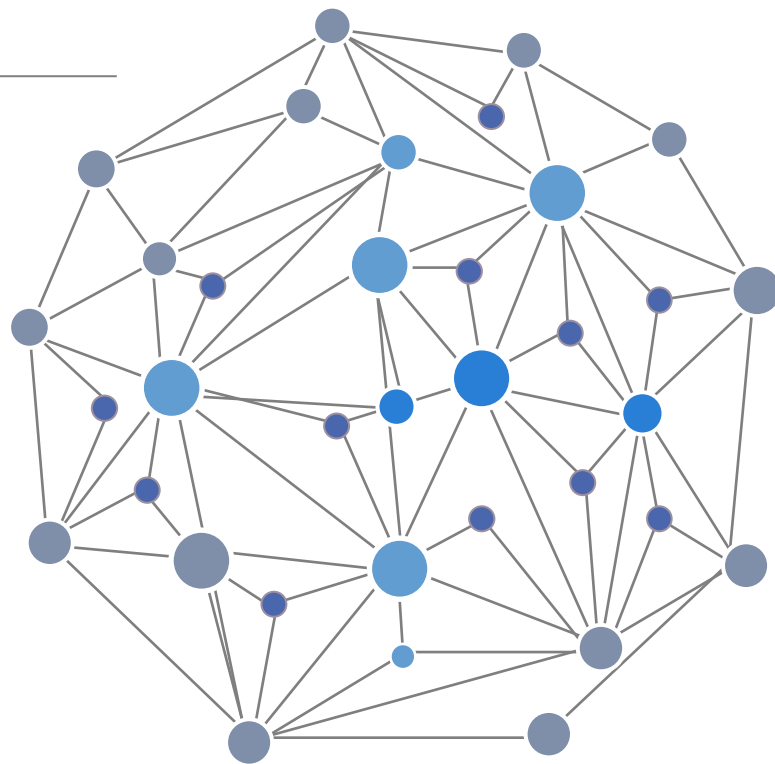

Web 程序设计

第四讲 XHTML 与 HTML5

福州大学 计算机与大数据学院

软件工程系 陈昱

2022/10/11



XHTML 与 HTML5

- 什么是 XHTML 和 HTML5
- XHTML 和 HTML5 文件结构
- HTML 常用元素
 - 头元素中的标记
 - 文档内容基本标记
 - 列表
 - 链接
 - 图像
 - 表格
 - 表单

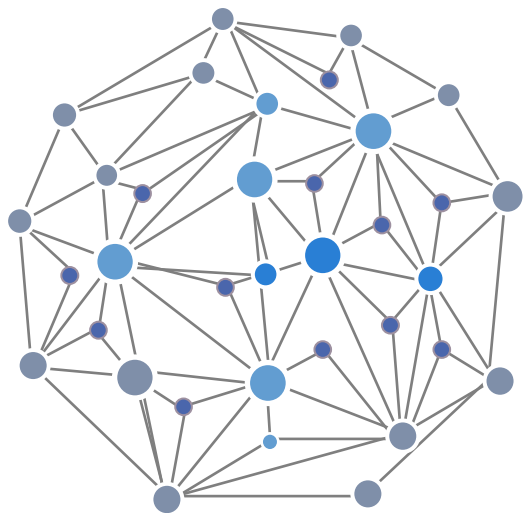
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
  <head><title>XYZ</title>
  </head>
  <body>
    <p>
      voluptatem accusantium do-
      totam rem aperiam eaque
    </p>
  </body>
</html>
```

XHTML

XHTML 与 HTML5

- HTML 常用属性
- div 和 span 标记
- 块元素和内联元素
- 将 HTML4 转换为 XHTML
- 文本格式化标记
- 框架





什么是 XHTML 和 HTML5?

什么是 HTML?

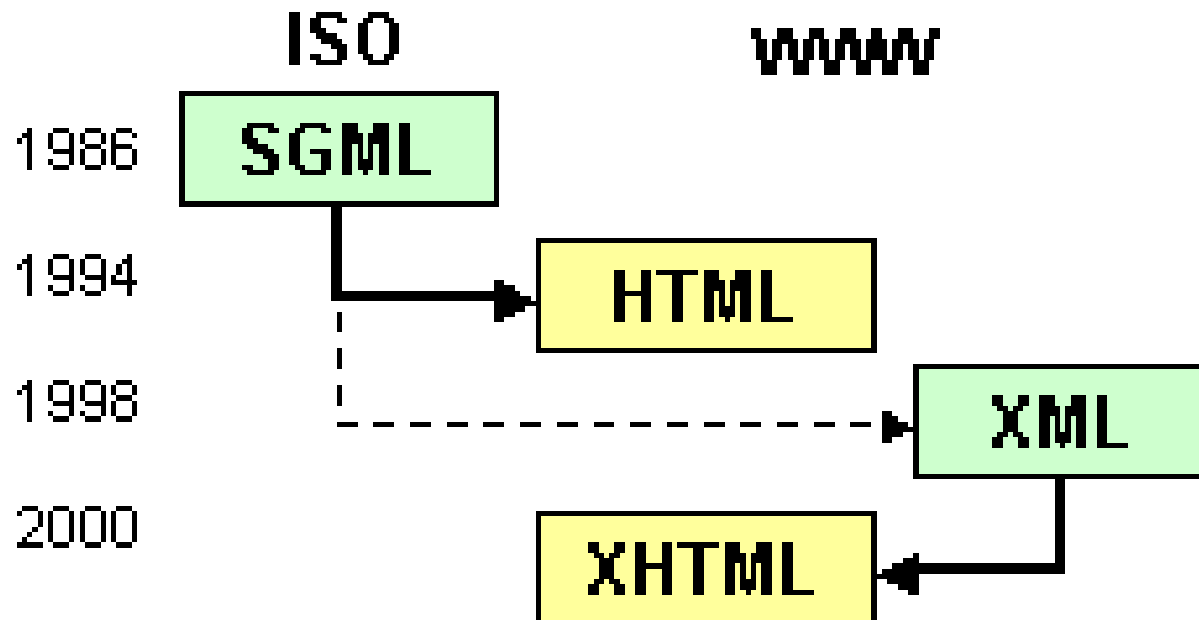
- HTML 也是源于 SGML 的一种置标语言
 - 它定义了一系列标记，每个标记表明了**一定的**（语义或是表现）**信息**
 - 标记易学易记
 - 使用广泛
 - **正式版本：HTML 4.01(1999), HTML 5.2 (2017)**
- 与 XML 不同在于，HTML 不是一种元语言，不能自定义标记，不能定义其他置标语言

什么是 XHTML?

- XHTML 是 The Extensible HyperText Markup Language 可扩展超文本置标语言的缩写
- XML 虽然数据描述能力强大，完全可以替代 HTML，但面对难以计数的现有网站，全面改用 XML 阻力过大

什么是 XHTML?

- 因此, W3C 在 HTML 4.01 的基础上, 用 XML 的语法规则对 HTML 重新定义, 就得到了 XHTML



XHTML 的思想

- 用 XML 的语法对 HTML4 进行改写：
 - 利用 DTD 对语法进行规定
 - 标记名称统一用小写
 - 标记属性名称统一用小写
 - 属性值用双引号括起来
 - 标记必须配对
 - 一些标记没有结束标记的话必须带斜杠自关闭
 -
- 具有严格的 XML 语法，更规范，易于学习和上手

注意事项

- 所谓 XHTML 就是用 XML 重新定义的 HTML 语言
 - 用原来 HTML 的标记名作为 XML 里的标记
- XHTML 还是 XML!
 - 一个 XHTML 文档也是有效的 XML 文档
 - 只不过长得很像 HTML~

注意事项

- XHTML 的作用在于定义文档的整体结构，而不是定义显示细节
 - 它是结构层技术
 - 因此你写的 XHTML 文件可能在浏览器里显示器来起来很简洁
- 建立 XHTML 的目的就是实现 HTML 向 XML 的平稳过渡

XHTML 的版本

- XHTML 1.0 (2000)
 - <http://www.w3.org/TR/xhtml1>
- XHTML 1.1 (2001)
 - 兼容 1.0, 小幅改变
- 目前推荐遵循的是 W3C 正式通过的 XHTML 1.0 以及 XHTML 1.1 标准

XHTML 的版本

- XHTML 2 (Draft)
 - 语法不兼容 XHTML 1.x 和 HTML, 改动较大
 - W3C 比较激进的改革尝试, 事实证明接受度太低。W3C 决定集中力量发展 HTML 5, 2010 年底工作组解散 (2010-12-17: The XHTML2 Working Group is closed.)

新一代的标记语言

- HTML 5
 - 2004 年由 WHATWG 提出，于 2007 年被 W3C 接纳，并成立了新的 HTML 工作团队
 - HTML 5 支持传统的 HTML 语法，也支持 XML 的语法
 - 使用 XML 语法书写时也称为：XHTML5

漫画：混乱的标记语言

XHTML2/HTML5

<http://www.cnblogs.com/JustinYoung/archive/2009/07/30/misunderstanding-markup-xhtml-2-comic-strip.html>

- XHTML 和 HTML5 才是王道 ~



编写工具

- (X)HTML 文档是纯文本文件，可以使用任意文本编辑器编写
- 后缀通常写为 **.html(建议)** 或 **.htm**，少数用 **.xhtml**
- VSCode, ATOM, Sublime Text ...
- Visual Studio, Eclipse ...
 - 支持 Web 工程的 IDE 一般都支持 (X)HTML 的编写

XHTML 和 HTML5 文件的 基本结构

XHTML 文件的基本组成

- 标记
 - `<tag>`
- 空标记 (自关闭)
 - `<EmptyTag />`, 比如: `
`
- 属性
 - `<tag attribute="value">Hello</tag>`
- 元素
 - `<tag>Content</tag>`

XHTML 文件的基本结构

- XML 声明
- 文档类型声明
- 文件体标记结构：

根元素 html

– 头元素 head

- 标题元素 title

– 主体元素 body

XHTML 文件的基本结构

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!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" xml:lang="en">
```

```
  <head>
```

```
    <title>Hello, World!</title>
```

```
    .....
```

```
  </head>
```

```
  <body>
```

```
    .....
```

```
  </body>
```

```
</html>
```

XML 声明

- 虽然看起来加上 XML 声明是个比较好的做法，但由于一些浏览器软件的不完善，**建议不要加上 XML 声明**
- 主要是 IE6，加上 XML 声明，浏览器会采用 Quirks Mode 工作 (非正常工作模式)，造成页面显示问题
 - www.aoao.org.cn/blog/2007/01/browser-mode/
 - http://en.wikipedia.org/wiki/Quirks_mode

文档类型声明

- DOCTYPE 声明必须放在每一个 XHTML 文档顶部，在所有数据和标记之上，在 XML 声明之下
- XHTML 文件的 DOCTYPE 声明是必须的！
- 回顾引用公共 DTD 的形式：

<!DOCTYPE 根元素

PUBLIC "DTD 名称" "外部 DTD 的URL">

XHTML 1.0 的三种文档类型

- XHTML 1.0 提供了三种 DTD 声明可供选择:
- Transitional
 - 要求比较宽松的 DTD, 它允许你继续使用 HTML 4.01 中的标记 (但是要符合 XHTML 的写法)
- Frameset
 - 专门针对框架页面设计使用的 DTD (不推荐使用)
- Strict
 - 要求严格的 DTD, 不能使用任何表示信息样式的标识和属性, 如居中 <center>

XHTML 1.0 Transitional

- 方便原来使用 HTML 的设计师过渡到 XHTML
- 可以方便地将原先的 HTML 文档转换成符合 XHTML 的文档
- 对于我们新时代的开发者的，基本可以忘掉

```
<!DOCTYPE html PUBLIC
```

```
"-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

XHTML 1.0 Frameset

- 文档中要用到框架 (Frameset) 结构时使用
- 现代网页设计不推荐使用框架!
 - 在移动设备上用户体验不好
- 也可以直接忘掉 ~

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


XHTML 1.0 Strict

- 与 XHTML 1.1 非常接近，推荐使用
- 要求严格，不能使用任何 `<center>` 这样的标识和属性

```
<!DOCTYPE html PUBLIC
```

```
"-//W3C//DTD XHTML 1.0 Strict//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

根元素 <html> </html>

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

- **语言声明:** **xml:lang="value"**
- 易用性 (accessibility) 方面的考虑
- 值由语言类型和方言种类构成, 比如:
 - en, fr, de, en-US, en-GB
- 中文有多种表示法: zh-CN/zh-TW,
zh-Hans/zh-Hant

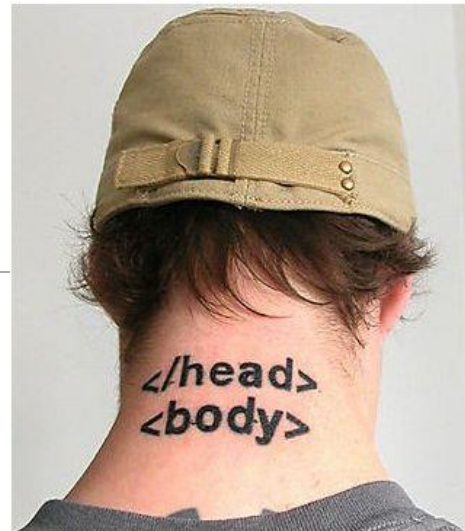
<html> 的两个子元素

- 头元素

- <head> </head>
- 出现在主体元素之前，包含将在主体元素信息之前装载的信息

- 主体元素

- <body> </body>
- 是这个文档展示在浏览器上的主要内容



HTML5 的文档结构

```
<!DOCTYPE html>
<!-- greet.html
      A trivial document
-->
<html lang = "en">
  <head>
    <title> Our first document </title>
    <meta charset = "utf-8" />
  </head>
  <body>
    <p>
      Greetings from your Webmaster!
    </p>
  </body>
</html>
```

采用 XHTML 语法书写 HTML5

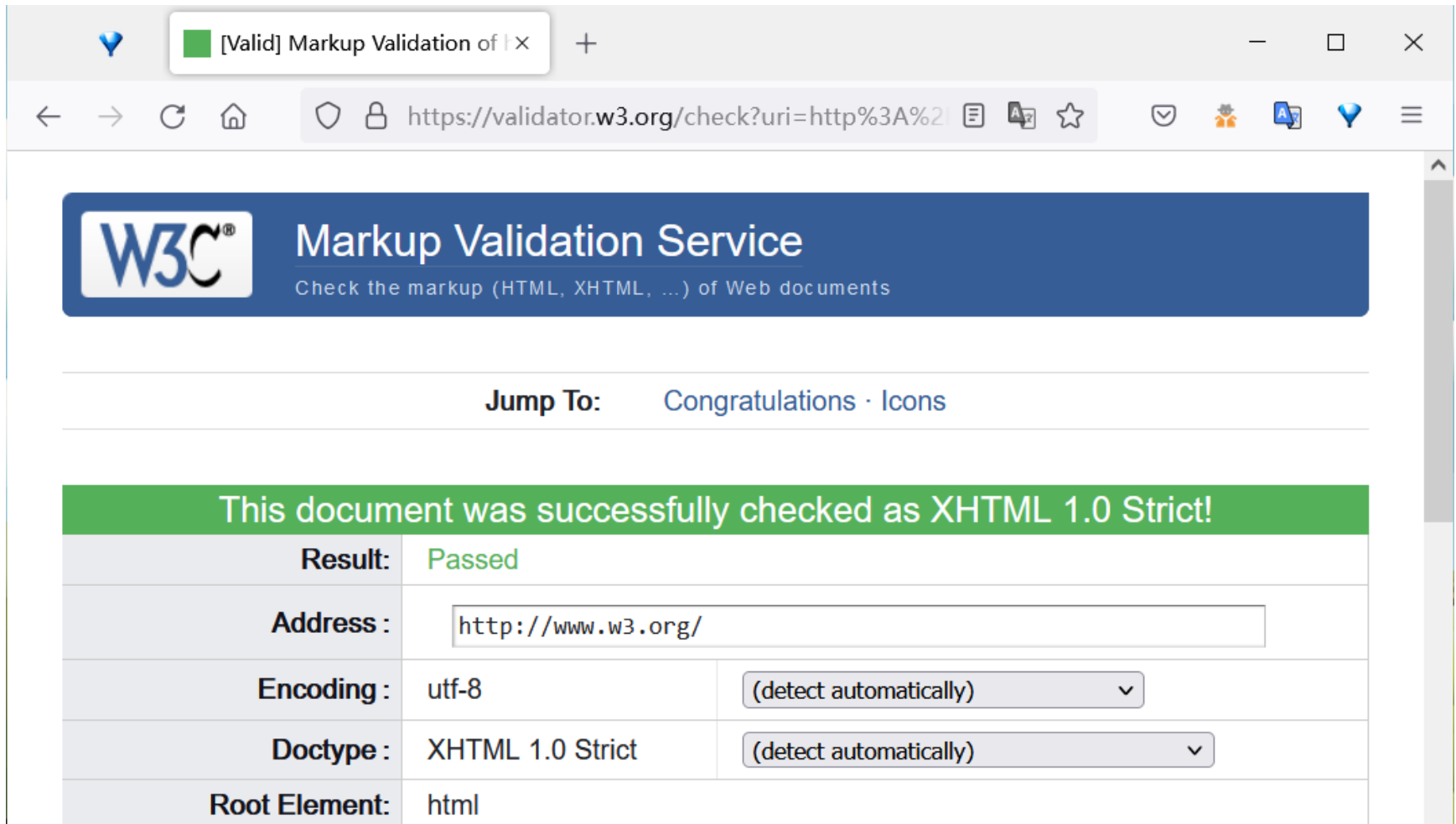
- 采用 XHTML 语法的原因：
 - HTML 的语法规则宽松，导致马虎的甚至是有歧义文档；XHTML 语法更加严格，产生更加清晰和标准的文档结构
 - HTML 处理器对语法的要求比 HTML 规范更宽
 - XHTML 的语法正确性容易被验证
- 我们后续介绍的内容既适用于 XHTML，又适用于 HTML5（我们统称为 HTML）

W3C Validation Service

- 验证一个文档的结构是否符合规范
 - <http://validator.w3.org>
- 可以
 - 指定一个URL来验证
 - 上传一个文件验证

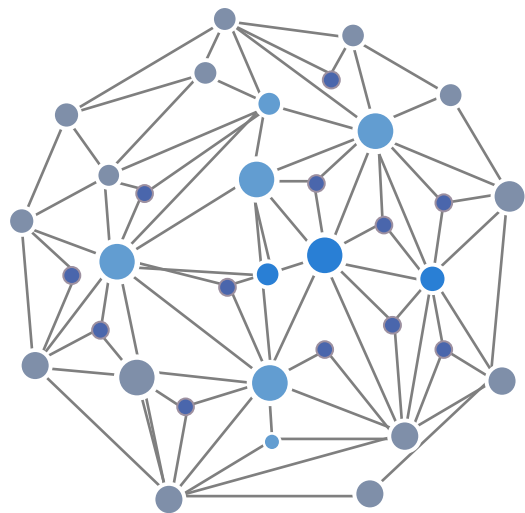
W3C Validation Service

- <http://validator.w3.org/>



The screenshot shows the W3C Markup Validation Service interface. The browser's address bar displays the URL `https://validator.w3.org/check?uri=http%3A%2F%2Fwww.w3.org%2F`. The page header features the W3C logo and the text "Markup Validation Service" with a subtitle "Check the markup (HTML, XHTML, ...) of Web documents". Below the header, there are links for "Jump To: Congratulations · Icons". A green banner states "This document was successfully checked as XHTML 1.0 Strict!". A table below provides details of the validation:

Result:	Passed	
Address :	<input type="text" value="http://www.w3.org/"/>	
Encoding :	utf-8	(detect automatically) ▾
Doctype :	XHTML 1.0 Strict	(detect automatically) ▾
Root Element:	html	



HTML 中的常用标记

HTML 中的常用标记

- 头元素中的标记
- 文档内容基本标记
- 列表
- 链接
- 图像
- 表格
- 表单

头元素中的标记

- `<title>` 文档标题
- `<meta>` 定义关于 HTML 文档的元数据
- `<base>` 定义页面上所有链接的默认基地址
- `<link>` 定义文档与外部资源之间的关系
- `<script>` 定义客户端脚本, 如 Javascript
- `<style>` 定义文档的样式信息, 如 CSS

头元素例子

```
<head>  
  <title>Hello World</title>  
  <meta name="author" content="my name" />  
  <meta name="description"  
    content="It is a sample for my course." />  
</head>
```

<meta> 元标记

- 站点的简要介绍

<meta name="description" content="..." />

- 站点的关键词

<meta name="keywords" content="..." />

- 设置站点作者信息

<meta name="author" content="..." />

- generator, revised, copyright ...

定义语言编码

- 在没有XML声明下的折衷方案

```
<meta http-equiv="Content-Type"  
      content="text/html; charset=utf-8" />
```

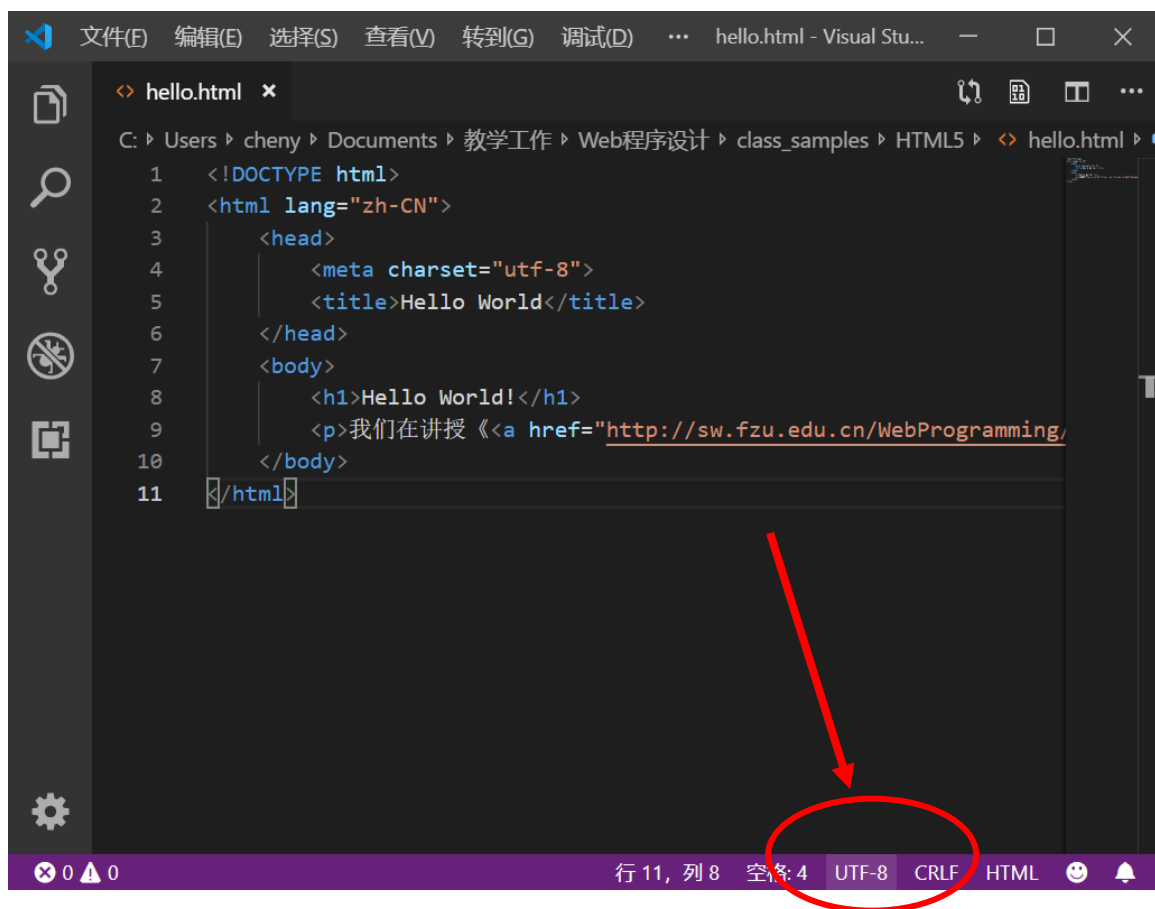
- HTML5 下还可以写成

```
<meta charset="utf-8" />
```

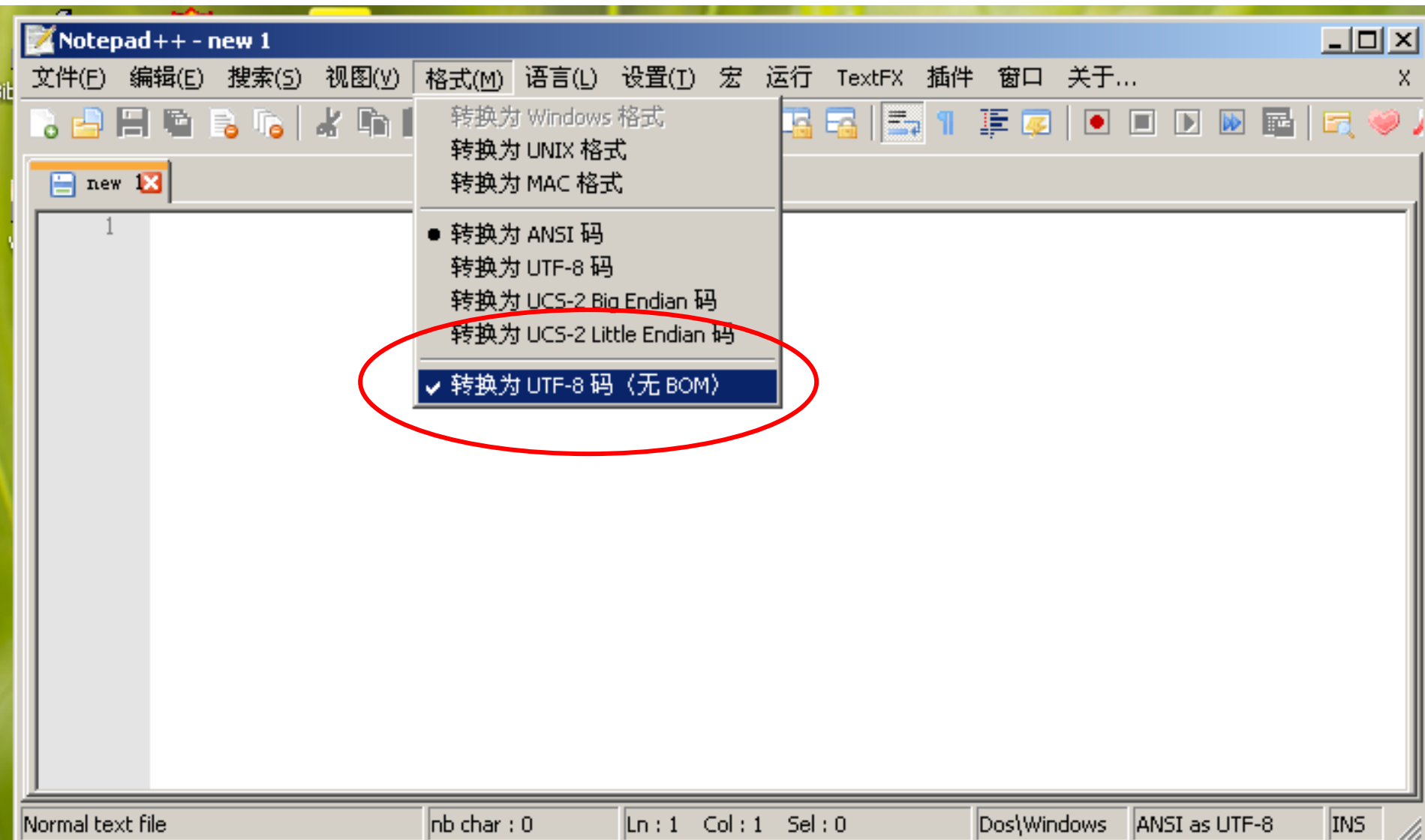
- 注意编辑器的保存格式要与你写的charset一致

编辑器中的编码选择

- VSCode 为例



UTF-8(无BOM, Byte Order Mask)



基本标记 Basic Tags

- `<h1>` to `<h6>`: 标题一到标题六
- `<p>`: 段落
- `
`: 换行
- `<hr />`: 水平线

[View Sample](#)

列表 Lists

- HTML 提供了三种类型的列表:
 - 编号的 (有序的)
 - 有项目符号的 (无序的)
 - 定义列表

List Type	Element	Item Element
Ordered List	<code>...</code>	<code>...</code>
Unordered List	<code>...</code>	<code>...</code>
Definition List	<code><dl>...</dl> <dt>...</dt>, <dd>...</dd></code>	

有序列表

- 有序列表以标记表示，每个表项用标记
- 浏览器会给每个表项编上序号

```
<p>The Lord of the Rings Trilogy</p>
<ol>
  <li>The Fellowship of the Rings</li>
  <li>The Two Towers</li>
  <li>The Return of the King</li>
</ol>
```

无序列表

- 无序列表以标记表示，每个表项用标记
- 浏览器会给每个表项加上项目符号（比如方块）

```
<p>College Sports</p>  
<ul>  
  <li>Soccer</li>  
  <li>Basketball</li>  
  <li>Shuttlecock</li>  
</ul>
```

定义列表

- 用于建立术语表
- 用<dl>元素建立列表，表内用dt元素标记术语，用dd元素标记术语的解释，**一个dt可以跟随多个dd**

```
<dl>
  <dt>Term 1</dt>
  <dd>Definition 1</dd>
  <dt>Term 2</dt>
  <dd>Definition 2-1</dd>
  <dd>Definition 2-2</dd>
</dl>
```

列表嵌套 Nesting Lists

- 列表可以任意嵌套，各种类型的列表可以互相嵌套

```
<ul>
  <li> Favorite Football teams
    <ul>
      <li> New England Patriots</li>
      <li> Tennessee Titans</li>
      <li> Dallas Cowboys</li>
    </ul>
  </li>
  <li> Favorite Players
    <ol>
      <li> Tom Brady - Patriots</li>
      <li> Steve McNair - Titans</li>
      <li> Adam Vinitari - Patriots</li>
    </ol>
  </li>
</ul>
```

超链接 Hypertext Link

- 锚标记 `<a>` 用来定义一个超链接
- 除了被包含的文本外，还需要为锚标记增添链接目标
- 链接的目标用 `href` 属性来定义

```
<a href="http://www.fzu.edu.cn">福州大学</a>
```

绝对与相对链接

- 链接可以是绝对的，比如
`http://www.fzu.edu.cn`
- 也可以是相对的，即链接到当前服务器的其他文档

```
<a href=" ../flyingmoss.html" >  
    The miracle of moss in flight  
</a>
```

- 链接不仅可以连接到其他的HTML文档，也可以连接到其他类型文件

内部链接

- 链接还可以跳转到页面的其他部分
- 可以给几乎所有的标记增加id属性，比如 `<h2 id="moss">Moss</h2>`，然后跳转到它

```
<h2 id="moss">Moss</h2>
```

.....

```
<a href="#moss">Go to moss</a>
```


图像 Image

- 图像标记 **img** 用来为HTML添加图像

```

```

- **src 属性用来指定图片的位置**
- **alt 属性用来指示替代文本（注意！在 XHTML 规范中，这是必不可少的）**
- 此外，还有 width, height 等很多属性，查阅 HTML 参考手册

表格 Table

- **table** 元素定义标记
- **tr** 元素定义表格中的一行
- **td** 元素定义数据**单元格**

```
<table>
  <tr>
    <td>Row 1, cell 1</td>
    <td>Row 1, cell 2</td>
  </tr>
  <tr>
    <td>Row 2, cell 1</td>
    <td>Row 2, cell 2</td>
  </tr>
</table>
```

表头 Table Heading

- **th** 标记是单元格表头，像 td 标记一样，th 标记也要用在 tr 标记里边。

```
<table>  
  <tr>  
    <th>Column 1 heading</th>  
    <th>Column 2 heading</th>  
    <th>Column 3 heading</th>  
  </tr>
```

单元格跨越 Cell Span

- **td** 标记的两个属性可以实现单元格跨越
- **colspan** 属性，意即“列跨越”，在列的方向上将会跨越指定数目的单元格
- **rowspan** 属性，意即“行跨越”

`<td colspan="2">`

Row 2, cell 2, also spanning Row 2, cell 3

`</td>`

.....

`<td rowspan="2">`

Row 3, cell 1, also spanning Row 4, cell 1

`</td>`

表单 **Form**

- 表单可以用来在网页中发送数据
- 表单本身是没有什么用的，需要编一个程序来处理输入表单中的数据
- 实际用在 XHTML 中的表单标记有 **form、input、textarea、select 和 option**

form 表单标记

- 表单标记 form 定义的表单里面
 - 行为属性 **action** 告诉表单当提交的时候将内容发往何处
 - 可选的方法属性 **method** 告诉表单数据将怎样发送，有 **get** 和 **post** 两个值

```
<form action="processing.php" method="post">
```

```
.....
```

```
</form>
```

input 标记

- 文本框 `text`
- 密码框 `password`
- 提交按钮 `submit` 与重置按钮 `reset`
- 隐含框 `hidden`
- 文件选择输入框 `file`
- 单选框 `radio`
- 复选框 `checkbox`
- 图像按钮 `image`
- 按钮 `button`

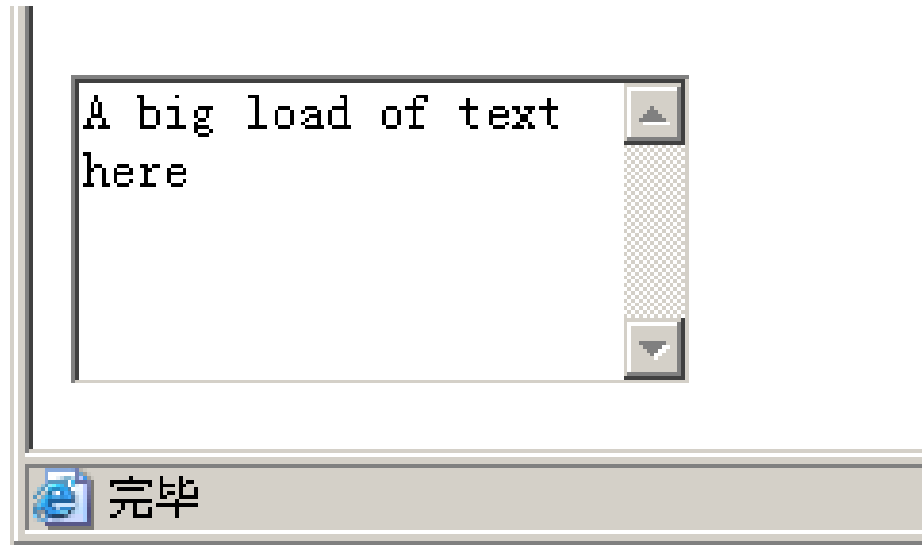
HTML5 新的 Input 类型

- HTML5 拥有多种新的表单输入类型，这些新功能提供了更好的输入控制和验证。
 - email
 - url
 - number
 - range
 - date pickers (date, month, week, time, datetime, datetime-local)
 - search
 - color

textarea

- textarea是一个比较大的文本框
- 它必须有行属性rows和列属性cols

```
<textarea rows="5" cols="20">A big load of text here</textarea>
```



select 和 option

- 选定标记 select 与选项标记 option 一起可以制作一个下拉选框
- size 属性可以改变下拉菜单项的显示个数

```
<select name="Technologies">  
  <option label="ASP.NET">ASP.NET</option>  
  <option label="C#">C#</option>  
  <option label="VB.NET">VB.NET</option>  
  <option label="XPath">XPath</option>  
  <option label="XSL">XSL</option>  
  <option label="XQuery">XQuery</option>  
</select>
```

fieldset, legend, label

- <fieldset> 元素可将表单内的相关元素分组
- <legend> 元素为 <fieldset>元素定义标题
- <label>元素为 input 元素定义标注

```
<form>  
  <fieldset>  
    <legend>健康信息</legend>  
    <label>身高: </label> <input type="text" />  
    <label>体重: </label> <input type="text" />  
  </fieldset>  
</form>
```

健康信息	
身高: <input type="text"/>	体重: <input type="text"/>

第四讲课后练习

- 学习
 - 课本 1-6, 15-18 章
- 辅助阅读
 - 中文版: HTML 初级指南、中级指南
- 电子书到群文件或网络课程中心上获取



隐藏



后退



前进



主页



字体



选项(O)

目录(C)

索引(N)

搜索(S)

书签(U)

HTML和CSS网页标准指南

HTML初级指南

快速启动

标签、属性和元素

文档标题

段落

标题

列表

连接

图象

表格

表单

综合在一起

CSS初级指南

应用CSS

选择符、属性和值

颜色

文本

边界和补白

边框

综合

HTML中级指南

span和div

元标签

表格

定义列表

Javascript

不良标签

CSS中级指南

类class和标识id选择符

组合和嵌套

伪类

快捷属性

背景图片

HTML高级指南

CSS高级指南

HTML和CSS网页标准指南

HTMLDog指南

介绍

用符合Web标准的HTML和CSS构建你的网站

欢迎来到**传承标准的HTML和CSS网页标准指南**，此标准指南来自著名的国外XHTML和CSS的实用指南网站HTMLDog，经Neo翻译而成。最初由Lv_V整理制作CHM电子书，后由CY重新整理制作。在本指南内，你不仅学到最常用的制做网页的技术，最重要的是，你还可以深入了解实用的、最新的**Web标准（Web Standards）**。

总共**六个**指南，基本上覆盖了HTML和CSS的范围。初级指南是按部就班的，你应该按顺序逐个学习。而中级和高级指南相对比较独立，包含各种技巧和技术，你可以按照你的需求选择性地学习。

目录

HTML初级指南

这是按部就班的最基础教程，如果你尚未入门，你应该从这里开始。

CSS初级指南

也是按部就班的，如果你学完了HTML出家指南，进一步学习吧。

HTML中级指南

对初学者和精通者都有用的一些HTML技术。

CSS中级指南

基于基本的CSS的一些“多余”和“古怪”的东西。

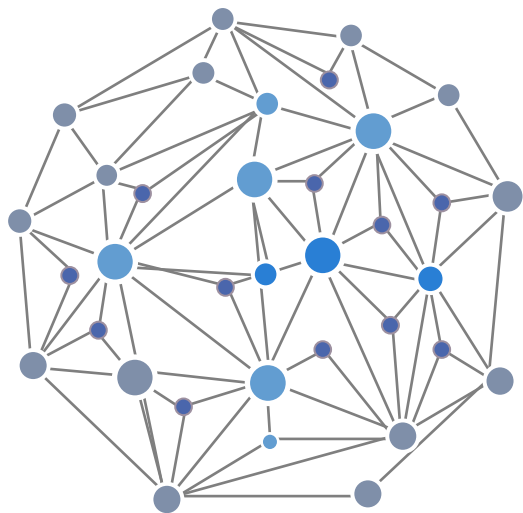
HTML高级指南

发挥HTML的兼容标准，增强亲和力和易用性。

Reference

- <http://www.w3school.com.cn>
- <http://www.w3schools.com/html>
- <http://www.w3schools.com/xhtml>
- <http://www.htmldog.com>

- <http://en.wikipedia.org/wiki/XHTML>
- <http://en.wikipedia.org/wiki/HTML5>



HTML 常用属性

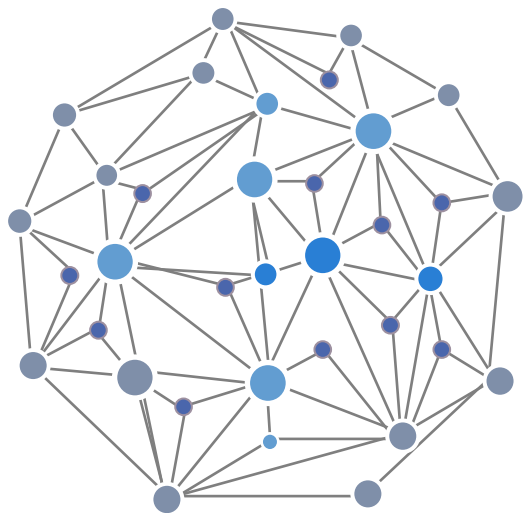
HTML 核心属性 Core Attributes

- 可以用于所有的标记
 - `class` 类属性
 - `id` 标识属性
 - `title` 指定一个标题
 - `style` 定义一个内联样式 (CSS)
 - `lang` 声明语言类型 (HTML5)
 - `xml:lang` 声明语言类型 (XHTML)
 - `xml:space` 命名空间 (XHTML)

详见: https://www.w3school.com.cn/tags/html_ref_standardattributes.asp

标记专用属性

- 还有许多属性是**专属于**某个标记名的
- 比如
 - a 中的 href
 - table 中 td 的 colspan, rowspan
 - form 中的 action, method



div 与 span 标记

无语义标记 **div** 和 **span**

- 语义：应是对标记的内容有一定附加意义的，比如 `h1` 表示标题，`p` 表示段落
- `div` 与 `span` 并不对内容做任何的解释，因此是没有语义的，只是用来对内容划分区域
- 使用时一般会添加类属性 `class` 或标识属性 `id`，并结合 CSS 进行排版

div 标记

- div: “division” 的简称
- div 可以包含段落、标题、表格甚至其它部分
- 通常用来对文档进行区块划分，比如页眉，页脚，正文，边栏

div sample

```
<div id="header">  
    .....页眉  
</div>  
<div id="content">  
    .....内容栏  
</div>  
<div id="sidebar">  
    .....边栏  
</div>  
<div id="footer">  
    .....页脚  
</div>
```

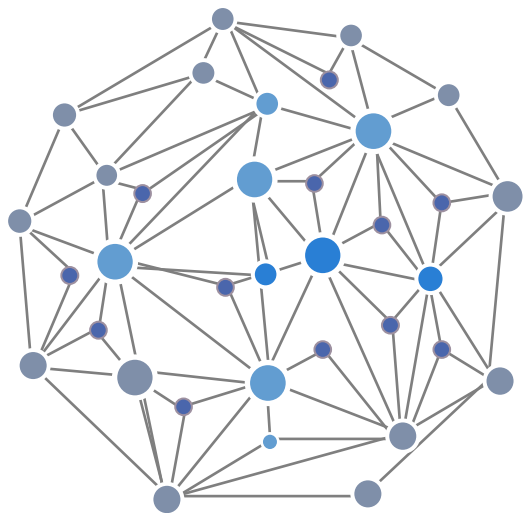
span 标记

- span 一般用在页面中标注某一小段文字，结合 CSS 能够使得页面一个段落中的某些元素显示效果和其他文字不一样

```
<p>这是一段文本这是一段文本这是一段文本  
<span class="purple">这是一段紫色的文本 </span>  
这是一段文本这是一段文本这是一段文本  
</p>
```

div 和 span 的差异

- div 和 span 的差异主要在于：
- div 是块元素
- span 是内联元素



块元素和内联元素

Block Element and Inline Element

块元素

- 块元素 (block element) 一般是其他元素的容器元素，可以包含块元素和内联元素
- 块元素一般都从新行开始，它可以容纳内联元素和其他块元素
- 常见的块元素如段落标记 `<p>`, `<h1>` ~ `<h6>` 等都是块元素

内联元素

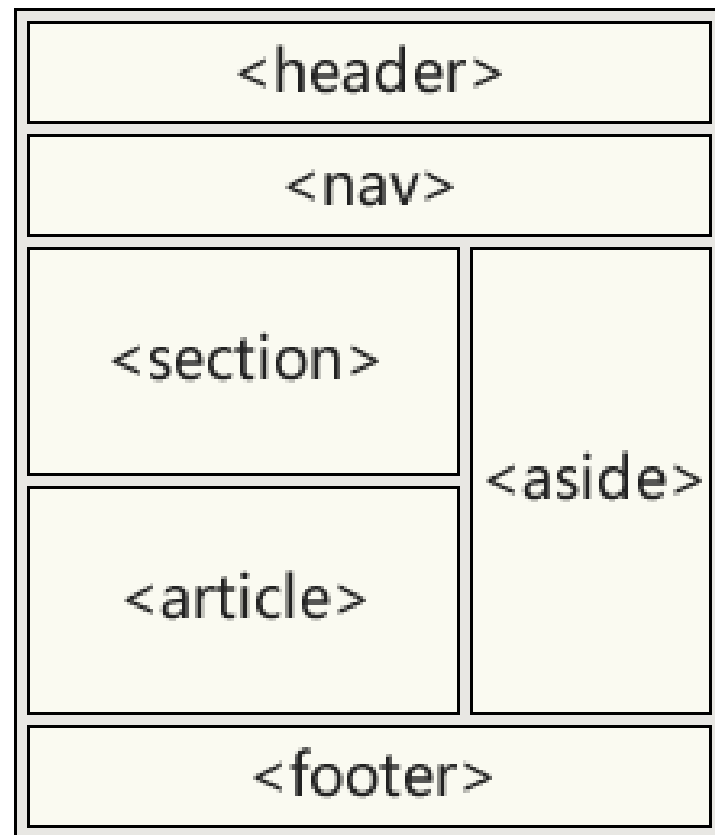
- 内联元素(inline element)一般都是基于语义级(semantic)的基本元素
- 内联元素只能容纳文本或者其他内联元素
- 常见内联元素如 `<a>`, ``, `` 等

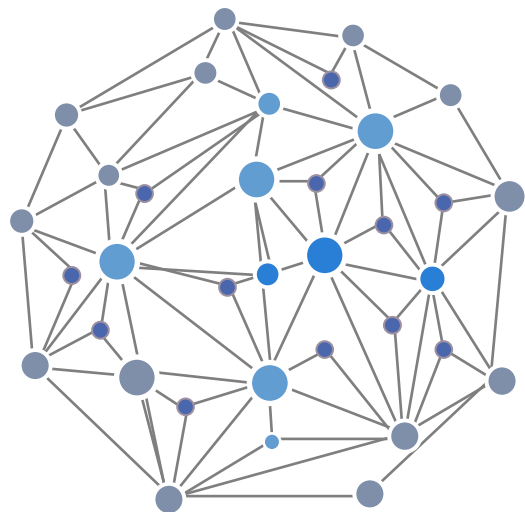
div 与 span 的差异

- div 作为块元素，div 区域内的对象与区域外的上下文之间会自动换行
- span 作为 内联元素，span 区域内的对象与区域外的上下文之间不会自动换行
- div 与 span 可同时使用，但 div 可包含 span，span 不可包含 div（由 DTD 决定）

HTML5 中新的语义元素

Tag	Description
<article>	定义文章。
<aside>	定义页面内容以外的内容。
<details>	定义用户能够查看或隐藏的额外细节。
<figcaption>	定义 <figure> 元素的标题。
<figure>	规定自包含内容，比如图示、图表、照片、代码清单等。
<footer>	定义文档或节的页脚。
<header>	规定文档或节的页眉。
<main>	规定文档的主内容。
<mark>	定义重要的或强调的文本。
<nav>	定义导航链接。
<section>	定义文档中的节。
<summary>	定义 <details> 元素的可见标题。
<time>	定义日期/时间。





将 HTML4 转换为 XHTML

将 HTML4 转换为 XHTML 文档

- 我们经常会碰到这样的问题：将老系统中的 HTML 文档转换成 XHTML 文档

XHTML 1.0 转换注意事项

1. 将 DOCTYPE 改成 Strict XHTML
2. 添加 xmlns 和 xml:lang 属性到 html 元素
3. 文档只有一个根元素<html>
4. 所有的标记名必须**小写和关闭**，空标记要在后面加上正斜杠自关闭
5. 所有的属性必须是**小写的**，属性值必须用**英文半角引号（"）**括起来
6. 将所有**的特殊字符转换成实体引用**

XHTML 1.0 转换注意事项

7. 精简属性是不允许的

- (比如 `<input type="checkbox" checked />`)

8. name 属性不再是合法的 (除了在表单元 素中) , 应该使用 id 来代替

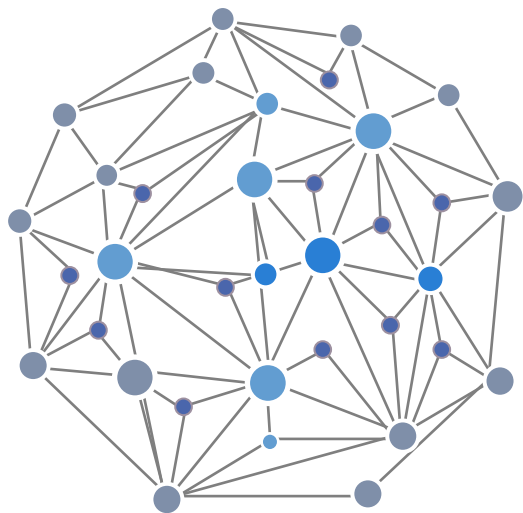
9. 目标属性 target 不是一个合法的属性

- 不过出于国情需要, 可能有的时候还有用
- `target="_blank"` 友情链接的需要

10. 在 img 里, alt 属性是必须的

推荐教程

- XHTML 结构化教程
 - http://www.w3school.com.cn/xhtml/xhtml_structural_01.asp
 - http://www.w3school.com.cn/xhtml/xhtml_structural_02.asp



文本格式化

改变文本的外观和意义

文本格式化

- XHTML/HTML 中有大量的标记用于改变文本的外观或是意义
- 分成两类：
 - 纯粹用于改变文本样式的标记
 - Physical Styles
 - 用于表示文本含义（语义）的标记
 - Content-Based Styles

Physical Styles

- 纯粹用于表现文本样式的标记:

...

加粗 **Bold**

<i>...</i>

意大利体 *Italic* (非斜体)

<tt>...</tt>

打字机字体 (等宽字体)

<big>...</big>

加大字体大小

<small>...</small>

减小字体大小

^{...}

上标 superscript

_{...}

下标 subscript

Physical Styles

- XHTML 是 XML，最好只包含有语义信息的内容
- 这些效果都能改用 CSS 实现
- 不过 XHTML 1.0 Strict 中还能使用

Obsolete Physical Style Tags

废弃的物理样式标记 (不要再使用)

- `<blink>` 交换前景背景色
 - `<center>` 居中
 - `<s>`, `<strike>` 定义加删除线的文本
 - `<u>` 定义下划线文本
 - `` 定义文本字体
-
- 这些早期的 HTML 标记已从 XHTML 1.0 Strict DTD 中删除, 不要再使用

用于表示文本含义的标记

- 强调，插入和删除
 - 缩写
 - 地址
 - 引用
 - 计算机输出
-
- 注意：有些浏览器对这些标记没有默认的风格(标记后的文本看过去跟普通文本类似)

强调，插入和删除

`...`

强调，通常显示为 bold

`...`

强调，通常显示为 italics

`<ins>...</ins>`

插入文本，通常显示为下划线

`...`

删除文本，通常显示为穿过文本的删除线

缩写

- `<abbr>` 定义缩写(abbreviation)

```
<abbr title="United States of America">USA</abbr>
```

- `<acronym>` 定义只取首字母的缩写

```
<acronym title="People Republic of China">PRC</acronym>
```

- 使用标题属性 **title** 来表示全称

地址

- `<address>` 定义地址信息
- 一般以斜体显示

`<address>`

Donald Duck

BOX 555

Disneyland

USA

`</address>`

Donald Duck

BOX 555

Disneyland

USA



完毕

引用

- `<blockquote>` 定义长的引用（单独的一段）
 - `<q>` 定义短的引用（行内引用）
 - `<cite>` 定义引用（未来可能取代q）
-
- `blockquote` 是块元素
 - `q` 和 `cite` 是内联元素，一般用在 `<p>` 内部

计算机输出类标记

- `<code>` 计算机代码文字（内联元素）
- `<pre>` 预格式化的文字（块元素）
- `<var>` 代码中的变量
- `<kbd>` 表明应由用户键入的文本

`<code> <var>isFormatted</var> = true;</code>`

`<p>Now type <kbd>woo hoo</kbd> </p>`

计算机输出类标记

```
<pre>
function ShowPRETag()
{
    string strMsg = "This is a pre formatted text."
    document.write(strMsg)
}
</pre>
```

其他

- `<dfn>`
 - 定义术语，用于高亮显示一个第一次使用的术语。用title属性表示解释。

Bob's **<dfn title="Dog">canine</dfn>** mother
and **<dfn title="Horse">equine</dfn>** ...

- `<samp>`
 - 用于举例


字符序列 `<samp>ae</samp>` 排版时通常会被转换成 æ 连字的形式。

其他

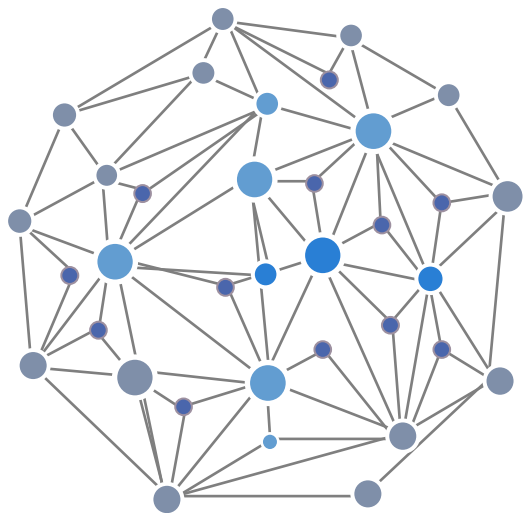
- `<bdo>`

重新定义文本的显示方向

```
<p> <bdo dir="rtl">god lmth</bdo> </p>
```



```
html dog
```



框架

Frames

框架概述

- 框架可以在浏览器中划分矩形区域，每个区域显示不同的文档
- W3C 不建议继续使用框架，因此框架通不过 XHTML 1.1 的验证，但还是有很多文档中仍会见到它
- 在文档中用<frameset>定义框架的结构，用<frame>指定每个区域显示的文档

XHTML 1.0 Frameset

- 由于XHTML 1.0 Strict和 1.1中已经删除了框架，因此使用框架必须使用特殊的DTD:

```
<!DOCTYPE html PUBLIC
```

```
"-//W3C//DTD XHTML 1.0 Frameset//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

<frameset>

- <frameset> 的作用是定义浏览器中框架的数目和布局
- 它替代了普通文档中的<body>元素，两者只能有其一
- <frameset> 必须有一个 rows 或者 cols 属性，可以为数值，百分数或者星号，默认值为 1

<frameset> Example

<frameset rows = "150, 200, 300">

The Lord of the Rings Trilogy

1. The Fellowship of the Rings
2. The Two Towers
3. The Return of the King

The Lord of the Rings Trilogy

1. The Fellowship of the Rings
2. The Two Towers
3. The Return of the King

College Sports

- Football
- Basketball

Heading 1 (h1)

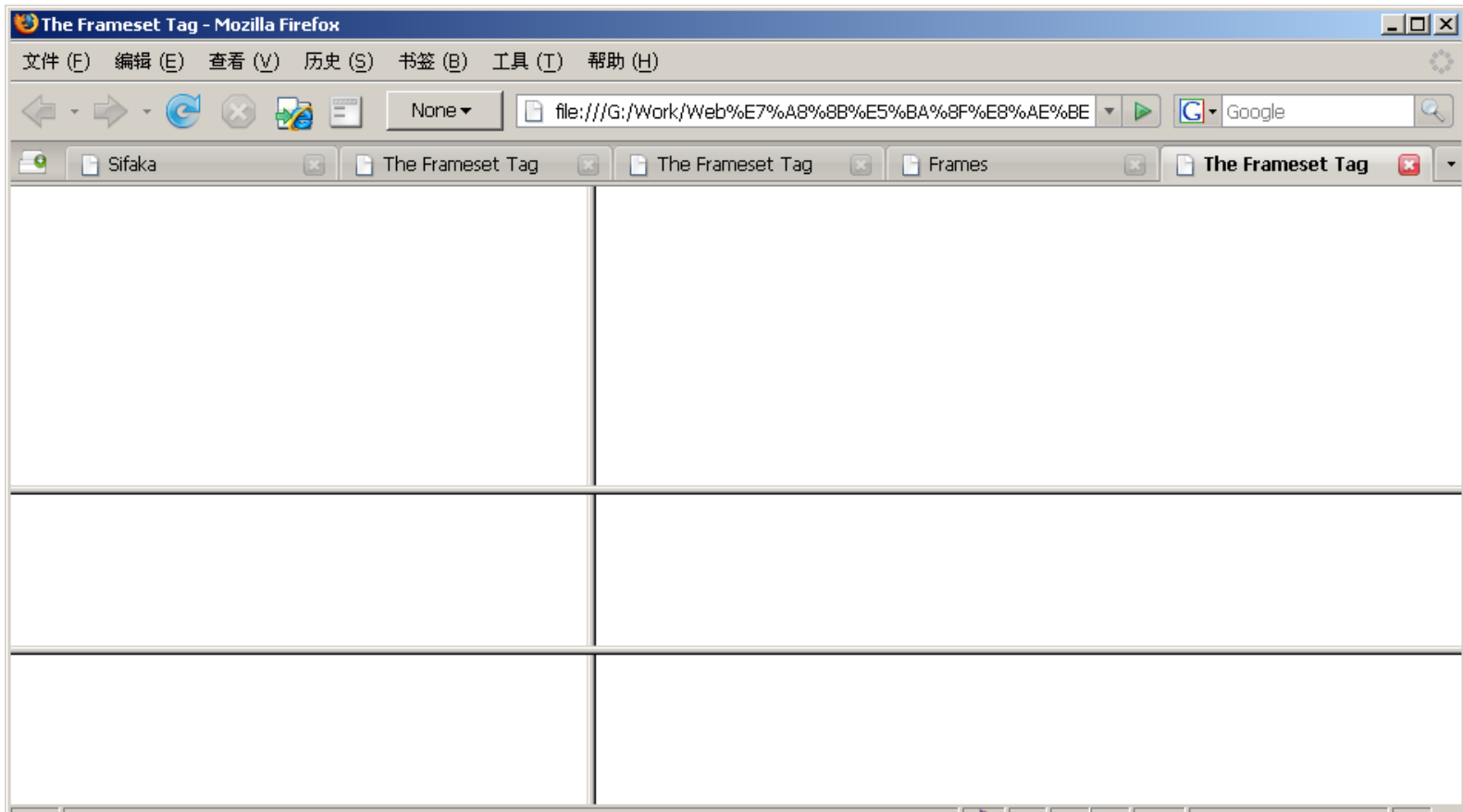
Heading 2 (h2)

Heading 3 (h3)

Heading 4 (h4)

<frameset> Example

**<frameset rows = "50%, 25%, 25%"
cols = "40%, *">**



<frame>

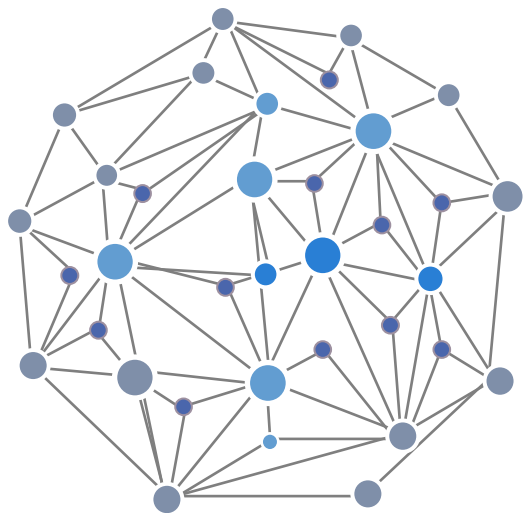
- **<frame>** 标记用以指定框架的内容，只能用在**<frameset>**中
- 框架集中的**<frame>**按照行的顺序显示

```
<frameset cols = "20%, *">
```

```
  <frame src = "contents.html" />
```

```
  <frame src = "fruits.html" name = "descriptions" />
```

```
</frameset>
```



阅读材料

Search Engine is Our Friend.

浏览器 Quirks Mode

- 浏览器的模式问题 Quirks Mode vs Standards Mode
 - <http://www.aoao.org.cn/blog/2007/01/browser-mode/>
- http://en.wikipedia.org/wiki/Quirks_mode
(代理)

关于 Web Standards 的高级网站

- **A List Apart (ALA)**
 - <http://www.alistapart.com/>
- **456 Berea Street**
 - <http://www.456bereastreet.com/>
 - <http://www.456bereastreet.com/archive/>
 - <http://www.456bereastreet.com/lab/>



标记查询手册 Tags Reference

- W3Schools HTML 参考手册
 - <http://w3school.com.cn/tags/index.asp>

THANKS

本章结束

陈昱

福州大学 计算机与大数据学院 软件工程系

