

IT 之行 始于足下



# 第七章

## WEB原则与页面布局基础



足下软件教育



# 本章单词

Structure	(体系, 构造, 构造)
Presentation	(简介, 演示, 外观, 表象)
Behavior	(反应, 行为, 举止)
XML	( <b>Extensible Markup Language</b> , 即可扩展标识语言)
DTD	( <b>Document Type Definition</b> , 文档类型定义)
Transitional	(过渡性的)
Strict	(严格的)
margin	(外边的空白, 外边距, 外补丁)
padding	(填充物, 内边距, 内补丁)
block	(街区, 块)
inline	(内联, 行内)
overflow	(泛滥, 溢出)
relative	(相正确)
absolute	(绝正确)
vertical	(垂直的, 竖直的)





# 预习检验



提问



- ④ 当使用margin属性对元素的四个外边距进行设置，依次的顺序为( )。
  - A. top bottom left right
  - B. left right top bottom
  - C. top left bottom right
  - D. top right bottom left ✓
  
- ④ 使用position属性可设置元素的定位机制。要使元素以它的父元素为参照，进行相对定位，应为postion属性赋下列（ ）的值。
  - A. static
  - B. absolute
  - C. relative ✓
  - D. fixed



当页面中有多种元素在垂直于浏览器窗口的方向上发生层叠时，可使用( )属性设置它们的层叠顺序，此属性的值越( )则此元素就越接近浏览者。

- A. z                      大
- B. index                小
- C. z-index            大     ✓
- D. z-number          大

要将一种元素隐藏，而且要求此元素原来占据的页面空间被释放，应为此元素设置( )属性，并赋值为( )。

- A. visibility                none
- B. visibility                hidden
- C. display                    hidden
- D. display                    none     ✓



④ CSS盒状模型要求盒子由四个content , margin , padding , border部分构成。由盒子的外部到盒子内部, 这四个部分的顺序依次是( )。

- A. border          padding          margin          content
- B. margin          border          padding          content          ✓
- C. content          padding          border          margin
- D. border          margin          content          padding



# 网站重构与WEB原则

## → 网站重构

- ✓ 把“未采用CSS，大量使用HTML进行定位、布局，或者虽然已经采用CSS，但是未遵照HTML结构化原则的站点”变成“让标识回归标识的原本意义。经过在HTML文档中使用结构化的标识以及用CSS控制页面体现，使页面的实际内容与它们呈现的格式相分离的站点。”的过程就是**网站重构**。

## → 网站重构 >>> WEB原则





- ➔ 网页主要由三个部分构成：**构造**（Structure）、**体现**（Presentation）和**行为**（Behavior）。它们都有各自的原则。
- ➔ 构造
  - 构造是指一种网页有哪些内容以及这些内容怎样被清楚地组织在一起。
  - 网页构造的原则主要是XML语言和XHTML语言。
- ➔ 体现
  - 体现是指一种网页的内容以何种样式来呈现。
  - 网页体现的原则是CSS语言。
- ➔ 行为
  - 行为是指一种网页及其所它包括的网页元素以及它所属的浏览器窗口怎样与顾客交互。
  - 网页行为由JavaScript脚本语言来实现。





- ➔ XHTML是HTML的替代者，它是从HTML继承而来，但使用了XML的规则进行严格规范。
- ➔ XHTML与老式的HTML的不同之处，主要来自于XML语法规则的格式要求，即要求“格式良好”。
  - ✓ 必须有且仅能有一种根元素<html></html>
  - ✓ 标签必须正确地嵌套
  - ✓ 标签必须关闭
  - ✓ 标签名和属性名必须使用小写
  - ✓ 属性必须赋值
  - ✓ 属性值必须包括在引号内
  - ✓ 特殊字符必须用实体编码来表达
  - ✓ 图片必须有替代文字
  - ✓ 注释的内容中不能包括”--”



## → XHTML文档的有效性

- ✓ 网页文档中能够使用的标签有哪些，标签有哪些可用的属性，属性可用哪些值，标签之间怎样嵌套，特殊字符怎样表达等等。

## → XHTML文档的有效性经过DTD（Document Type Definition，文档类型定义）来定义和验证。

- ✓ 在文档第一行（<html>标签开始之前）添加**DOCTYPE**即文档类型申明。

## → 万维网联盟（W3C）为XHTML制定了三种DTD

### ✓ 过渡的(Transitional)

→ `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "url">`

### ✓ 严格的(Strict)

→ `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "url">`

### ✓ 框架集的(Frameset)

→ `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" " url">`





# 构造与体现分离

- CSS的出现，使得网页的构造与体现比较清楚地分离成为可能。
- 对于网页包括的数据内容，由HTML标签负责承载，同步，文档内众多标签的先后顺序和嵌套关系，也阐明了这些数据内容的构造。
- 对于网页中的每一种元素最终在浏览器中以何种样式来呈现，则由CSS的样式规则来负责定义。
- CSS样式规则一般编写在网页的头部<head></head>中，这么可使得用来承载数据和描述构造的HTML代码与控制样式的CSS代码分离开来。假如需要两者更进一步地分离，则可将CSS样式规则转移到独立的外部样式表文件中。
- 网页的构造与体现分离，带来的好处。





# WEB原则与CSS实现网页布局

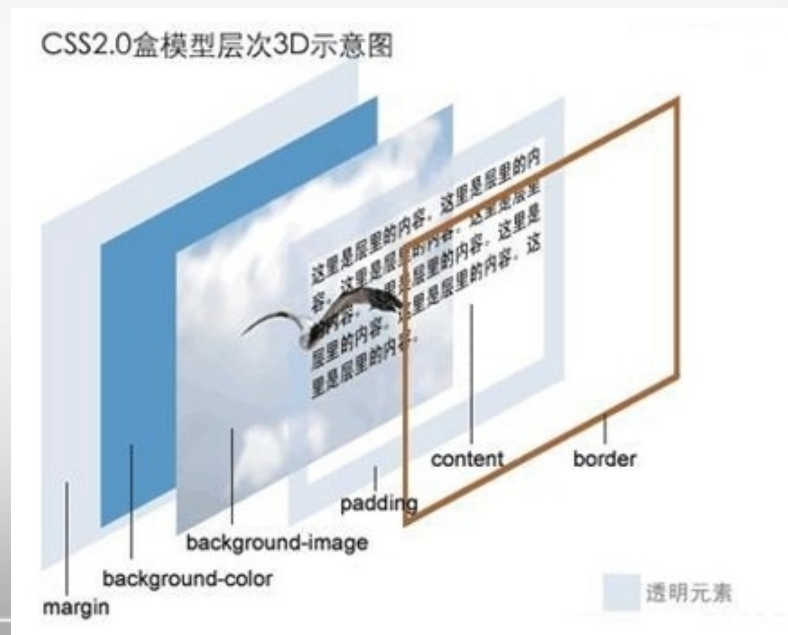
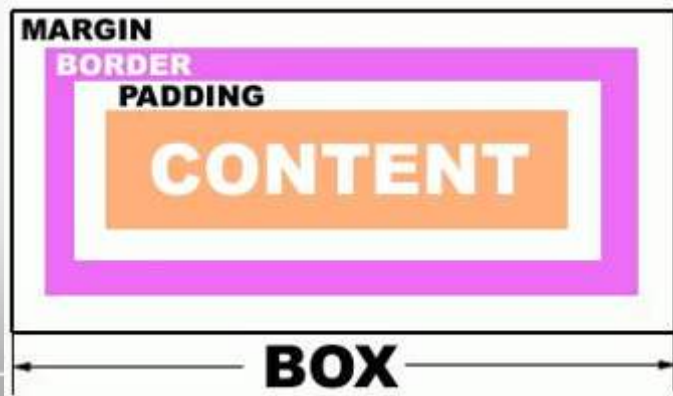
- 网页布局又可了解为**网页排版**。简朴地说就是：网页中的某一块内容应该放置在网页的哪个位置，占用多大的面积。
- 大量使用表格实现页面布局，不符合WEB原则。
- 目前普遍采用**CSS**结合**div**标签来实现网页的布局。
- 将整个网页的全部内容**分割**成为相对独立的**内容块**，这些内容块分别用一种**div**标签来**包裹**，再使用**CSS**的属性分别设置这些**内容块的位置和尺寸**。这么整个页面的版式就受到了控制，从而实现了网页的布局。





# CSS的盒状模型

- ➔ CSS的盒状模型（Box Model），是使用CSS实现网页布局的基础。
  - ➔ 页面中可放置内容的窗口元素称为盒子即Box，盒子就是一种矩形区域。
  - ➔ 每个盒子都有内容Content，内边距Padding，边框Border，外边距Margin。
  - ➔ 盒状模型用于描述它们之间的层次，关系与相互的影响。
- ✓ margin: 外边距
  - ✓ border: 边框
  - ✓ padding: 内边距，又称内补丁或间隙
  - ✓ content: 内容





## 两种盒子

- ➔ 盒子还有**块级盒子(block box)**和**内联盒子(inline box)**之分。这两种盒子分别是**块级元素**与**内联元素**的默认盒状模型。
- ➔ 块级元素
  - ✓ div、table、form、fieldset、h1~h6、p、ol、ul、li、hr等。
- ➔ 内联元素
  - ✓ span、a、label、input、select、textarea、img、embed、br等。
- ➔ 块级元素显示为独立的一块（矩形区域），它的前后都会换行；
- ➔ 内联元素不会造成换行，它会和其他内联元素一起在一行内显示。
- ➔ 内联元素只能包括文本和其他内联元素；而块级元素则能包括内联元素和其他块级元素。
- ➔ 原则文档流

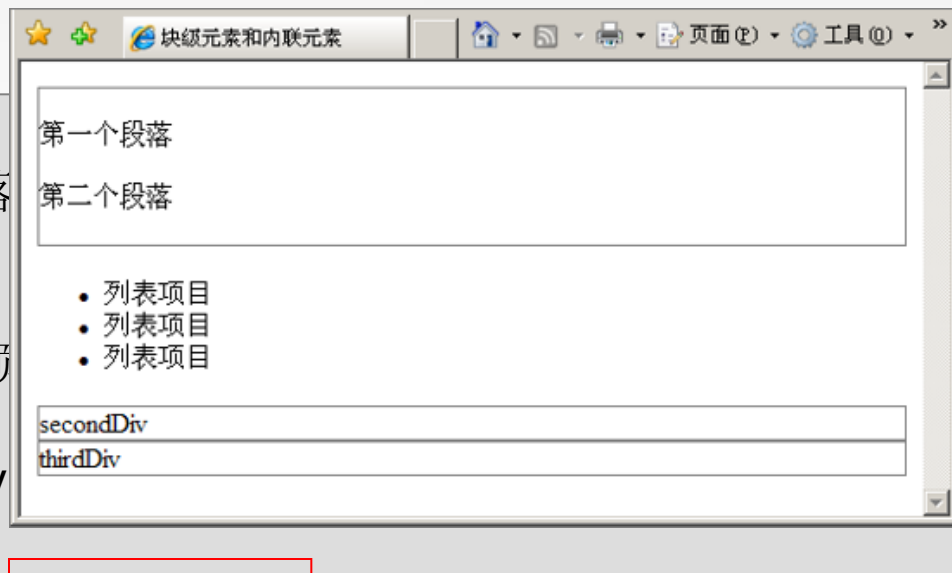


# 盒状模型的转换

→ CSS属性display能够设置网页元素使用的盒状模型。

- ✓ 为块级元素的display属性赋值为inline，那么这个块级元素转变为内联元素；
- ✓ 为内联元素的display属性赋值为block，则这个内联元素转变为块级元素。

```
<div id="firstDiv">  
    <p class="myp">第一种段落  
</div>  
<ul id="myul">  
    <li>列表项目</li> <li>列表项  
</ul>  
<div id="secondDiv">secondDiv</div>  
<div id="thirdDiv">thirdDiv</div>
```



演示 例7-1







# 盒状模型的转换

```
<style>
  #firstDiv, #secondDiv, #thirdDiv{
    border:1px solid gray;
  }
  #myul li{
    display:inline;
  }
  .myp{
    display:inline;
  }
  #secondDiv, #thirdDiv{
    display:inline;
  }
</style>
```



演示 例7-1





# 外边距

→ CSS 外边距属性用来设置一种元素所占空间的边沿到相邻元素之间的距离。使用 **margin** 属性来设置外边距。

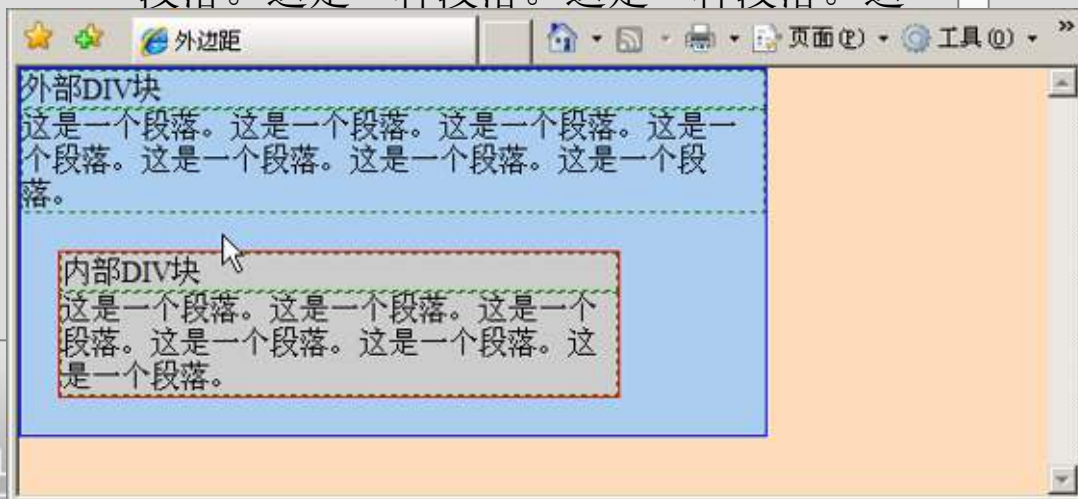
属性	意义	取值
<b>margin-top</b> <b>margin-right</b> <b>margin-bottom</b> <b>margin-left</b>	设置元素的顶外边距。 设置元素的右外边距。 设置元素的底外边距。 设置元素的左外边距。	可使用像素单位的数字或百分比，也可使用 <b>auto</b> 。
<b>margin</b>	简写属性。在一种申明中设置外边距属性。	依次设置顶外边距、右外边距、底外边距、左外边距。



# margin

```
<body>
  <div id="outerDiv">
    <p>外部DIV块</p>
    <p>这是一种段落。这是一种段落。这
      是一种段落。这是一种段落。这是一种段
      落。
    这是一种段落。这是一种段落。 </p>
    <div id="innerDiv">
      <p>内部DIV块</p>
      <p>这是一种段落。这是一种
        段落。这是一种段落。这是一种段落。这
```

```
<style type="text/css">
*{margin:0;padding:0;}
body{background-color:#fedcba;}
#outerDiv{
  border:1px solid blue;
  width:400px;
  margin:0px;
  background-color:#abcdef;
}
#innerDiv{
  border:1px solid red;
  width:300px;
  margin:20px;
  background-color:#cccccc;
border:1px dashed green;}
</style>
```



演示 例7-2



以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/488136064120006116>