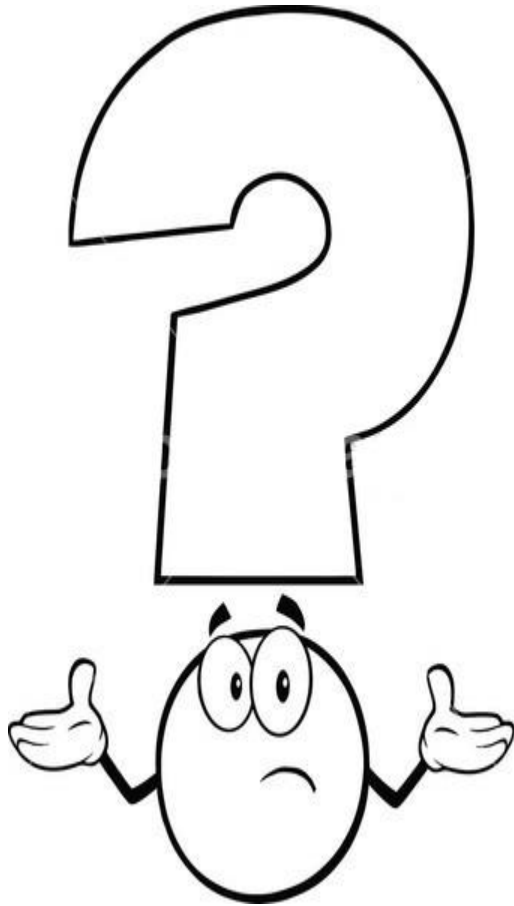
- **易于维护和改版**

只要简单的修改几个CSS文件就可以重新设计整个网站的页面。

从以上的描述来看，采用CSS+DIV对网站重构可以大大提升网站用户与搜索引擎的友好度。

CSS+DIV所以成为目前网页布局主流

- **Table 最大缺点是代码臃肿**，至少需要写下 `<table><tr><td>` 这三个标签之后，才能开始真正的内容；另外，Table 的各种标签中还包含了复杂的属性定义，而 Div 只需 `<div>` 一个标签。
- **页面渲染性能问题**：浏览器需要将整个表格完全读完后才会开始渲染。
- **不利于搜索引擎优化**：搜索引擎喜欢内容与修饰分开。
- **可访问性差**：屏幕朗读软件和盲文浏览器无法很好地理解 Table 中的内容。
- **一般table只用做于数据处理。**



Questions ?