

HTML自学笔记#1

原创 Zerg_Wang 于 2019-06-01 17:00:56 发布 165 收藏
分类专栏： Web Development

编辑 版权



Web Development 专栏收录该内容

0 订阅 3 篇文章

环境：VS Code

插件：HTML Snippets、open in browser

一些概念

HTML：Hyper Text Markup Language，超文本标记语言，是一种描述网页内容的语言，不被视作编程语言。

元素：网页由元素构成，元素内容即开始标签与结束标签之间的代码。有些元素没有内容，且无开始、结束标签，自成一体，如
以及<hr/>，本身就是一个完整的元素，这些叫空元素。

属性：为元素提供附加信息，一般在开始标签中定义，如：（居中显示标题）

```
<h1 align="center">this is my title</h1>
```

样式

即style，是用于改变元素样式的属性。

style的使用越来越广泛，可普遍用于多数元素中，取代了一些旧的属性，例如：

```
1 <body bgcolor="red">
2 </body>
```

被替换为

```
1 <body style="background-color:yellow">
2 </body>
```

也可用style定义文本对齐方式：

```
<h1 style="text-align:center">This is a heading</h1>
```

文字效果

文字显示

```
1 <b>      粗体
2 <strong>加重语气
3 <em>     强调
4 <i>      斜体
5 <big>    大号字
6 <small>  小号字
7 <sub>    下标
8 <sup>    上标
9 <ins>    插入字
10 <del>   删除字
```

11 | `<bdo dir="rtl">` 反向字（就是翻转输出为“字向反”）

对应效果：

正常字

粗体 **加重语气** *强调* *斜体* 大号字 小号字 _{下标} ^{上标} 插入字 ~~删除字~~

不难发现，粗体和加重语气字体样式是一样的，实际上，``是逻辑标签，``是实体标签，对于搜索引擎在检索网页时会更看重逻辑标签，相同的还有``和`<i>`，其中``也是逻辑标签。

代码显示

`<kbd>`、`<samp>`、`<code>`分别模拟键盘输入字符、计算机输出字符、源代码的样式，若不加属性修饰，则表现一致。

正常字符

代码。
键盘码。
计算输出码。

如果要使用`<code>`在网页上显示代码，应配合`<pre>`使用，因为`<pre>`会保留多余的空格和空行。

```
1 | <p>C++ Code:</p>
2 | <code><pre>
3 |     #include <stdio.h>
4 |     int main()
5 |     {
6 |         printf("Hello World");
7 |         return 0;
8 |     }
9 | </pre></code>
```

C++ Code:

```
#include
int main()
{
    printf("Hello World");
    return 0;
} https://blog.csdn.net/Zerg_Wang
```

然而，`stdio.h`却不见了，这是因为在HTML中`<`和`>`为特殊符号，要显示这两个符号，则需要用写成：`#include <stdio.h>`

这与某些编程语言中的转义字符类似，但不是通过在要输入字符加特殊符号，而是直接使用特定字符串代替，例如“`<`”用“`<`”替代。（注意分号也是其中一部分，引号不是）

引用

`<cite>`：书名引用，效果：

`<p><cite>The Great Gatsby</cite> by Fitzgerald</p>`

The Great Gatsby by Fitzgerald

`<q>`：短引用，自动添加双引号：

`<p>洪世贤：<q>你好骚啊！</q></p>`

洪世贤: "你好骚啊! "

<blockquote> : 长引用，会对缩进进行处理：

```
1  <p>古语有云：</p>
2  <blockquote>大道之行也，天下为公，选贤与能，讲信修睦。
3      故人不独亲其亲，不独子其子，使老有所终，壮有所用，
4      幼有所长，矜、寡、孤、独、废疾者皆有所养，男有分，女有归。
5      货恶其弃于地也，不必藏于己；力恶其不出于身也，不必为己。
6      是故谋闭而不兴，盗窃乱贼而不作，故外户而不闭，是谓大同。
7  </blockquote>
```

古语有云：

大道之行也，天下为公，选贤与能，讲信修睦。故人不独亲其亲，不独子其子，使老有所终，壮有所用，幼有所长，矜、寡、孤、独、废疾者皆有所养，男有分，女有归。货恶其弃于地也，不必藏于己；力恶其不出于身也，不必为己。是故谋闭而不兴，盗窃乱贼而不作，故外户而不闭，是谓大同。

https://blog.csdn.net/Zerg_Wang

<abbr> : 缩写，鼠标移到缩略词上会显示全称。

```
<abbr title="Hyper Text Markup Language">HTML</abbr>
```



<address> : 显示作者、网站编辑者等人的信息：

```
1  <address>
2      Zerg Wang<br/>
3      zergwang@example.com<br/>
4      Guangzhou, China
5  </address>
```

Zerg Wang
zergwang@example.com
Guangzhou, China

链接

target使用

```
<a href="https://www.bilibili.com" target="_blank">访问同性交友网站</a>
```

没有target属性，单击链接则在当前页跳转，若target为_blank，单击链接则在新标签页打开。

命名锚的用法

假如说某个页面内容很多，要找到自己所需内容较为麻烦，这时可通过命名锚快速定位，例如说我要快速跳转到第三章，则有以下两个步骤：

- 1. 首先在第三章标题处，我们设置一个定位符（也可以叫锚），定位符名为C3，之后我们通过这个定位符来跳转。

```
<h1><a name="C3">第三章</a></h1>
```

2. 接下来，我们写个跳转到第三章的链接即可：

```
<a href="#C3">跳转到第三章</a>
```

注意要在定位符前加上#号。

链接可在其所指之处的前面，也可以在后面，无所谓~

除了在当前页面跳转，也可以跨网页跳转，例如我在网页1，要跳转到网页2的A处，则要在A设置定位符（假设C8），然后在网页1中：

```
<a href="网页2的网址#C8">跳转</a>
```

其他

路径

同其他编程语言的路径写法基本一致，文件夹间用“/”而不是Windows下的“\”。

标题

h1、h2、h3、h4、h5、h6分别对应的是一级标题、二级标题……而不是第一个标题、第二个标题。

此外，不应为了凸显某些正文内容而故意设置其为标题，这是因为搜索引擎会通过检索标题来编制网页索引。

图片

```

```

后面的width和height不是必须。若未规定width和height则按原图大小显示，若仅规定一个，按规定值成比例显示，若都规定，按规定值拉伸、压缩显示，不会截取部分显示图片。

可为图像添加alt属性，用于图片无法正常显示时的替代文本信息：

```

```

分割线

<hr/>，效果：

```
1 | <h1>this is my title</h1>
2 | <hr/>
3 | <p>somtheing I wanna say</p>
```

this is my title

somtheing I wanna say

换行符

换行符
，使用以及与另起一段的区别：

```
1 | <p>12<br/>3</p>
2 | <p>aab</p>
```

12
3

aab

可以看出行距是不一样的，这是因为HTML默认会在段落、标题元素前后添加一个额外的空行。

注释

```
1 | <!--这是注释-->
2 |
3 | <!--这
4 |     还
5 |     是
6 |     注
7 |     释
8 | -->
```

可单行也可多行。

HTML语法特点

在编写HTML发现，虽然其有严格、完善的语法，但浏览器对于HTML代码并不严格，即使是肉眼可见的严重错误，运行后可能也会一切正常.....此外，HTML在编写上也有一些“不成文”规定，以下举几个例子：

属性简写

```
1 | 错误: <input checked />
2 |
3 | 正确: <input checked="checked" />
```

关闭标签

所有元素均有开始标签和关闭标签没有关闭标签，例如<html>和</html>，但有时不写关闭标签网页仍能正常工作。

对于空元素，也必须关闭。例如换行，写
是不对的，应为
，当然两者都能正常运行。

元素嵌套

```
<p><a href="index.html">这是一个链接</p></a>
```

这是一个错误的写法，但仍能正常运行.....正确写法：

```
<p><a href="index.html">这是一个链接</a></p>
```

大小写

HTML对大小写不敏感，<h1>和<H1>均可，但一般都是纯小写。

引号

HTML对单双引号也不敏感，属性中的值可以用单引号，也可以用双引号，若值本身带有双引号，则必须用单引号引入。

然而，一般属性值，什么引号都不带，也能正常运行.....

空白忽略

源代码中的连续多个空格只算一个，连续多个空行也只算一个空格，例如：

```
1 | <h1>this          is my title</h1>
2 | <h1>this
3 |
4 |         is my
5 |
6 |         title
7 | </h1>
```

和<h1>this is my title</h1> 是一样的。

版本声明

HTML一路发展，有多个版本，不同版本之间，语法、元素不完全相同，因此需要版本声明语句<!DOCTYPE> 方便浏览器解析，例如HTML5版本的写法：

```
<!DOCTYPE html>
```

XML与XHTML

XML (Extensible Markup Language，可拓展标记语言) 与HTML语法基本一致，但语法更为严格，上面提到的HTML的种种不规范写法，在XML中就行不通了（无法正常运行）。

功能上XML与HTML略有不同。两者都可定义网页内容（存储数据），然而，HTML更偏向于设计网页的布局、显示效果等等，而XML的使用目的是数据的交流与传输，或者说，HTML专注于数据的展示效果，XML则专注于数据的传输共享。网页在设计时，使用HTML定义网页的布局，数据部分不会像上面一样一股脑写在<body>中，而是通过连接到外部的XML中。如果涉及到数据的更改或、环境改变、网站变更等，直接动XML即可，无需对HTML中用于布局的代码大幅修改。

至于XHTML（可扩展超文本标签语言，Extensible Hyper Text Markup Language），按我的理解，就是遵从XML严格语法规则的HTML语言……

参考资料：

<http://www.w3school.com.cn/>