39-HTML语言：DTD到底是什么？

你好，我是winter。今天，我们来聊一聊HTML语言。

我们平时写HTML语言，都习惯把关注点放到各种标签上，很少去深究它的语法。我想你应该会有模糊的感觉，HTML这样的语言，跟JavaScript这样的语言会有一些本质的不同。

实际上，JavaScript语言我们把它称为“编程语言”，它最大的特点是图灵完备的，我们大致可以理解为“包含了表达一切逻辑的能力”。像HTML这样的语言，我们称为“标记语言（mark up language）”，它是纯文本的一种升级，“标记”一词的概念来自：编辑审稿时使用不同颜色笔所做的“标记”。

在上世纪80年代，“富文本”的概念在计算机领域的热门，犹如如今的“AI”和“区块链”，而Tim Berners-Lee当时去设计HTML，也并非是凭空造出来，他使用了当时已有的一种语言：SGML。

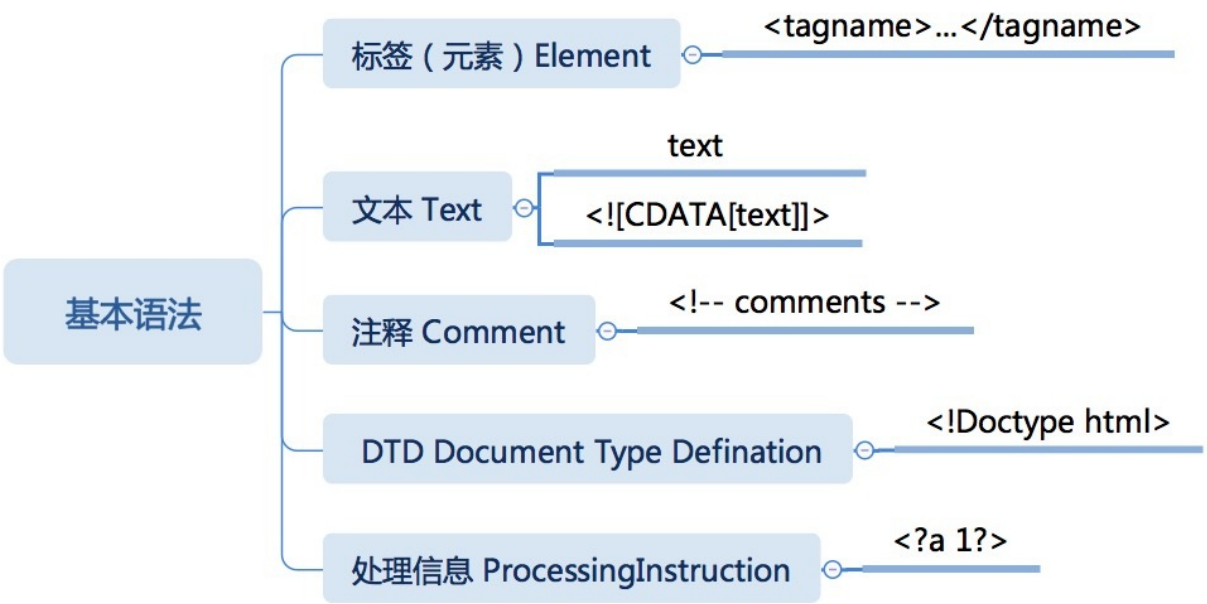
SGML是一种古老的标记语言，可以追溯到1969年IBM公司所使用的技术，SGML十分复杂，严格来说，HTML是SGML中规定的一种格式，但是实际的浏览器没有任何一个是通过SGML引擎来解析HTML的。

今天的HTML仍然有SGML的不少影子，那么接下来我们就从SGML的一些特性来学习一下HTML。这里我最想讲的是SGML留给HTML的重要的遗产：基本语法和DTD。

基本语法

首先，HTML作为SGML的子集，它遵循SGML的基本语法：包括标签、转义等。

SGML还规定了一些特殊的节点类型，在我们之前的DOM课程中已经讲过几种节点类型，它们都有与之对应的HTML语法，我们这里复习一下：



这里我们从语法的角度，再逐个具体了解一下。

标签语法

标签语法产生元素，我们从语法的角度讲，就用“标签”这个术语，我们从运行时的角度讲，就用“元素”这个术语。

HTML中，用于描述一个元素的标签分为开始标签、结束标签和自闭合标签。开始标签和自闭合标签中，又可以有属性。

- 开始标签：<tagname>
  - 带属性的开始标签：<tagname attributename="attributevalue">
- 结束标签：</tagname>
- 自闭合标签：<tagname />

HTML中开始标签的标签名称只能使用英文字母。

这里需要重点讲一讲属性语法，属性可以使用单引号、双引号或者完全不用引号，这三种情况下，需要转义的部分都不太一样。

属性中可以使用文本实体（后文会介绍）来做转义，属性中，一定需要转义的有：

- 无引号属性：<tab> <LF> <FF> <SPACE> &五种字符
- 单引号属性：' &两种字符
- 双引号属性：" &两种字符

一般来说，灵活运用属性的形式，是不太用到文本实体转义的。

文本语法

在HTML中，规定了两种文本语法，一种是普通的文本节点，另一种是CDATA文本节点。

文本节点看似是普通的文本，但是，其中有两种字符是必须做转义的：< 和 &。

如果我们从某处拷贝了一段文本，里面包含了大量的 < 和 &，那么我们就有麻烦了，这时候，就轮到我们的CDATA节点出场了。

CDATA也是一种文本，它存在的意义是语法上的意义：在CDATA节点内，不需要考虑多数的转义情况。

CDATA内，只有字符组合]]>需要处理，这里不能使用转义，只能拆成两个CDATA节点。

注释语法

HTML注释语法以<!--开头，以-->结尾，注释的内容非常自由，除了-->都没有问题。
如果注释的内容一定要出现 -->，我们可以拆成多个注释节点。

DTD语法（文档类型定义）

SGML的DTD语法十分复杂，但是对HTML来说，其实DTD的选项是有限的，浏览器在解析DTD时，把它当做几种字符串之一，关于DTD，我在本篇文章的后面会详细讲解。

ProcessingInstruction语法（处理信息）

ProcessingInstruction多数情况下，是给机器看的。HTML中规定了可以有ProcessingInstruction，但是并没有规定它的具体内容，所以可以把它视为一种保留的扩展机制。对浏览器而言，ProcessingInstruction 的作用类似于注释。

ProcessingInstruction 包含两个部分，紧挨着第一个问号后，空格前的部分被称为“目标”，这个目标一般表示处理 ProcessingInstruction 的程序名。

剩余部分是它的文本信息，没有任何格式上的约定，完全由文档编写者和处理程序的编写者约定。

DTD

现在我们来讲一下DTD，DTD的全称是Document Type Defination，也就是文档类型定义。SGML用DTD来定义每一种文档类型，HTML属于SGML，在HTML5出现之前，HTML都是使用符合SGML规定的DTD。

如果你是一个上个时代走过来的前端，一定还记得HTML4.01有三种DTD。分别是严格模式、过渡模式和frameset模式。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

严格模式的DTD规定了HTML4.01中需要的标签。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

过渡模式的DTD除了html4.01，还包含了一些被贬斥的标签，这些标签已经不再推荐使用了，但是过渡模式中仍保留了它们。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
```

frameset结构的网页如今已经很少见到了，它使用frameset标签把几个网页组合到一起。

众所周知，HTML中允许一些标签不闭合的用法，实际上这些都是符合SGML规定的，并且在DTD中规定好了的。但是，一些程序员喜欢严格遵守XML语法，保证标签闭合性，所以，HTML4.01又规定了XHTML语法，同样有三个版本：

版本一

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

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

版本二

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

版本三

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

其实你看看就知道，这些复杂的DTD写法并没有什么实际作用（浏览器根本不会用SGML引擎解析它们），因此，到了HTML5，干脆放弃了SGML子集这项坚持，规定了一个简单的，大家都能记住的DTD：

```
<!DOCTYPE html>
```

但是，HTML5仍然保留了HTML语法和XHTML语法。

文本实体

不知道你注意到没有，HTML4.01的DTD里包含了一个长得很像是URL的东西，其实它是真的可以访问的——但是W3C警告说，禁止任何浏览器在解析网页的时候访问这个URL，不然W3C的服务器会被压垮。我相信很多好奇的前端工程师都把它下载下来打开过。

这是符合SGML规范的DTD，我们前面讲过，SGML的规范十分复杂，所以这里我并不打算讲SGML（其实我也不会），但是这不妨碍我们了解一下DTD的内容。这个DTD规定了HTML包含了哪些标签、属性和文本实体。其中文本实体分布在三个文件中：HTMLsymbol.ent HTMLspecial.ent和HTMLlat1.ent。

所谓文本实体定义就是类似以下的代码：

```
&lt;
 
&gt;
&amp;
```

每一个文本实体由&开头，由;结束，这属于基本语法的规定，文本实体可以用#后跟一个十进制数字，表示

字符Unicode值。除此之外这两个符号之间的内容，则由DTD决定。

我这里数了一下，HTML4.01的DTD中，共规定了255个文本实体，找出这些实体和它们对应的Unicode编码，就作为本次课程的课后小问题吧。

总结

今天的课程中我们讲了HTML的语法，HTML语法源自SGML，我们首先介绍了基本语法，包含了五种节点：标签（元素）、文本、注释、文档类型定义（DTD）和处理信息（ProcessingInstruction）。

之后我们又重点介绍了两部分内容：DTD和文本实体。

DTD在HTML4.01和之前都非常的复杂，到了HTML5，抛弃了SGML兼容，变成简单的`<!DOCTYPE html>`。

文本实体是HTML转义的重要手段，我们讲解了基本用法，HTML4.01中规定的部分，就留给大家作为课后问题了。

今天的课后问题是：HTML4.01的DTD中，共规定了255个文本实体，请你找出这些实体和它们对应的Unicode编码吧。

精选留言：

- 天天 2019-04-25 09:46:32
  https://m.baidu.com/sf_edu_wenku/view/8fce2c4819e8b8f67c1cb9e9
  文本实体以及code 码

- 阿成 2019-04-25 09:18:52
  <![CDATA[<html></html>]]> 这样好像不行啊，不知道为啥。。。

- 孙清海 2019-04-25 09:05:56

知识真的是无穷无尽!现在反复锤炼winter老师的课程,打造出一个自己的知识脉络.