HTML语言：DTD到底是什么?

2019-04-25 winter

重学前端

进入课程 >

讲述：winter

时长 10:43　大小 9.82M

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。

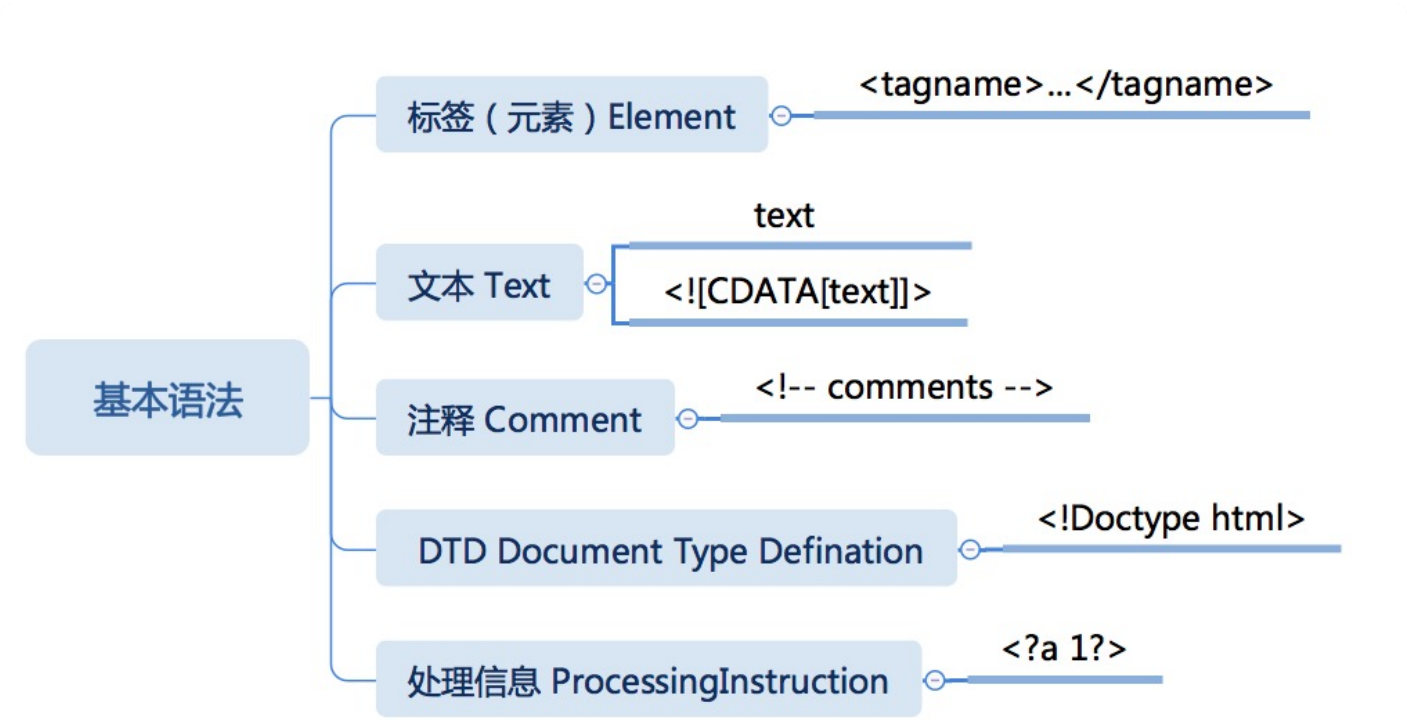
SGML 是一种古老的标记语言，可以追溯到 1969 年 IBM 公司所使用的技术，SGML 十分复杂，严格来说，HTML 是 SGML 中规定的一种格式，但是实际的浏览器没有任何一个是通过 SGML 引擎来解析 HTML 的。

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，一定需要转义的有下面几种。

- 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/str
```

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
```

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/
```

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。

上一篇 CSS动画与交互：为什么动画要用贝塞尔曲线这么奇怪的东西?

下一篇

精选留言 (9)

写留言

Dylan-Tseng

2019-04-28

很感谢老师的课程分享，我的整个知识体系中(不止前端)，只有广度，谈深度只有前端我了解多一些，在这40节课程中，大部分老师涉及的都有做了解，但继续深入还是空白，很感谢老师弥补了我很多的知识盲区，每篇文章我会花很多时间去推敲老师说的每句话的深意，有的可能花2-3天的时间。在我构建了一个完整的知识系统同时，我更能理直气壮的跟人高谈阔论某一个细节。
个人太喜欢老师的讲解方式了，点到为止，如果你去深究，发现里面知识点是另外一种天地，前人教导我的。我也常常给其他新人说的，思想原理很重要，无所谓语言。老师很多的时候是传递我们一种思想。而我就喜欢将思想运用到项目。真的收获颇多。
如果后面老师有设计模式，算法。数据结构的课程，一定要让我们知道啊，工作了才知道这些东西太重要了。

1 9

mfist

2019-04-25

遍历文件，通过字符串匹配得到实体和code码，好像只有253个。篇幅原因只能贴一半了。
{"nbsp":" ","iexcl":"&#161¡","cent":"&#162¢","pound":"&#163£","curren":"&#164¤","yen":"&#165¥","brvbar":"&#166¦","sect":"&#167§","uml":"&#168¨","copy":"&#169©","ordf":"&#170ª","laquo":"&#171«","not":"&#172¬","shy":"&#173","reg":"&#174®","macr":"&#175¯","deg":"&#176°","plusmn":"&#177±","sup2":"&#178²","sup3":"&#179³","acute":"&#180´","micro":"&#181µ","para":"&#182¶","middot":"&#183·","cedil":"&#184¸","sup1":"&#185¹","ordm":"&#186º","raquo":"&#187»","frac14":"&#188¼","frac12":"&#189½","frac34":"&#190¾","iquest":"&#191¿","Agrave":"&#192À","Aacute":"&#193Á","Acirc":"&#194Â","Atilde":"&#195Ã","Auml":"&#196Ä","Aring":"&#197Å","AElig":"&#198Æ","Ccedil":"&#199Ç","Egrave":"&#200È","Eacute":"&#201É","Ecirc":"&#202Ê","Euml":"&#203Ë","Igrave":"&#204Ì","Iacute":"&#205Í","Icirc":"&#206Î","Iuml":"&#207Ï","ETH":"&#208Ð","Ntilde":"&#209Ñ","Ograve":"&#210Ò","Oacute":"&#211Ó","Ocirc":"&#212Ô","Otilde":"&#213Õ","Ouml":"&#214Ö","times":"&#215×","Oslash":"&#216Ø","Ugrave":"&#217Ù","Uacute":"&#218Ú","Ucirc":"&#219Û","Uuml":"&#220Ü","Yacute":"&#221Ý","THORN":"&#222Þ","szlig":"&#223ß","agrave":"&#224à","aacute":"&#225á","acirc":"&#226â","atilde":"&#227ã","auml":"&#228ä","aring":"&#229å","aelig":"&#230æ","ccedil":"&#231ç","egrave":"&#232è","eacute":"&#233é","ecirc":"&#234ê","euml":"&#235ë","igrave":"&#236ì","iacute":"&#237í","icirc":"&#238î","iuml":"&#239ï","eth":"&#240ð","ntilde":"&#241ñ","ograve":"&#242ò","oacute":"&#243ó","ocirc":"&#244ô","otilde":"&#245õ","ouml":"&#246ö","divide":"&#247÷","oslash":"&#248ø","ugrave":"&#249ù","uacute":"&#250ú","ucirc":"&#251û"}

5

阿成

2019-04-25

<![CDATA[<html></html>]]> 这样好像不行啊，不知道为啥。。。

1 2

郭郭

2020-03-25

非常完美地弥补了之前学习html的知识盲区❤

一步

2019-05-23

为什么 w3school 这个文本实体文档 http://www.w3school.com.cn/tags/html_ref_symbols.html
找不到   < 这些符号呢?

是零壹呀

2019-05-06

为什么我爬下来，只有253个？有一样的么

wuxiii

2019-05-05

如果低版本的浏览器是用h5的DTD，浏览器是否可以正常解析?

1

天天

2019-04-25

https://m.baidu.com/sf_edu_wenku/view/8fce2c4819e8b8f67c1cb9e9
文本实体以及code 码

孙清海

2019-04-25

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.