# WEB

## 协议

### request

|  |  |
| --- | --- |
| 请求行 | 请求方法 空格 URL 协议版本CRLF |
| 消息报头headers【可选】 | Key：val CRLF  …keyval…  Key：val CRLF |
| CRLF |  |
| 消息正文（可选，用于POST) |  |

GET /abc?name=xxx HTTP/1.1

Host: 127.0.0.1:8080

User-Agent: Mozilla/5.0

### response

|  |  |
| --- | --- |
| 状态行 | 协议版本 状态码 状态说明CRLF |
| 消息报头headers【可选】 | Key：val CRLF  …keyval…  Key：val CRLF |
| CRLF |  |
| 消息正文（可选) |  |

HTTP/1.1 200 OK

Content-Length: 51

Content-Type: text/plain

空行

数据

### 响应状态码

1xx：指示信息--表示请求已接收，继续处理  
2xx：成功--表示请求已被成功接收、理解、接受  
3xx：重定向--要完成请求必须进行更进一步的操作  
4xx：客户端错误--请求有语法错误或请求无法实现  
5xx：服务器端错误--服务器未能实现合法的请求

### 数据类型

"txt": "text/plain",

"html": "text/html",

"png": "image/png",

"jpeg": "image/jpeg",

"jpg": "image/jpeg",

"xml": "text/xml",

"css": "text/css",

"gif": "image/gif",

"ico": "image/x-icon",

"js": "text/javascript",

"json": "application/json",

"pdf": "application/pdf",

"svg": "image/svg+xml",

"swf": "application/x-shockwave-flash",

"tiff": "image/tiff",

"wav": "audio/x-wav",

"wma": "audio/x-ms-wma",

"wmv": "video/x-ms-wmv",

application/json;charset=utf8 请求数据类型中包含字符编码

## HTML

<html>

<head>

<title>网页标题</title>

</head>

<body>

<h1>一级标题</h1>

段落文字

</body>

</html>

注释： <!-- \*\*\* -->

### 文本

<meta charset="utf-8">

中文支持

只保留空格，标签内文字写换行不起作用

<p> 段落标记 在前后都加入空行

<h1> 1级标题 一直到<h6>

<br/> 换行

<center> 使内容居中

<b>粗体

<i>斜体

<sup>上标

<sub>下标

<big>大号字体

<small>小号字体

<strong>加强语气

<del>删除线　　~~内容~~

<ins>下划线　　内容

<blockquote>　　大段文字　　 会自动插入换行和外边距

<bdo dir=”rtl”>hello</bdo> 倒序输出：olleh

预格式文本

保留标签内文字格式

<pre>

for(int i=0;i < 10;i++)

{

cout << i ;

}

</pre>

显示xml内容：

<xmp>

<!--这里放HTML代码-->

</xmp>

### 实体

保留字符，如< 使用&lt;

|  |  |  |  |
| --- | --- | --- | --- |
| **显示结果** | **描述** | **实体名称** | **实体编号** |
|  | 空格 | &nbsp; | &#160; |
| < | 小于号 | &lt; | &#60; |
| > | 大于号 | &gt; | &#62; |
| & | 和号 | &amp; | &#38; |
| " | 引号 | &quot; | &#34; |
| ' | 撇号 | &apos; (IE不支持) | &#39; |
| ￠ | 分（cent） | &cent; | &#162; |
| £ | 镑（pound） | &pound; | &#163; |
| ¥ | 元（yen） | &yen; | &#165; |
| € | 欧元（euro） | &euro; | &#8364; |
| § | 小节 | &sect; | &#167; |
| © | 版权（copyright） | &copy; | &#169; |
| ® | 注册商标 | &reg; | &#174; |
| ™ | 商标 | &trade; | &#8482; |
| × | 乘号 | &times; | &#215; |
| ÷ | 除号 | &divide; | &#247; |

### 区域

<div>定义一个区域

<span>定义文本区域

<**p**>abcd<**span**>XX</**span**>efghi</**p**>

### 列表

<ul>为无序列表 <ol>为有序列表

<ul>

<li type="disc">aaa</li>  
 <li type="square">bb</li>  
 <li type="circle">cc</li>

</ul>

Type默认为disc（实体圆）

自定义列表

<**dl**>  
 <**dt**>aaa</**dt**>  
 <**dd**>scriptAA</**dd**>  
 <**dt**>bbb</**dt**>  
 <**dd**>scriptBB</**dd**>  
</**dl**>

### 表格

<table border="1" width = 200 height = 100>

<caption>表格标题</caption>

<tr><th>1号 </th><th>2号 </th></tr>

<tr><td>1,1 </td><td>1,2 </td></tr>

<tr><td>2,1 </td><td>2,2</td></tr>

</table>

tr创建一行，th创建标题行内容，td创建单元格内容

<td>&nbsp;</td> 空单元格

colspan="4" 合并4列

可以在界面设置一个无边框的表格，在表格中放置内容进行排版

js中使用 \*\*.deleteRow(x); 删除第x行

### 属性

属性内容用 属性=”..”

如果属性值本身含有双引号，那么必须使用单引号。如：

name=’aaa”bbb”ccc’

#### 通用属性

|  |  |
| --- | --- |
| id | 唯一标识一个标记对，表示一个数据，对应css中 #\* |
| name | 可以重复 ，可以表示一组数据 |
| class | 对应css中 .\* |
| width  height | 元素宽高 "100%"表示占所在空间的比例(最外层元素则为浏览器宽度)，位置不够时会自动伸缩 |
| title | 当鼠标移动到标签时提示title内容 |
| align | "center"  img居中需要包含在div中  <div align="center"><img ></div> |

#### 标签属性

|  |  |
| --- | --- |
| bgcolor | 用于body  <body bgcolor="red"></body>  bgcolor=”#FFFFFF”bgcolor="red"  或者rgb(255,255,255)或者white  也可以用图片地址 |
| background | 设置背景图像，图像小的时候则重复铺入  <body background ="1.png" > |
| border | 用于table  Border属性为数字，0表示无边框  <table border=”1”> |
| Bool属性 | checked="checked" 或 checked 都代表勾选，不写代表默认不勾选 |

### 链接

address 标记中放置的文本显示为链接样式文字

<address>

<a href="http://www.xxx.com">网站名</a><br/>

</address>

#### 跳转属性

\_blank表示新页面打开

\_self当前页面打开

<a href=”....” target=”\_blank”>链接名</a>

#### 文本连接

<a href="b.html">本网站的一个页面</a>

#### 图像链接

<a href=”.......”> <img src=”a.jpg”/></a>

#### 页面内连接

<a name="s">开头</a>

<a href="#s">跳转到开头</a>

#代表当前页面

#### 页面嵌套

<embed src="#" width="600"></embed>

Src使用#代表当前页面，url可以显示其他网页

### 表单

#### 定义

<form id="f1">

======表单元素=====

<input type="text" id="inputContent" />

</form>

f1.\*\* 可以通过元素id或者name访问到元素

name或id可以标识表单，通过f1.inputContent.value 可以获取表单中输入数据，f1调用方式对于p这类标签不起作用

也可以使用 In= document.getElementById("inputContent");

#### 表单的提交

<form name="f1" action="b.html" method="get" target = "\_blank">

用户名:<input type="text" name="user"/>

<input type="submit" value="提交">

</form>

submit是提交按钮（特殊按钮），会调用form中的action

数据会传送到b.html页面中

target为打开新页面方式，默认为当前页面打开(\_self)，包括\_blank ，\_parent,\_top

method为post代表把输入数据按照http协议中的post方式传送到服务器，get将输入的数据追加到action指定的地址后面（在地址栏可以看到），并传送到服务器。

表单提交时会将form中所有表单元素内容进行提交（非表单元素或没有name属性不提交）

#### 按钮

<input type="button" onclick = "alert('xx');" value="按钮名"/>

<button onclick = "alert('xx');">btn</button>

#### 输入框

用户名:<input type="text" name="name"/>

密码：<input type="password" name="code"/>

#### 文件选择

<input type="file"/>

#### 单选按钮

<input type="radio" name="sex" value="male"/>Male

<input type="radio" name="sex" value="female"/>Female

获取选择内容

var sexval = document.getElementsByName("sex");

for(var i = 0; i < sexval.length;++i )

{

if(sexval[i].checked)

alert("check "+ sexval[i].value);

}

#### 组选按钮

<input type="checkbox" name="x"/>A

<input type="checkbox" name="x"/>B

#### 下拉列表

<select name="number" id=”nu”>

<option value="1">1</option>

<option value="2">2</option>

<option value="3" selected="selected">3</option>

</select>

获取数据：

var val = document.getElementById('nu').value;

或者

var val = document.getElementsByName(number)[0].value;

#### 标题框

<filedset>

<legend>信息</legend>

<form>

<lable>身高：<input type="text"/></lable>

</form>

</filedset>

Lable可以确保其中的内容为一行

#### 文本框

<textarea>多行文本，支持回车符 cols和rows设置区域大小

#### 重置输入内容

<input type="reset" value="重置"/>

重置form中的输入内容

### 图像

#### 图像显示

<img src=”位置” />

位置是相对位置，可以是当前目录相连的的目录，也可以是其它网页目录中的内容。

align=”bottom” 默认值

也可以是middle,top,left,right

<img src=”\*\*\*” alt=”\*\*\*”>

如果图像无法显示，则显示alt内的内容

#### 图像映射

为图像中的某些区域创建映射，使其可以点击跳转

<img src="x.jpg" usemap="#map1"/>

<map name="map1">

<area shap="rect" coords="x1,y1,x2,y2" href="b.html"/>

</map>

### base64

<img src='data:image;base64,\*\*\*base64string\*\*\*'>

### mate

刷新，content中包含刷新等待时间（秒），分号后写跳转到的页面地址

<meta http-equiv="refresh" content="0;2.jsp"/>

### 框架

可以在页面中嵌套其他网页

<**iframe src="https://\*\*\*\*"**></**iframe**>

a标签中的target属性的\_parent和\_top在框架中就可以体现出来。

### 功能

#### 刷新页面

1秒刷新一次

<meta http-equiv="Refresh" content="1">

### html5标记

#### 标签

|  |  |
| --- | --- |
| 区域（类似div） | 使用 |
| header | 一个内容区域或整个页面的标题 |
| article | 强调独立区域 |
| footer | 区域块的底部 |
| section | 强调同级别分类 |
| nav | 包含导航翻页等操纵链接 |
| aside | 用来表示当前页面或文章的附属信息 |
| hgroup | 对h1-h6的标签进行分组(将相关的放在一个hgroup下) |
| figure |  |

|  |  |
| --- | --- |
| input类型 | 使用 |
| email |  |
| url |  |
| number |  |
| range |  |
| date |  |

|  |  |
| --- | --- |
| 标签 | 使用 |
| canvas |  |
| video |  |
| audio |  |
| embed |  |
| mark |  |
| progress |  |
| meter |  |
| time | <time datetime="2017-01-01T20:00+01:00">2017-01-01</time>  datetime属性可以确保显示正确，+01：00表示时差多一个小时 |
| ruby |  |
| rt |  |
| rp |  |
| wbr |  |
| command |  |
| details |  |
| datagrid |  |
| keygen |  |
| output |  |
| source |  |
| menu |  |

#### 属性

|  |  |
| --- | --- |
| 属性 | 使用 |
| contenteditable | <p contenteditable="true"> xxx</p>  标记后元素内容在页面可编辑 |
| designMode | 设置可编辑状态，只能在js中用  document.designMode="on" |
| hidden | bool属性（不需要写内容），让元素隐藏 |
| spellcheck | 输入内容拼写检查 |

|  |  |
| --- | --- |
| **表单属性** | **使用** |
| **formaction** | <input type="submit" value="b1" formaction="f2.html"/>  可以指定submit按钮提交的action，而不用固定走外部form的action |
| **formmethod** | 指定form中提交按钮提交的方式(get,post) |
| **formtarget** | 指定form中提交按钮提交的打开新页面方式(\_blank等) |
| **autofocus** | 鼠标焦点设置，bool属性(不需要写内容) |
| **required** | 如果定义该属性的元素没有输入内容，提交时会提示，不允许提交 |

## CSS

### 行内样式

直接在标记对加属性

　<p style="font-family:宋体; color:Red;font-size:30px;text-align:center">

红色宋体30px居中显示</p>

### 内嵌式

style标记对中定义

定义ｐ标记对内容颜色为红色

<style>

p{color:Red;}

</style>

### 链接式

一般写在head中

<link rel="stylesheet" type="text/css" href="a.css" />

css文件

h1,h2

{

color:Blue;

font-size:small;

}

p{

color:Red;

}

### 选择器

#### class选择器

class在同一页面可以重复使用

.a{color:Red;}

.b{color:Blue;}

两个ｐ使用不同样式

<p class="a">xxxx</p>

<p class="b">yyyy</p>

#### id选择器

id在同一页面不能重复使用

#first{color:Red;}

#second{color:Blue}

iｄ不能使用数字

<p id="first">xxxx</p>

<p id="second">yyyy</p>

#### 多级选择

对于id为div1的区域中，选择p标签进行设置

**#div1 p**{  
 **color**: **aqua**;  
}

对于div标签，选择id为content\_1进行设置

**div#content\_1**{  
 **color**: **aqua**;  
}

### 布局

#### Div布局

<**body**>  
 <**div id = "content"**>  
 <**div id = "content\_head"**>标题</**div**>  
 <**div id = "content\_1"**>板块1</**div**>  
 <**div id = "content\_2"**>板块2</**div**>  
 <**div id = "content\_bottom"**>底部</**div**>  
 </**div**>  
</**body**>

Float让content\_1和content\_2从左开开始效果，最后clear属性取消浮动

Css中可以不声明div标签

**body**{  
 **margin**: 0**px**;  
}  
**div#content**{  
 **width**: 100%;  
 **height**: 800**px**;  
 **background-color**: **grey**;  
}  
**div#content\_head**{  
 **width**: 100%;  
 **height**: 10%;  
 **background-color**: **antiquewhite**;  
}  
**div#content\_1**{  
 **width**: 30%;  
 **height**: 80%;  
 **background-color**: **blueviolet**;  
 **float**: **left**;  
}  
**div#content\_2**{  
 **width**: 70%;  
 **height**: 80%;  
 **background-color**: **aqua**;  
 **float**: **left**;  
}  
**div#content\_bottom** {  
 **width**: 100%;  
 **height**: 10%;  
 **background-color**: **red**;  
 **clear**: **both**;  
}

#### Table布局

<**body marginheight="0px" marginwidth="0px"**>  
<**table width="100%" height="900px" style="background-color**: **gray"**>  
 <**tr**>  
 <**td colspan="2" width="100%" height="10%" bgcolor="antiquewhite"**>标题</**td**>  
 </**tr**>  
 <**tr**>  
 <**td width="30%" height="80%" bgcolor= "blueviolet"**>板块1</**td**>  
 <**td width="70%" height="80%" bgcolor="aqua"**>板块2</**td**>  
 </**tr**>  
 <**tr**>  
 <**td colspan="2" width="100%" height="10%" bgcolor="red"**>底部</**td**>  
 </**tr**>  
</**table**>  
</**body**>

## xpath

|  |  |
| --- | --- |
| 方法 | 说明 |
| nodename | 选取此节点的所有子节点 |
| / | 从根节点选取  从匹配选择的当前节点选择文档中的节点，只选择当前节点的子节点 |
| // | 从匹配选择的当前节点选择文档中的节点，不考虑它们的位置，选择子节点及子节点的任意深度节点 |
| . | 选取当前节点。 |
| .. | 选取当前节点的父节点。 |
| @ | 选取属性 |
| \* | 选择所有内容 |
| @\* | 选择所有属性 |
| //title/text() | 获取title标签包围的文字 |
| //title[@lang] | 选取所有拥有名为 lang 的属性的 title 元素。 |
| //title[@lang='eng'] | 选取所有 title 元素，且这些元素拥有值为 eng 的 lang 属性。 |
| //bookstore/book[price>35.00] | 选取 bookstore 元素的所有 book 元素，且其中的 price 元素的值须大于 35.00。 |