w《尚学堂》马士兵HTML+CSS+JavaScript学习Notes

\*仅仅总结一些W3CSchool上面没有的,所有的代码都在Workspace中

No.1 HTML简介

1. 基本知识
   1. HTML是Hyper Text Markup Language（超文本标记语言）的缩写
   2. 超文本：文本+图片+音频+链接+…
   3. 标记语言：浏览器根据标记显示内容
   4. 来自于SGML（Standard General Markup Language,标准通用标记语言）
   5. 是写给浏览器的语言，专注于在Web上传递信息
2. 历史+外延
   1. 历史
      1. HTML1.0 –> HTML2.0（IETF制定）->HTML4.0（W3C制定）
         1. IETF：Internet Engineering Task Force（也制定HTTP等Protocol）
         2. W3C：World Wide Web Consortium（也制定XML等语言）
   2. 外延
      1. XHTML（X short for XML）
         1. 含义：符合XML（可以说是更严格语法标准的HTML）标准的HTML
      2. DHTML（D short for Dynamic）
         1. 含义：(X)HTML / CSS / JavaScript的综合运用

No.2 HTML语法标记规则

1. 标记的形式
   1. “头尾式”：<元素名称 属性定义>被该元素修饰的内容</元素名称>
      1. Eg. <font>被font标签修饰的文字</font>
   2. “尾式”<元素名称 属性定义/>
      1. Eg. <img src = ”…” />
   3. “头式”<元素名称 属性定义>
      1. Eg. <br>
2. 其他注意事项
   1. 属性的值可以位于尖括号内部，属性值必须加上””或者’’，不写的话也可以浏览器也可以解析，但是不推荐（不符合XHTML）；
   2. 又开始必须有结束（XHTML）
   3. 标签不能错位嵌套使用
      1. Eg. <a><b></a></b>，这样是不行的；
      2. Eg.<a><b></b></a>，这样是可行的
   4. 标签不区分大小写
      1. 但是应该养成良好的编程习惯，推荐使用小写

NO.3 HTML程序实例（基础版本）

<html>

<head>

<title>典型的HTML程序实例</title>

</head>

<body>

我会努力的！

<!—

这里是注释，你看不到我哦！

🡪

</body>

</html>

No.4 HTML中head的常用标签

1. meta标签
   1. 用途：用于设置一些头信息（其实是在头里面设置一些该.html页面的信息）。
   2. 用法：“头式”<meta http-equiv=’content-type’ content=’text/html; charset=gb2312’>
2. title标签
   1. 用途：用于设置菜单标题栏中的标题
   2. 用法：“头尾式”
3. style标签
   1. 用途：定义CSS格式
   2. 用法：“头尾式”
4. Script标签
   1. 用途：定义脚本语言
   2. 用法：“头尾式”

No.5 Body 中的标签

1. body的属性，Eg.background and bgcolor and text and link and alink and vlink and so on.But 这些属性现在都放在CSS文件当中了。
2. 链接标签：<a href = “index.html”>链接文字内容</a>
3. 锚点标签：<a name = “Anchor’s name”>Label Content</>
   1. 程序实例

<html>

<head><title>标题</title></head>

<body>

<a href=”index.html#anchorPoint”9>contact.html</a>

</body>

</html>

No.6 路径问题

1. 本地路径
   1. “C:\dir1\dir2\...”
2. 绝对路径
   1. “http://www.baidu.com/img\_url.jpg”
3. 相对路径(相对于URL地址来说)
   1. “images/01.jpg”
   2. “../../images/01.jpg

No.7 URL URI URN说明

1. URL
   1. Uniform Resource Locator(同一资源定位符)
   2. 网络协议+主机名称+端口号+资源名称（定位标记）
   3. Htttp://www.baidu.com:80/index.html#top
   4. 带有参数的URL及编码问题
2. URN
   1. Uniform Resource Name
   2. 持久可用的资源标准名称
   3. 例如：邮箱名称Count.MCristo@qq.com
3. URI
   1. Uniform Resource Identifier
   2. 包含URL和URN

No.8 html中引用css语句

<link rel=”stylesheet” href=”css文件路径” style=”text/css”>

No.9 Z轴的定位：style的属性z-index（其值采用的是数值形式，1是最底层）

---------JavaScript----------

No.10 JavaScript语言简介

1. JavaScript是一种描述性语言，它与HTML结合起来，用于增强网页的交互性；
2. 由NetScape利用SUN的Java语言开发，但是与Java是完全不同的语言，是一种函数式语言；
3. 客户端的JavaScript语言必须要有浏览器的支持；
4. JavaScript语言的语法
   1. 基础语法
   2. DOM（Document Object Model）
   3. BOM（Browser Object Model）

No.11 写在HTML中Body中的JavaScript只要出现语法上的错误就不会再执行下一句了。

No.12 JavaScript的变量是弱类型变量，只有一个变量类型var。

No.13 JavaScript的函数可以定义在当前文档的任何部位。

No.14 JavaScript的对话框

1. 警告框(alert)：出现一个提示信息
2. 询问框(prompt)：返回输入的值
3. 确认框(confirm)：根据不同的选择，返回true(确定)/false(取消)

No.15 JavaScript中的内置对象

1. This
   1. 指的是当前的对象（当前的标签）
   2. 实例：Demo\_JavaScript\_Built-In Object.html
2. for…in
   1. in后面跟一个对象，对此对象中的所有元素循环一次（注意JavaScript中的数组名称是下标的值）
3. with
   1. 为一段代码建立一个缺省的对象，任何无对象的属性引用，都将使用该缺省的对象
4. New
   1. 用于生成一个新的对象

No.16 窗口中的对象和元素（JavaScript中的BOM）

* + - 1. Window
         1. 当前窗口
         2. 新开窗口
         3. 通过本地窗口控制新开窗口
      2. Location
         1. 获取或设置现有文档的URL
      3. History
         1. 先前访问过的URL的历史列表
         2. 常用方法：back() 、go(number)
      4. Document
         1. 当前文档的对象

Document.write()：向客户端浏览器输出内容

Document.formName：可以使用这个方法得到表单名称

Document.referrer

No.17 JavaScript的学习参考文档：

1. DHTML手册.chm
2. JavaScript Programmer’s Reference.pdf - Wrox Press
3. JavaScript Examples Bible[2007] –John Wiley & Sons
4. Professional JavaScript For Web Developers.pdf[2005] – Wrox

No.18 JavaScript调试

1. IE浏览器：工具-Internet选项-高级-打开禁用脚本调试 and 显示每个错误的通知

No.19 JavaScript中（相互关联的）列表（很重要）

No.20 正则表达式要学习一下

No.21 JavaScript中的Document详解

No.22 JavaScript中经常使用document.getElementById();来获取HTML页面中id标示的标签

No.23 JavaScript中的innerHTML是用来动态指定标签里面的内容的如，document.getElementById(“username”).innerHTML = “<font color=’blue’>”;

No.24 无论是单进程还是多进程一定要理解程序执行的顺序（一般来说单进程的程序较多），这样对于程序的理解和编写有很大的帮助。

No.25 JavaScript中的Tree也要理解，就是一级目录、二级目录等等