

## HTML 初步

### 1 HTML 概念

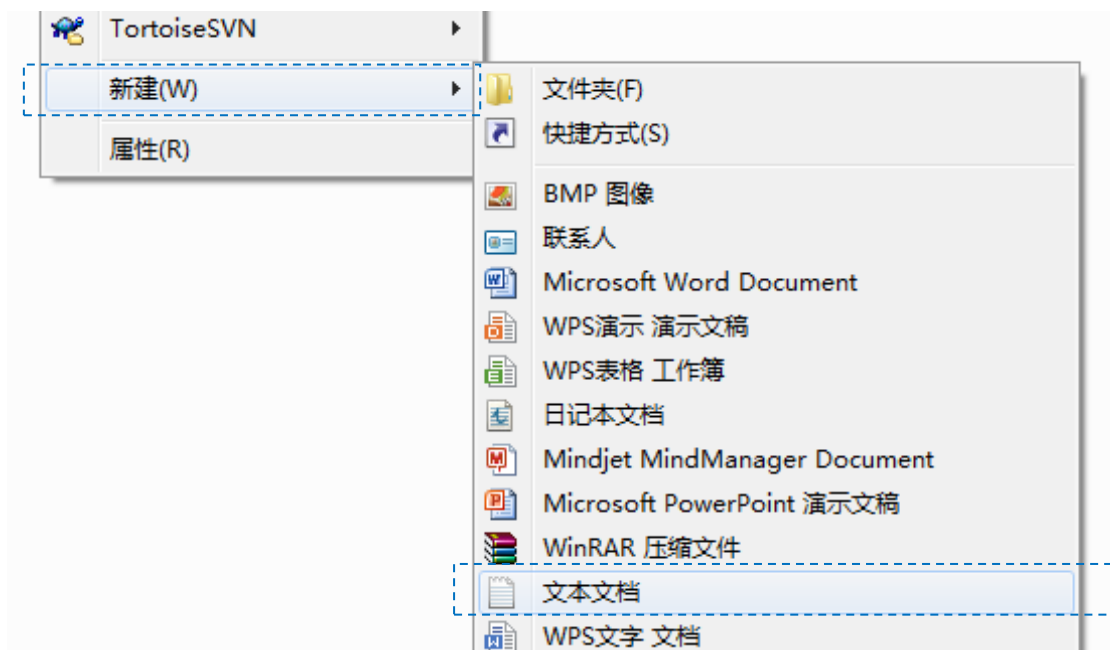
1.1 用来描述网页的一种语言

1.2 **Hyper Text Markup Language** 超文本标记语言

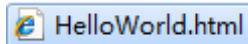
1.3 HTML 不是一种编程语言，而是一种标记语言 (markup language)。通常来说，至少要具备数据容器、流程控制、异常处理机制等特性才能够称为一门编程语言。标记语言的组成元素是一套标记标签 (markup tag)，HTML 使用 *标记标签* 来描述网页

### 2 第一个 HTML 文档：HelloWorld

2.1 新建一个纯文本文件



2.2 将扩展名修改为.html 或.htm——两种均可，但同一项目中一定要统一

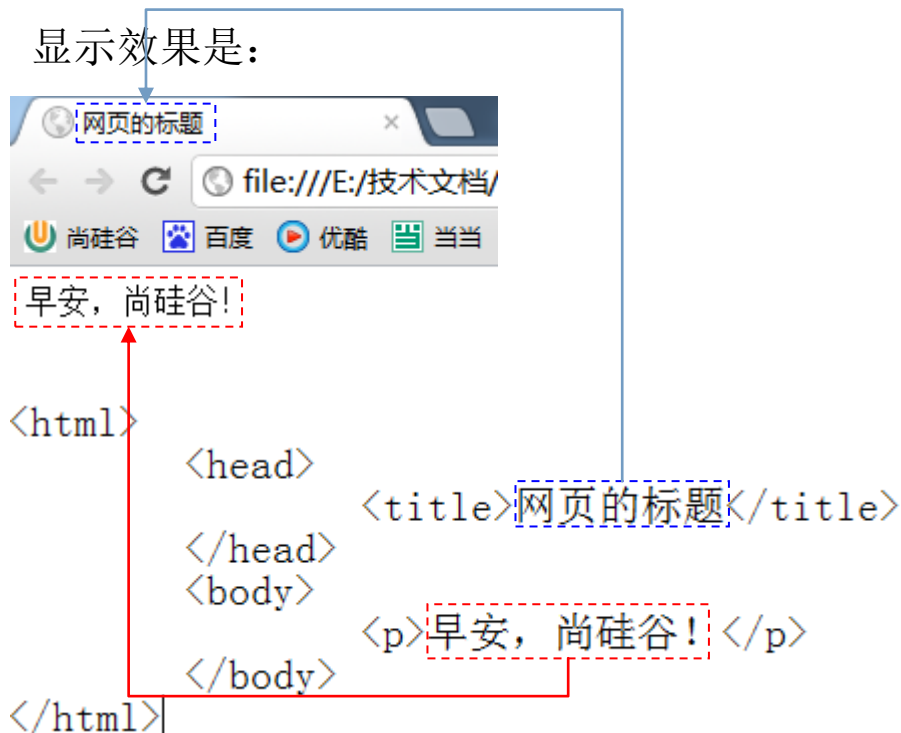


## 2.3 用记事本打开并编辑如下代码

```
HelloWorld.html - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
<html>
    <head>
        <title>网页的标题</title>
    </head>
    <body>
        <p>早安，尚硅谷！</p>
    </body>
</html>
```

## 2.4 在 IE（或 Chrome、Opera、Firefox 等浏览器）中打开，

显示效果是：



## 3 HTML 文档规范

### 3.1 格式良好的 HTML 文档标签必须正确关闭

正确关闭：<p>早安，尚硅谷！</p>

未正确关闭: `<p><a href="#">早安, 尚硅谷! </p>`

单标签也要正确关闭: `<br />`

单标签未正确关闭: `<br>`

### 3.2 格式良好的 HTML 文档标签不能交叉嵌套

```
<table border="0" cellspacing="5" cellpadding="5">
```

```
<tr>
  <td>Data</td>
</tr>
```

⇒ 正确嵌套

```
<tr><td>Data</tr></td>
```

⇒ 交叉嵌套

```
</table>
```

### 3.3 格式良好的 HTML 文档标签可以有属性，但属性必须有值，且属性值必须使用引号引起来，单引号、双引号均可，但推荐使用双引号

正确: `<input type="text" disabled="disabled" />`

错误: `<input type="text" disabled=disabled />`

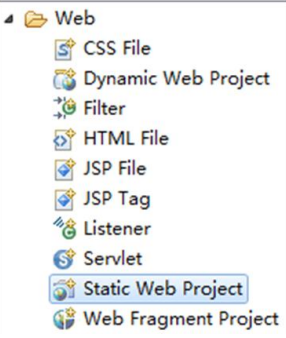
错误: `<input type="text" disabled />`

## 4 HTML 注释

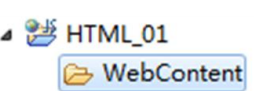
```
<!-- 这里是HTML注释，注释里的内容不会显示在页面上 -->
<p>早安, 尚硅谷! </p>
```

## 5 在 eclipse 中创建 HTML 文档

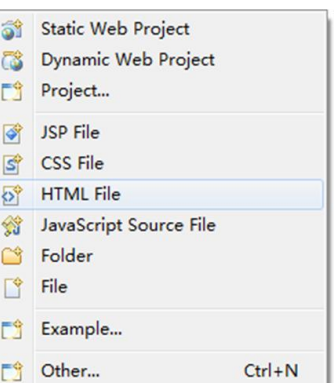
1. 创建静态Web工程  
Static Web Project



2. 定位到  
WebContent 目录



3. 新建HTML文件



3. 通常使用默认值：transitional，若使用框架则选择frameset

New HTML File (4.01 frameset)	html 4.01 frameset
New HTML File (4.01 strict)	html 4.01 strict
New HTML File (4.01 transitional)	html 4.01 transitional

## 6 HTML 文档的构成

```

1
2 <!DOCTYPE html PUBLIC          文档类型声明
3   "-//W3C//DTD HTML 4.01 Transitional//EN"
4   "http://www.w3.org/TR/html4/loose.dtd">
5
6 <html> 根标签
7
8   <head>
9     元信息
10    <meta http-equiv="Content-Type"
11      content="text/html; charset=UTF-8">
12
13    <title>Insert title here</title>
14    页面标题
15  </head>
16
17  <body>
18    <!-- 主体部分 -->
19  </body>
20
21 </html>
  
```

### 6.1 文档类型声明

6.1.1 <!DOCTYPE> 声明位于文档中的最前面的位置，处于 <html> 标签之前。此标签可告知浏览器文档使用哪种 HTML 或 XHTML 规范。

### 6.1.2 HTML 4.01 规定了三种文档类型：Strict、Transitional 以及 Frameset

**Strict:** 严格。如果您需要干净的标记，免于表现层的混乱，请使用此类型。请与层叠样式表（CSS）配合使用

**Transitional:** 过度。Transitional DTD 可包含 W3C 所期望移入样式表的呈现属性和元素。如果您的读者使用了不支持层叠样式表（CSS）的浏览器以至于您不得不使用 HTML 的呈现特性时，请使用此类型

**Frameset:** 框架。被用于带有框架的文档。除 frameset 元素取代了 body 元素之外，Frameset DTD 等同于 Transitional

### 6.1.3 <!DOCTYPE> 标签没有结束标签

## 6.2 页面元信息

6.2.1 <meta> 元素可提供有关页面的元信息（meta-information），比如针对搜索引擎和更新频度的描述和关键词。

6.2.2 <meta> 标签位于文档的头部，不包含任何内容。可以有多个

<meta> 标签的属性定义了与文档相关联的“名称/值对”。

由 name 或 http-equiv 属性定义“名称/值对”中的名称

由 content 属性定义“名称/值对”中的值

如：name=“keywords” content=“java 培训,北京 Java 培训,java 课程...” 设置关键字，使网页更容易被搜索引擎找到

如：name=“description” content=“尚硅谷 java 培训革命者，...” 网页描述

当服务器向浏览器发送文档时，会先发送许多名称/值对。虽然有些服务器会发送许多这种名称/值对，但是所有服务器都至少要发送一个：content-type:text/html。这将告诉浏览器准备接受一个 HTML 文档

### 6.2.3 HTML 与 XHTML 之间的差异

在 HTML 中，<meta> 标签没有结束标签。

在 XHTML 中，<meta> 标签必须被正确地关闭。

### 6.2.4 注意：

<meta> 标签永远位于 head 元素内部。

元数据总是以名称/值的形式被成对传递的。

## 7 制作表格

```
<table>
  { 定义表格的行
    <tr>
      <th>学号</th> <th>姓名</th> <th>科目</th> <th>成绩</th>
    </tr>
    <tr>
      <td>001</td> <td>王鹏</td> <td>高数</td> <td>97</td>
    </tr>
    <tr>
      <td>002</td> <td>许华</td> <td>C++</td> <td>94</td>
    </tr>
    <tr>
      <td>003</td> <td>赵菊</td> <td>离散</td> <td>96</td>
    </tr>
  }
</table>
```

表头

单元格，相当于表格的列

7.1 代码： </table>

效果:

学号	姓名	科目	成绩
001	王鹏	高数	97
002	许华	C++	94
003	赵菊	离散	96

→ 表头, 加粗显示  
→ tr 定义的行  
→ td 定义的单元格

## 7.2 显示表格线

```
<style type="text/css">
    table {
        background-color: black;
        border-spacing: 1px;
    }
    th,td {
        background-color: white;
    }
</style>
```

CSS 代码: `</style>`

学号	姓名	科目	成绩
001	王鹏	高数	97
002	许华	C++	94
003	赵菊	离散	96

效果:

## 7.3 合并单元格

### 跨行合并

```
<table>
<tr>
<th>学号</th><th>姓名</th><th>科目</th><th>成绩</th>
</tr>
<tr>
<td>001</td><td>王鹏</td><td>高数</td><td rowspan="2">97</td>
</tr>
<tr>
<td>002</td><td>许华</td><td>C++</td>
</tr>
<tr>
<td>003</td><td>赵菊</td><td>离散</td><td>96</td>
</tr>
</table>
```

→ 表示在垂直方向上跨两行  
→ 将被覆盖的单元格删掉

学号	姓名	科目	成绩
001	王鹏	高数	97
002	许华	C++	
003	赵菊	离散	96

## 跨列合并

```
<table>
  <tr>
    <th>学号</th><th>姓名</th><th>科目</th><th>成绩</th>
  </tr>
  <tr>
    <td>001</td><td>王鹏</td><td>高数</td><td>97</td>
  </tr>
  <tr>
    <td>002</td><td>许华</td><td>C++</td><td>96</td>
  </tr>
  <tr>
    <td>003</td><td>赵菊</td><td colspan="2">离散</td>
  </tr>
</table>
```

占据两个单元格的位置

学号	姓名	科目	成绩
001	王鹏	高数	97
002	许华	C++	96
003	赵菊	离散	

## 8 制作表单

### 8.1<form>…</form>标签

①action 属性：规定当提交表单时，向何处发送表单数据。

action 是一个必须有的属性，它的值是一个 URL

②method 属性：规定如何发送表单数据。

GET：规定以 GET 请求提交表单

POST：规定以 POST 请求提交表单

③enctype 属性：规定在发送到服务器之前应该如何对表单数据进行编码，不上传文件时不需要指定该属性

multipart/form-data：在上传文件时指定该值即可

### 8.2表单元素总览

HTML 标签	表单元素
---------	------



input	type 属性值	
	text	文本框
	password	密码框
	submit	提交按钮
	reset	重置按钮
	radio	单选按钮
	checkbox	多选框
	button	普通按钮
	file	文件上传框
	hidden	表单隐藏域
textarea		多行的文本域
select + option		下拉列表

### 8.3 文本框

账号 `<input type="text" name="account" value="请输入账号" />`

账号

提交文本框内容时的键      文本框默认值

### 8.4 密码框

密码: `<input type="password" name="pwd" />`

密码:

提交密码框内容时的键



## 8.5 提交按钮

```
<input type="submit" value="在按钮上显示的内容" />
```

在按钮上显示的内容

## 8.6 重置按钮

```
<input type="reset" value="点我重置">
```

点我重置

## 8.7 单选按钮

要实现“一组值中只能选一个”的效果，必须设定相同的

name 属性值

```
<!-- 单选按钮 -->
季节:
<input type="radio" name="season" value="spring" />春天
<input type="radio" name="season" value="summer" checked="checked" />夏天
<input type="radio" name="season" value="autumn" />秋天
<input type="radio" name="season" value="winter" />冬天
```

一组值      被选中后提交的值      将当前单选按钮设置为默认被选中

季节: ☐ 春天 ☒ 夏天 ☐ 秋天 ☐ 冬天

## 8.8 多选框

```
<!-- 多选框 -->
爱好:
<input type="checkbox" name="hobby" value="reading" />读书
<input type="checkbox" name="hobby" value="swimming" checked="checked" />游泳
<input type="checkbox" name="hobby" value="shopping" />购物
<input type="checkbox" name="hobby" value="sleeping" checked="checked" />睡觉
```

爱好: ☐ 读书 ☒ 游泳 ☐ 购物 ☒ 睡觉

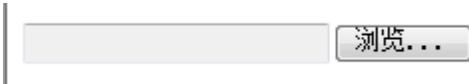
## 8.9 普通按钮

```
<!-- 普通按钮 -->
<input type="button" value="普通按钮" />
```

普通按钮

## 8.10 文件上传框

```
<!-- 文件上传框 -->  
<input type="file" name="uploadFile" />
```



## 8.11 表单隐藏域

表单隐藏域在页面上没有任何显示

表单隐藏域的作用是将一些不需要或不希望用户看到的信息随表单一起提交

<code>&lt;input type="hidden"</code>	<code>name="hiddenField"</code>	<code>value="hiddendValue" /&gt;</code>
↓ 定义一个表单隐藏域	↓ 通过name属性的值来引用当前隐藏域的值	↓ 当前隐藏域的值

## 8.12 多行文本框

多行文本域是用来在页面上输入大段文本的控件

在表单中多行文本域是用 `textarea` 标签定义的

**name 属性：**提交表单后通过 `name` 属性引用多行文本域中的值

**rows 属性：**多行文本域控件显示多少行（高度）

**cols 属性：**多行文本域控件显示多少列（宽度）

多行文本域没有 `value` 属性

```
<textarea name="textareaName" rows="8" cols="40">请在这里输入文本</textarea>
```



## 8.13 下拉列表

在表单中，下拉列表由 select 和 option 标签共同组成

请选择一种你最喜欢吃的水果：

```

<select name="fruit">
  <option value="apple">苹果</option>
  <option value="banana">香蕉</option>
  <option value="orange" selected="selected">橘子</option>
  <option value="peach">桃子</option>
  <option value="pear">梨</option>
</select>

```

通过name属性引用value的值

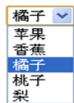
默认被选中

被选中的值的value属性将被提交到服务器端

在页面上列表中显示的内容

实际的显示效果

请选择一种你最喜欢吃的水果：



## 9 超链接

```

<a href="targetPage.html">这是一个超链接</a>

```

指向



This is targetPage.html

显示效果

这是一个超链接

点击

HTML 文档中涉及到的路径问题

- 绝对路径：文件在磁盘上存储的实际物理地址
  - D:\DevWor\...\ML\WebContent\page\target.html
- 相对路径
  - 以自身为基准
    - “./page/target.html” 表示当前目录下的page目录下的target.html文件，  
“ ./” 可以省略不写
    - “../page/target.html” 表示上一级目录下的page目录下的target.html文件
  - 以Web应用的根目录为基准
    - 需动态获取Web应用的根目录的路径

## 10 插入图片

在 HTML 中，图像由 <img> 标签定义

src 属性：指定图像的 URL 地址

alt 属性：为图像定义一串预备的可替换的文本，当图像无法显示时，显示图像的位置可以显示这段文本

下面是插入网页的图片：<br />  


下面是插入网页的图片：



## 11 列表

- 无序列表：无序列表是一个项目的列表，此列项目使用粗体圆点（典型的小黑圆圈）进行标记

```
<ul>
  <li>
    Coffee
  </li>
  <li>
    Milk
  </li>
</ul>
```

- Coffee
- Milk

- 有序列表：有序列表也是一列项目，列表项目使用数字进行标记

```
<ol>
  <li>
    Coffee
  </li>
  <li>
    Milk
  </li>
</ol>
```

1. Coffee
2. Milk

## 12 标题

标题（Heading）是通过 <h1> - <h6> 等标签进行定义的

<h1> 定义最大的标题。<h6>定义最小的标题

不要试图用标题标签来控制字体大小

## 实例

```
<h1>This is a heading</h1>
<h2>This is a heading</h2>
<h3>This is a heading</h3>
```

**This is a heading**

**This is a heading**

**This is a heading**

## 13 段落

段落是通过 `<p>` 标题定义的

`<p></p>` 标签自带换行效果，但单纯换行的话建议使用 `<br />` 标签，更专业的做法是使用 CSS 对元素进行合理布局

## 实例

```
<p>This is a paragraph</p>
<p>This is another paragraph</p>
```

This is a paragraph










This is another paragraph

## 14 HTML 颜色

颜色由一个十六进制符号来定义，这个符号由红色、绿色和蓝色的值组成（RGB）。

每种颜色的最小值是 0（十六进制：#00）。最大值是 255（十六进制：#FF）。

这个表格给出了由三种颜色混合而成的具体效果

Color	Color HEX	Color RGB
	#000000	rgb(0,0,0)
	#FF0000	rgb(255,0,0)
	#00FF00	rgb(0,255,0)
	#0000FF	rgb(0,0,255)
	#FFFF00	rgb(255,255,0)
	#00FFFF	rgb(0,255,255)
	#FF00FF	rgb(255,0,255)
	#C0C0C0	rgb(192,192,192)
	#FFFFFF	rgb(255,255,255)

## 15 框架

框架的用途是在一个窗口中，组织多个 HTML 文档

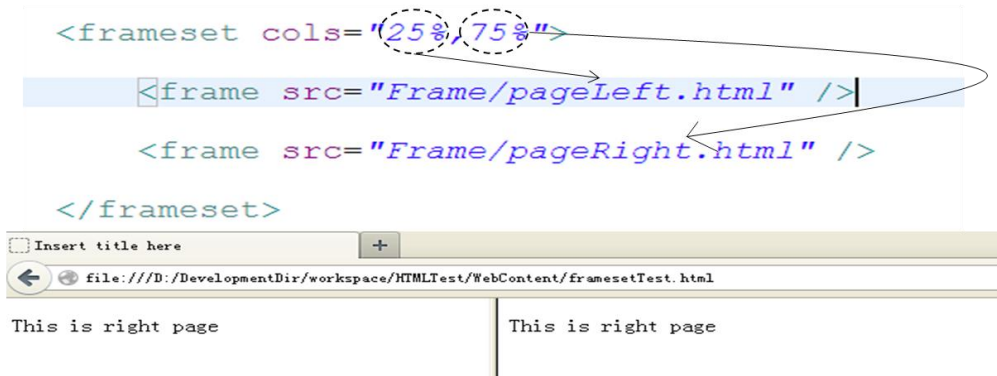


框架结构标签（<frameset>）

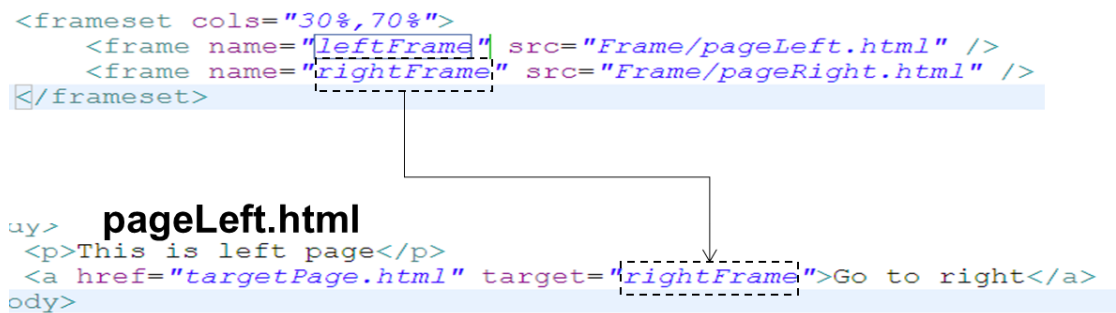
- 用于组织框架结构
- cols属性：“列”，控制框架有多少列以及分配比例
  - cols= “25%,25%,50% ” 表示将窗口分成3列，宽度分别占窗口总宽度的25%、25%、50%
- rows属性：“行”，控制框架有多少行以及分配比例
  - rows= “15%,15%,70% ” 表示将窗口分成3行，高度分别占窗口总高度的15%、15%、70%
- 框架是可以嵌套的，所以尽量不要在同一个frameset标签中同时指定cols属性和rows属性
  - 在HTML文档中，frameset标签会取代body标签

框架标签（Frame）

- 指定要加入框架的HTML页面
- src属性：指定HTML文件的URL

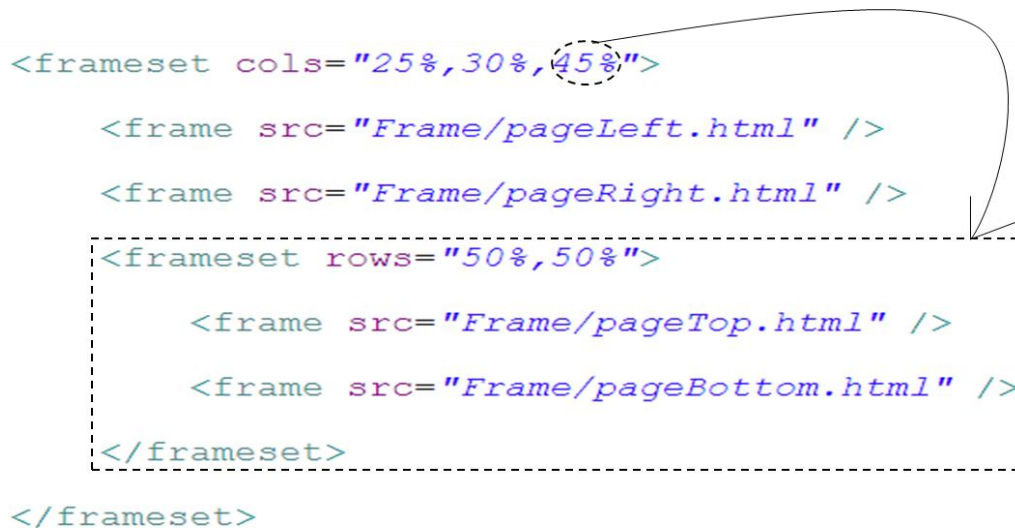


框架之间页面的互相访问

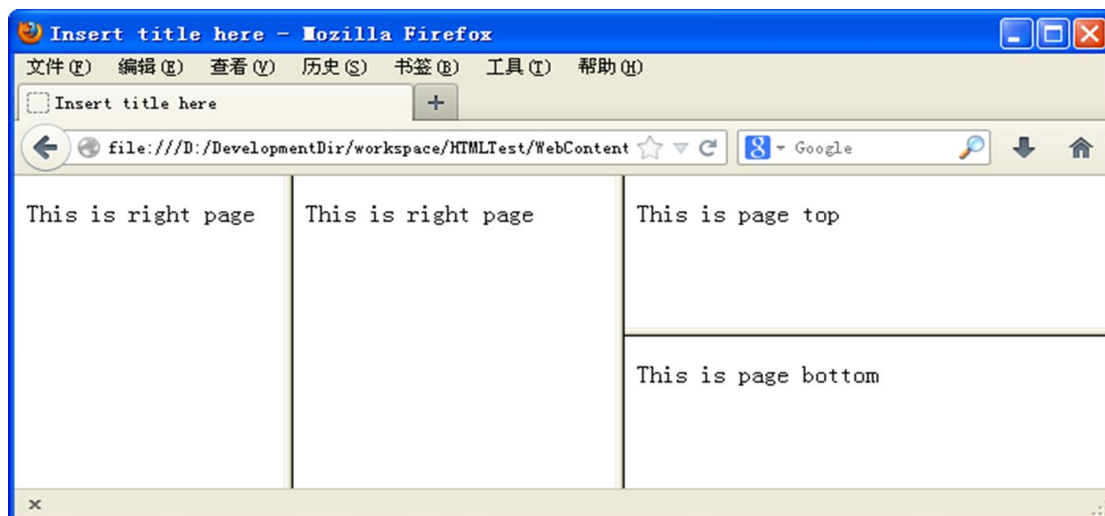


## 让targetPage.html在rightFrame中显示

框架的嵌套







不能使用框架的情况

- 部分浏览器不支持框架
- noframes标签：当浏览器不支持框架时，<frameset> </frameset> 标签内的内容将会被忽略，而显示<noframes> </noframes>标签内的内容。
- 通常在<noframes> </noframes>标签内加入body标签

```
<frameset cols="25%,30%,45%">
  <frame src="Frame/pageLeft.html" />
  <frame src="Frame/pageRight.html" />
</frameset>
<noframes>
  <body>您的浏览器不支持框架! </body>
</noframes>
```