***Web开发学习***

目录

[简介 3](#_Toc465714250)

[1. Web基础 3](#_Toc465714251)

[2. 学习资料 3](#_Toc465714252)

[Web Html 4](#_Toc465714253)

[1. Html全称 4](#_Toc465714254)

[2. html标签的分类 4](#_Toc465714255)

[3. html 基本结构 4](#_Toc465714256)

[4. html元素通用属性 5](#_Toc465714257)

[5. 标题标签 5](#_Toc465714258)

[6. 文本内容的修饰 5](#_Toc465714259)

[7. 行级标签 5](#_Toc465714260)

[8. 块级标签 5](#_Toc465714261)

[9. 超链接<a> 6](#_Toc465714262)

[10. 表单元素 7](#_Toc465714263)

[WEB CSS 9](#_Toc465714264)

[1. CSS基本语法 9](#_Toc465714265)

[2. 基本属性 9](#_Toc465714266)

[3. CSS引用方式 9](#_Toc465714267)

[4. CCS选择器 9](#_Toc465714268)

[5. CSS特性 9](#_Toc465714269)

[6. 元素分类 10](#_Toc465714270)

[7. 盒子模型 10](#_Toc465714271)

[8. 布局模型 10](#_Toc465714272)

[9. 层模型 10](#_Toc465714273)

[Web JavaScript 11](#_Toc465714274)

[1. JS的执行流程 11](#_Toc465714275)

[2. JS的引入方式 11](#_Toc465714276)

[3. JS变量 11](#_Toc465714277)

[4. JS函数 11](#_Toc465714278)

[5. JS运算 11](#_Toc465714279)

[6. JS数据输出 11](#_Toc465714280)

[7. 数据类型 12](#_Toc465714281)

[8. JS内置对象 12](#_Toc465714282)

[9. JS创建对象 12](#_Toc465714283)

[HTML DOM 13](#_Toc465714284)

[1. Document 13](#_Toc465714285)

[2. 节点 13](#_Toc465714286)

[Jquery 15](#_Toc465714287)

[1. Jquery语法 15](#_Toc465714288)

[2. Jquery选择器 15](#_Toc465714289)

[JSON和XML 16](#_Toc465714290)

[1. 简介 16](#_Toc465714291)

[2. XML 16](#_Toc465714292)

# 简介

1. Web基础

Html+CSS+JavaScript Web前台

Html--网页基本内容和结构

CSS——页面美化

Javascript——网页的动态效果和数据的验证

静态和动态页面

Java web 后台（）

Oracle数据库DB

CSS——网页的皮肤

Html5+CSS3（现在最新CSS5）

Web 基础 (Browser)

Html

CSS

JvaScr

OOP using java(Server)

J2EE

Database(JDBC)

DDl

DML

DCI

1. 学习资料

[www.w3school.com.cn](http://www.w3school.com.cn)

WWW.RUNOOD.COM

软件：notepad++，atom

Html注释 <!--注释内容-->

CSS注释 /\*注释内容\*/

JS注释 单行//注释内容，多行/\*注释内容\*/

（注：文本中空格、换行不转义）

(注：正斜杠“/”反斜杠”\”； 多个设置，采用就近原则）

# Web Html

鄢客Alex Email:alex.yan@infosys.com

1. Html全称

—— Hyper Text Markup Language(html)

是超文本标记（标签）语言，是一种纯文本的语言；非编程语言

1. html标签的分类

单标签

A）带有属性的单标签

Syntax:

<标签名称 属性名称1=”属性值1” 属性名称2=”属性值2”/>

eg:

<hr width=”80%” align=”center”/> //hr水平线

B）没有属性的单标签

<br/> //换行

双标签

1. 带有属性

<font color=”red” size=”5”>被修饰的内容</font>

1. 没有属性

<html></html>

<head></head>

1. html 基本结构

<!doctype>

<html>

<!—html 头部内容，页面文档标题，编码格式，外部样式,JavaScript (这句注释了)-->

<head>

<title></title>

<meta charset=”utf-8”>

<link 外部样式(.css)>

<script>jscode和外部引入</script>

</head>

</html>

<title></title>---设置标签信息（查询、搜索）

<meta>---标记网页的基本信息，设置网页关键字，作者信息

<meta name=”keyword” content=”关键字1，关键字2”>——多个关键字（逗号分隔）

<meta name=”author” content=”author\_name”>

<meta http-equiv=”Content-Type” content=”text/html”; charset=”utf-8” >//网页的编码格式（gbk(简体中文)、iso-885901(英文)、big-5(繁体中文)、utf-8(国际通用)）【编码压缩率和通用性，utf-8通用性好压缩率低】

1. html元素通用属性

|  |  |
| --- | --- |
|  | 属性 |
| Class | Html元素定义的一个或多个类名（方便引入样式） |
| ID | 唯一的ID号（ID选择器） |
| Style | 应用在标签中的行内样式（inline style） |
| Title | 元素的额外信息（不常用） |

1. 标题标签

<h1>……..<h6> 其中h1标签字体最大，h6最小（字体都是加粗效果）；

Eg:

<h1>标题1</h1>

<h2>标题2</h2>

<h3>标题3</h3>

1. 文本内容的修饰

<b>粗体</b>

<i>斜体</i>

<u>下划线</u>

<s>删除线</s>

<sub>下标</sub>

<sup>上标</sup>

<strong>加强语气的效果</strong>

1. 行级标签

标题标签

段落标签

<p>自成段落内容</p>

1. 块级标签

有序、无序列表，定义列表，表格，表单，分区标签

a)有序标签

<ol type=”a” or “A” or “I” or “i” or “1”>

<li>User</li>

<li>UserName</li>

<li>Userlogn</li>

</ol>

并列多个选项，不同选项之间有先后顺序

b)无序标签

<ul type=”disc” or “circle”>

或<ul style=”list-style-type:none”>

<li>User</li>

<li>UserName</li>

<li>Userlogn</li>

</ul>

c)定义列表（html5弱化，定义某个名称或术语的解释）

<dl>

<dt>术语</dt>

<dd>术语的具体解释</dd>

</dl>

d)表格<table>（规整显示）

<table>

<tr>Row</tr>

<td>Column</td>

</table>

e)表单<table>（文本框、密码框、单多选按钮、下拉列表）

<input type=”text” or “password “ ”button” “checkbox”>

需要在<body>标签内嵌入一个<form>标签

<form action=”地址” method=”post or get”>

<input type=”” 属性1=”属性值”…/>

</form>

Eg：

<form action=”” method=”post”>

<p>User

<input type=”text” size=”20” maxlength=”20”/>

</p>

</form>

e)分区标签<div>（页面布局时区域块的划分）

在div整个大的‘’容器’’可以放置有无序列表、定义列表

常见的块级布局结构

1）Div-ul(ol)-li ——>应用于导航或菜单选项；

2）div-dl-dt-dd——>应用于图文混搭的场合；

3）table-tr-td——>规整显示内容，应用于导航

4）form-table-tr-td——>应用于表单布局的整合；

通常将marquee标签布局在<div>标签中。

<div>

</div>

1. 超链接<a>

<a href=”http://www.baidu.com” target=”\_blank”>

百度知道

</a>

Href——》本地、网络页面（链接地址）

Target——》取值 \_self（当前页面）、\_blank（新建页面中显示超链接内容）

超链接使用的场合

1. 网页之间的链接
2. 锚链接

完成步骤

Step1： 在页面的B位置标记<a id=”market”>目标位置</a>

Step2： 在A位置设置Href属性值”#标记名”

Eg：

<p>

<a href=”#C4”>查看第四章</a>

</p>

<h3>段落1</h3>

<p>段落1：福克斯的电脑</p>

<h3>段落2</h3>

<p>段落2：福克斯的电脑</p>

<h3>段落3</h3>

<p>段落3：福克斯的电脑</p>

<h3><a id=”C4”>段落4</a></h3>

<p>段落4：福克斯的电脑</p>

1. 功能性链接—发邮件的方式

<a href=”mailto:infosys\_User@infosys.com”>发邮件</a>

1. 表单元素

（涉及到文本框、下拉列表、单多选按钮、提交按钮、重置按钮等）

1. Form标签的基本语法

<form action=”页面” method=”提交方法”>

<!—布局文本框等-->

</form>

Action取值：规定表单提交信息时，向何处发送表单数据信息，取值为缺省时，默认提交到当前页面；

Method取值为post和get（get安全性不高，post安全性高一般都为post）

1. 文本框

<input>——将input标签内容嵌入到form标签内部

<input type=”类型” value=”取值” name=”” id=”” maxlength=”” size=”” checked=“checked”/>

|  |  |
| --- | --- |
| Input属性名称 | 说明 |
| Type | 指定表单元素的类型例如（Text、password、checkbox、radio、submit、Reset、File等，默认text） |
| Name | 给定表单元素名称 |
| ID | 给定表单元素ID（唯一） |
| Size | 指定表单元素的显示宽带（type为text或password时，以字符为单位来设置表单元素的显示大小 |
| Maxlength | 指定text或password表单元素的输入最大长度 |
| Checked | 只有一个值“checked”，表示被该属性修饰的 |
| 在使用下拉框时可以使用html5中type的date去使用日期等 | |

# WEB CSS

CSS样式表(Will.Zhu 朱文锦)

1. CSS基本语法
2. CSS不区分大小写
3. ‘;’分隔每条样式（一般多行，简单的可写成一行）
4. 注释/\*注释内容\*/
5. 结构

元素{

属性：值;

属性：值;

属性：值;

}

1. 基本属性

Visibility

1. CSS引用方式

a）内联式引用，直接把style写在标签内

e.g．

<p Style=”Color:red”>标签内容</p>

b）嵌入式引用，将style写在<head>标签中

e.g

<head>

<style></style>

</head>

c）外部引用（样式较多,写在\*.CSS文件中）

e.g

<link href=”style.css” rel=”stylesheet” type=”text/css” />

1. CCS选择器

标签选择器（例如针对p{}）

Id选择器，针对某个id=xxx（#xxx{}）

类选择器，针对某个Class=xxx（.xxx{}）

子选择器，针对元素的第一代孩子选择（xx>xx{}）

后代选择器，针对元素的所有孩子选择（xx xx{}）

通用选择器，所用的元素都生效（\*{}）

伪类选择器，a:hover{}(注有兼容问题)、a:link{}、a.visited{}等

分组选择器，（p,span,h1,h2,h3{}）

1. CSS特性

继承性（color、font等可以被继承、border不可继承）

权值性，在CSS样式重叠时，使用权值高低来判断（#id 的权值100、.Class的权值10、标签的权值1、继承的权值0.1）

层叠性，CSS多重引用方式，以最后的为主

1. 元素分类
2. 块状block

包含div、p、h1-h6、ol、ul、dl、table、blockquote；

每个块状元素占一行，其后的元素新起一行；

高度、宽带可以设置，默认为父容器的100%；

1. 内联inline

包含span、a、label、strong；

和其他的元素都在一行上；

元素的高度、宽带、顶底边距不能设置，宽度为内容的大小。；

c）块状内联inline-block）

包含img、input等；

和其他元素都在一行上；

元素的高度、宽带、顶底边距都能设置。

1. 盒子模型

边距：margin

填充：Padding

边框：border

1. 布局模型

布局方式（默认是Flow流体模式，从上自下垂直排列；默认块状元素宽度100%，内联元素都是从左到右）

可以通过float的形式进行调整。

1. 层模型

相对定位（相对自身位置）

Div{

Position:relative;

}

绝对定位（使用绝对坐标定位）

Div{

position: absolute;

}

# Web JavaScript

Java、javascript、JSP

Java：编译运行语言

Javascript：解释运行语言

（注：浏览器F12中Console中可以调试（Shift+Enter换行、定义变量、运行函数））

1. JS的执行流程

网页执行： a、用户从浏览器输入请求；

b、JS将请求包含的数据打包发送给远程服务器；

c、后台处理数据并返回；

d、浏览器接到返回值，并由JS重新对返回数据进行解包（解析）；

e、根据需求重新排版（布局）页面

1. JS的引入方式

—外部引入，在<head>标签中添加script

<script type=”text/javascrpt” src=”http://www.baidu.\*\*.js”/>

(尽量使用本地路径的js文件)

—内部嵌入，在任何地方（head、body）都可以使用JS

（数量不受限制）

1. JS变量
2. var关键字申明变量；
3. JS或Java中，大小写区分的。

（在使用Java和Java系（JS）的语言上，一般使用驼峰式命名规则，首字母小写，类的首字母大写）

1. JS函数

Note：<input type=”button” onclick=”functionName()”></input>

1. JS运算
2. 加运算“+”（字符、数值）
3. 减运算“-”（数值，不能转换为数值报错）
4. “\*”、“/”、”%”(求余)
5. “==”、”!=“、”!”、”===”（全等于、变量属性)、”!==”（不全等于）
6. “<”、”>”、“<=”、“>=”
7. “&&”(且)、”||”（或）
8. JS数据输出
9. alert()警告框
10. docment.write() 输出到HTML
11. xx.innerHTML
12. console.log
13. 数据类型

数值类型（整数、浮点、infinify、NaN）

字符串类型

布尔类型

Undefined类型

对象类型（object包含Array、Date、function）

1. JS内置对象

Number 原始数据类型的包装类

String 字符串数据类型的包装类

Boolean 布尔类型的包装类

Array 数组类型

Math 数学运算的工具类

Date 日期和时间的工具类

RegExp 正则表达式的包装类

Events 所有JS事件的包装类

Functions 函数类型的包装类

Window对象，表示浏览器创建的窗口

Screen对象，属于window对象，包含客户端显示器信息

History对象，属于window对象，包含了用户浏览历史记录

Document对象，包含整个HTML节点的dom树

This对象，代表当前类的类原型实例

1. JS创建对象

自定义类

数组

# HTML DOM

http://www.w3school.com.cn/htmldom/dom\_nodes.asp

1. Document

Document是当前html页面（文档）本身，在html的dom中，主要可以用于获取、修改、添加和删除元素或元素自身的属性（特征）

1. body：是<body>，如果使用了<iframe><frameset>(外层frame)；
2. domain：当前文档的域名；
3. open()、close()：打开、关闭一个新的文档；
4. write、writeln：向文档中写入；
5. cookie：以key-value进行保存；
6. 节点
   1. DOM节点

根据 W3C 的 HTML DOM 标准，HTML 文档中的所有内容都是节点：

•整个文档是一个文档节点

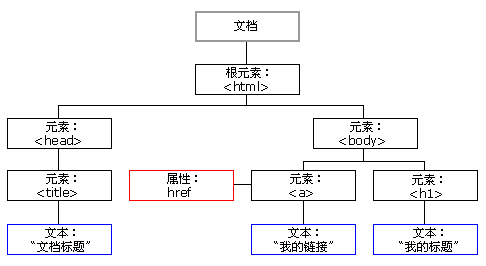
•每个 HTML 元素是元素节点

•HTML 元素内的文本是文本节点

•每个 HTML 属性是属性节点

•注释是注释节点

DOM 节点树



* 1. 节点关系

节点树中的节点彼此拥有层级关系——节点父、子和同胞。

父（parent）、子（child）和同胞（sibling）等术语用于描述这些关系。父节点拥有子节点。同级的子节点被称为同胞（兄弟或姐妹）。

•在节点树中，顶端节点被称为根（root）

•每个节点都有父节点、除了根（它没有父节点）

* 1. DOM方法

使用JS操作操作DOM，方法是我们可以在节点（HTML 元素）上执行的动作。

编程接口

可通过 JavaScript （以及其他编程语言）对 HTML DOM 进行访问。

所有 HTML 元素被定义为对象，而编程接口则是对象方法和对象属性。

方法是您能够执行的动作（比如添加或修改元素）。

属性是您能够获取或设置的值（比如节点的名称或内容）。

getElementById() 方法

getElementByClassName() 方法（IE8以下不能用）：

节点值（nodevalue）、nodeType

* 元素节点的 nodeValue 是 undefined 或 null
* 文本节点的 nodeValue 是文本本身
* 属性节点的 nodeValue 是属性值

Node.value (input等标签，内容取值)

* 1. DOM修改

修改 HTML DOM 意味着许多不同的方面：

* 改变 HTML 内容
* 改变 CSS 样式
* 改变 HTML 属性（node.属性=”值”、node.setattribute(“属性”,”值“)(IE8以下不支持)）
* 创建新的 HTML 元素（createElement、createTextNode、appendChild、insertbefore）
* 删除已有的 HTML 元素
* 改变事件（处理程序）

# Jquery

javascript库，提供快速处理html、

<script type=”text/javascript” src=”\*\*\*.js”></script>

1. Jquery语法

$=JQuery

1. Jquery选择器

和CSS选择器选择方式一致；

$(“标签名”)、$(“#ID”)、$(“.CLass”)

# JSON和XML

1. 简介

•JSON 指的是 JavaScript 对象表示法（JavaScript Object Notation）

•JSON 是轻量级的文本数据交换格式

•JSON 独立于语言 \*

•JSON 具有自我描述性，更易理解

\* JSON 使用 JavaScript 语法来描述数据对象，但是 JSON 仍然独立于语言和平台。JSON 解析器和 JSON 库支持许多不同的编程语言。

•XML 指可扩展标记语言（EXtensible Markup Language）

•XML 是一种标记语言，很类似 HTML

•XML 的设计宗旨是传输数据，而非显示数据

•XML 标签没有被预定义。您需要自行定义标签。

•XML 被设计为具有自我描述性。

•XML 是 W3C 的推荐标准

（**XML 被设计用来传输和存储数据。HTML 被设计用来显示数据。**）

1. XML

# 浏览器原理

WWW 的工作基于客户机/服务器计算模型，由Web 浏览器(客户机)和Web服务器(服务

器)构成，两者之间采用超文本传送协议（HTTP）进行通信， HTTP协议的作用原理包括四

个步骤：连接，请求，应答。根据上述HTTP协议的作用原理搜索，本文实现了GET请求的Web服

务器程序的方法，通过创建 TcpListener类对象，监听端口8080； 等待、接受客户机连

接到端口8080； 创建与socket字相关联的输入流和输出流;然后，读取客户机的请求信

息，若请求类型是GET，则从请求信息中获取所访问的HTML文件名，如果HTML文件存在，

则打开HTML文件，把HTTP头信息和 HTML文件内容通过socket传回给Web浏览器，然后关闭文件。否则发送错误信息给Web浏览器。最后，关闭与相应Web浏览器连接的socket 字。

一、HTTP协议的作用原理

WWW是以Internet作为传输媒介的一个应用系统，WWW网上最基本的传输单位是 Web网页。WWW的工作基于客户机/服务器计算模型，由Web 浏览器(客户机)和Web服务器(服务器)构成，两者之间采用超文本传送协议（HTTP）进行通信。HTTP协议是基于TCP/IP协议之上的协议，是Web浏览器和Web服务器之间的应用层协议，是通用的、无状态的、面向对象的协议。HTTP协议的作用原理包括四个步骤：

连接：Web浏览器与Web服务器建立连接，打开一个称为socket（套接字）的虚拟文件，此文件的建立标志着连接建立成功。

请求：Web浏览器通过socket向Web服务器提交请求。HTTP的请求一般是GET或POST命令（POST用于FORM参数的传递）。GET命令的格式为：

GET 路径/文件名 HTTP/1.0

文件名指出所访问的文件，HTTP/1.0指出Web浏览器使用的HTTP版本。

应答：Web浏览器提交请求后，通过HTTP协议传送给Web服务器。Web服务器接到后，进行事务处理，处理结果又通过HTTP传回给Web浏览器，从而在Web浏览器上显示出所请求的页面。

例：假设客户机与www.mycomputer.com:8080/mydir/index.html建立了连接，就会发

送GET命令： GET /mydir/index.html HTTP/1.0。主机名为www.mycomputer.com的Web服

务器从它的文档空间中搜索子目录mydir的文件index.html。如果找到该文件，Web服务器

把该文件内容传送给相应的Web浏览器。

为了告知 Web浏览器传送内容的类型，Web服务器首先传送一些HTTP头信息，然后传

送具体内容（即HTTP体信息），HTTP头信息和HTTP体信息之间用一个空行分开。

常用的HTTP头信息有：

① HTTP 1.0 200 OK

这是Web服务器应答的第一行，列出服务器正在运行的HTTP版本号和应答代码。代码

“200 OK”表示请求完成。

② MIME\_Version:1.0

它指示MIME类型的版本。

③ content\_type:类型

这个头信息非常重要，它指示HTTP体信息的MIME类型。如：content\_type:text/html

指示传送的数据是HTML文档。

④ content\_length:长度值

它指示HTTP体信息的长度（字节）。

关闭连接：当应答结束后，Web浏览器与Web服务器必须断开，以保证其它Web浏览器

能够与Web服务器建立连接。