

更多嵌入式 Linux 学习资料, 请关注: 一口 Linux 回复关键字:1024



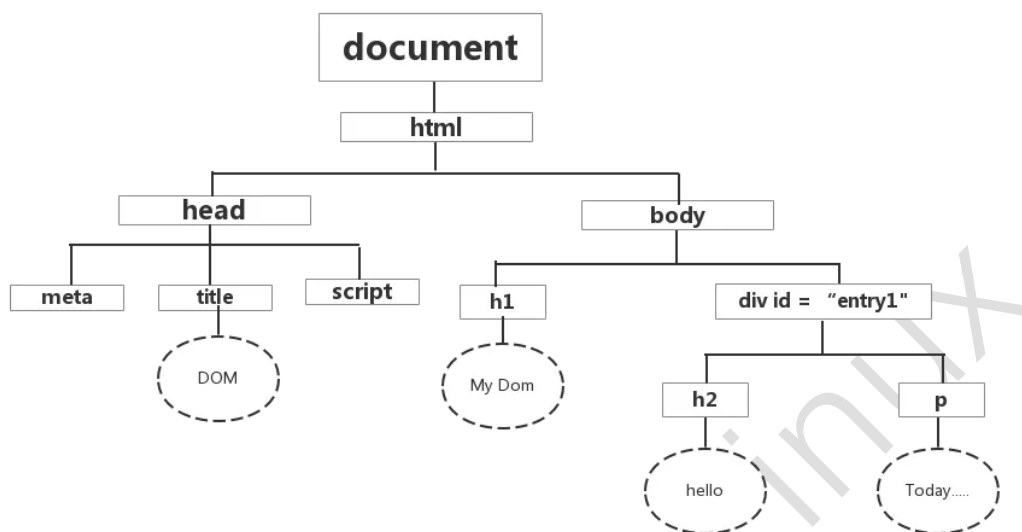
本文只针对想快速入门html的朋友, 通过本文, 读者可以快速学会如何编写基本的页面, 基本标签使用, 页面跳转等。  
所有操作都有亲自测试实例, 方便学习。

## 一、什么是 HTML 语言?

Html 是英文 HyperText Markup Language 的缩写, 中文意思是“超文本标志语言”, 用它编写的文件(文档)的扩展名是.html 或.htm, 它们是可供浏览器解释浏览的文件格式。

可以使用记事本、写字板或 FrontPage Editor 等编辑工具来编写 Html 文件。Html 语言使用标志对的方法编写文件, 既简单又方便, 它通常使用<标志名></标志名>来表示标志的开始和结束(例如<html></html>标志对), 因此在 Html 文档中这样的标志对都必须是成对使用的。

## 二、基础知识



### 1. HTML 语言的基本结构

<HTML> 文件开始

<HEAD> 标头区开始

<TITLE>...</TITLE> 标题区

</HEAD> 标头区结束

<BODY> 本文区开始

本文区内容

</BODY> 本文区结束

</HTML> 文件结束

举例

<html>

<head>

<title>江雪</title>

</head>

<body>

<center><b>江雪</b></center><br>

<center>千山鸟飞绝,</center><br>

<center>万径人踪灭.</center><br>

<center>孤舟蓑笠翁,</center><br>

<center>独钓寒江雪.</center>

</body>

</html>

将这一小段代码粘贴至文本文件中, 然后选择“另存为”, 将文件的后缀名改为 .htm 或者 .html 即可, 然后再所在的目录下就可看到一个 IE 的图标, 名字就是你所存的文件名称。

下图是执行效果:



### 江雪

千山鸟飞绝,

万径人踪灭.

孤舟蓑笠翁,

独钓寒江雪.

- html

```
<html>..... </html>
```

这是声明 HTML 文件的语法格式。每一个 HTML 文件, 都必须以开头, 以结束

- head

```
<head>..... </head>
```

这是文件头声明的语法格式。在这之内的所有文字都属于文件的文件头, 并不属于文件本体。

- title

```
<title>..... </title>
```

这是声明文件标题的语法格式。在这之中写下的所有内容, 都将写在网页最上面的标题栏中。

- body 体标签:要显示到浏览器中的主体文本

```
<body>..... </body>
```

- <center> </center> 居中
- <b>江雪 </b>

字体加粗

- `<br>`

换行标签:结束当前行,强制换行,它是一个空标签,没有结束标签,不要使用

- `<p></p>`

段落标签:在浏览器中显示段落,每一段落后会空一行。

- `<h1></h1>...<h6></h6>`

标题标签:标题显示标签从 h1 到 h6 逐渐减小字号,每一个标题标签后面系统会自动跟一个空行



## 一号标题

## 二号标题

### 三号标题

#### 四号标题

##### 五号标题

###### 六号标题

- 注释

在编辑 HTML 语言过程中,也可以使用注释。

语法格式为: `<!--文件注释-->`在这里插入代码片。

就好像 C 语言中的 `/* ..... */` 一样,中间的内容只是解释说明,并不被编译器所编译。

这是声明文件主体的语法格式。

在二者之间写下的内容都是文件的主体,也就是说将会被显示在客户区之中。

注意:几乎每一种 HTML 语言的语法都是以 `<>` 开头,以 `</>` 结束。

## 2. HTML 语言的基本单位

---

- 长度单位

长度单位可以用来定义水平线、表格边匡、图像等对象的长、宽、高等一系列属性, 同时也可以用来定义这些对象在页面上的位置等属性, 用来描述页面上可能遇到的各种长度。

长度的表示方法有两种: 绝对长度和相对长度。他们的单位都是像素(pixel)和百分比(%), 像素代表的是屏幕上的每个点, 而百分比代表的是相对于客户区的多少。下面我们就以水平线的宽度为例, 说明这两种表示方法。

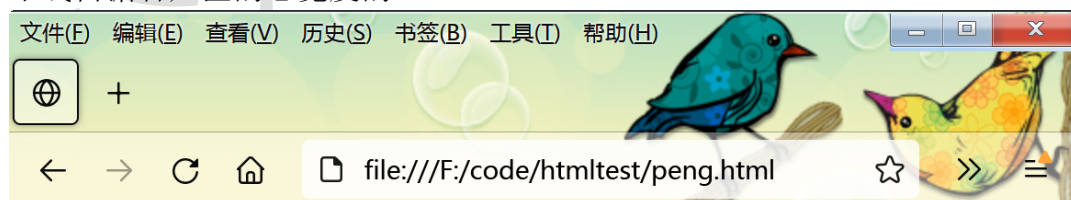
```
<html>
<head>
<title>HTML 语言的长度表示</title>
</head>
<body>
一口 Linux 欢迎您! HTML(HyperText Markup Language)是使用特殊标记来描述文档结构和表现形式的一种语言。
<hr width="500">    <!--绝对长度的声明-->
<hr width="50%">    <!--相对长度的声明-->
</body>
</html>
```

在文本编译器中编译, 改变网页的大小, 就会看到这两者表示长度方法的不同。

其中

标记是在页面上建立水平线的标记。width 是水平线元素中的一种属性, 用来表示水平线的宽度。

这里 width="500"即表示这个水平线的宽度是 500 像素; width="%50"即表示水平线占据客户区的总宽度的%50。



一口Linux欢迎您! HTML(HyperText Markup Language)是使用特殊标记来描述文档结构和表现形式的一种语言。

- 颜色单位 和长度单位一样, 颜色单位也是描述页面表现形式的一种很重要的数据类型。颜色单位有三种表示方法: **16 进制颜色代码**、**10 进制 RGB 码**、**直接颜色名称**。

这三种表示方法不同, 但是效果却是一样的。下面用一小段代码说明这三种颜色的表示方法。

```
<html>
<head>
<title>HTML 语言中颜色的不同表示方法</title>
</head>
<body>
<font color="red">静夜思</font><!--红色-->
<font color="#008000">窗前明月光</font><br><!--绿色-->
<font color="rgb(0,0,255)">疑是地上霜</font><br><!--蓝色-->
<font color="#ffff00">举头望明月</font><br><!--黄色-->
<font color="#800000">低头思故乡</font><!--栗色-->
</body>
</html>
```

执行结果:



注意: 本文件在保存时请选择 Unicode 字符型进行保存。

**16 进制颜色代码, 语法格式: #RRGGBB**。16 进制颜色代码之前必须有一个“#”号, 这种颜色代码是由三部分组成的, 其中前两位代表红色, 中间两位代表绿色, 后两位代表蓝色。不同的取值代表不同的颜色, 他们的取值范围是 00--FF。

**10 进制 RGB 码, 语法格式: RGB(RRR, GGG, BBB)**。在这种表示法中, 后面三个参数分别是红色、绿色、蓝色, 他们的取值范围是 0--255。以上两种表达方式可以相互转换, 标准是 16 进制与 10 进制的相互转换。

**直接颜色名称**, 可以在代码中直接写出颜色的英文名称。

以上三种表示方法的效果相同, 一般 16 进制颜色代码的表示方法比较常用。读者可以根据自己的喜好和实际情况来决定使用哪一种表示方法。

部分常用颜色代码:

`#ff0000` 红色的字

`#ff8000` 橙色的字

`#ffff00` 黄色的字

`#00ff00` 绿色的字

`#0080ff` 蓝色的字

`#0000a0` 靛色的字

`#8000ff` 紫色的字

`#000000` 黑色的字

`#c0c0c0` 灰色的字

### 3. URL 路径

---

URL(Universal Resource Locator)路径是一种互联网地址的表示法。

关于 URI 和 URL 概念, 可以参照下面这篇文章:

《[一文带你理解 URI 和 URL 有什么区别?](#)》

在这个数据里可以包括以何种协议连接、要连接到哪一个地址、连接地址的端口 (Port) 号以及服务器 (Server) 里文件的完整路径和文件名称等信息。

在 HTML 中, URL 路径分为两种形式: 绝对路径和相对路径。

- a. 绝对路径:

绝对路径是将服务器上磁盘驱动器名称和完整的能够写出来, 同时也会表现出磁盘上的目录结构。

语法格式: `<scheme>:<scheme_dependent_information>`

其中, 是某一种传输协议, 而 `<scheme_dependent_information>` 则是连接的位置信息。

例如:

`<a href="http://www.frontfree.net">`

`<a href="file:///D:/test/html.htm">`

- b. 相对路径: 相对路径是相对于当前的 HTML 文档所在目录或站点根目录的路径。

语法格式: 相对关系/部分路径/文件名



根据相对路径的参照点又可以分为相对文档的相对路径以及相对根目录的相对路径。

- <1>相对文档: 这种路径的表现形式是根据目标文档所在目录和当前文档所在目录之间的关系的一种表现形式。“../”表示上一级目录, 没有“../”表示当前目录。

例如当前文档的路径是: test/project1/index.htm。我们要找 test 目录下的 html.htm。而当前的目录是 project1, 我们要回到上一级目录中, 所以路径是 “../html.htm”。

- <2>相对根目录: 这种路径根据目标文档相对于根目录的关系的一种表现形式。在这种表达式种的第一个字符是“/”, 这个符号表示这个路径是一个相对于根目录的表达式。

例如:

```
<html>
<head>
<title>相对路径</title>
</head>
<body text="#ffff00" leftmargin="100">
<p><a href="../test/new.html">点击这里</a>。</p>
</body>
</html>
```

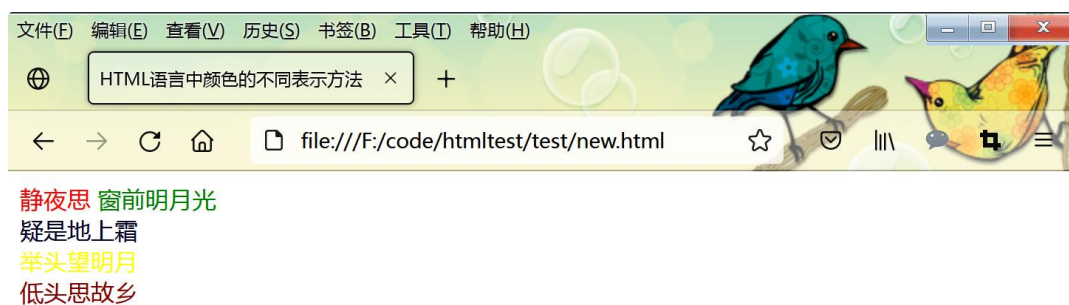


[点击这里](#).

在目录 test 下新建一个 new.html

点击[点击这里](#),





## 4. head 专题

head 这一部分是由标记开始, 以标记结束, 是 html 文档的首要部分。

head 部分所包含的元素:

元素	描述
title	文档标题
meta	描述非 html 标准的一些文档信息
link	描述当前文档与其他文档之间的连接关系
base	定义体试时默认的外部资源
script	脚本程序内容
style	样式表内容

### title

title 包含的内容将会被显示在浏览器窗口的标题栏中。给自己的网页起一个适当的名字, 可以体现出一些自己的个性。

```
<title> ..... </title>
```

在省略号处加入的文字或符号, 都将显示在浏览器的标题栏中。

### meta

meta>标签说明文档的自身信息. 它不会被浏览器显示给用户, 但却给软件程序和搜索引擎提供了关于文档的信息. 我们仅了解一下这个标签就足够了。

实例:

```
<!DOCTYPE HTML PUBLIC "-//D3C//DTD HTML 4.0 //EN"> -----文档类型,版本信息
<HTML>
<HEAD>
<TITLE>使用 meta 标签</TITLE>
<META NAME="Author" CONTENT="文档的作者信息"> ---文档的作者信息
<META HTTP-EQUIV="Expires" CONTENT="Fri ,31 Dec 2005 23:59:59 GMT">--Http 信息头,传递给服务器的文档附加信息,本行描述网页的过期时间
<META NAME="Description" CONTENT="This doc...">---文档内容描述
<META NAME="Keyword" CONTENT="music ,ice"> ---检索关键字
<META NAME="Copyright" CONTENT="&copy;2005 by lxz003">---版权信息
<META HTTP-EQUIV="Content-Language" CONTENT="en">----Http 信息头,描述文档的语言信息.
<META HTTP-EQUIV="REFRESH" CONTENT="20"> -Http 每 20 秒刷新一次网页
<meta http-equiv="content-type" content="text/html; charset=gb2312">--指定字符集
</HEAD>
<BODY>
</BODY>
</HTML>
```

## 定义语言

```
<meta http-equiv="content-type" content="text/html; charset="GB2312">
```

我们可以通过这样的语句设定语言的编码方式。这样, 浏览器就可以为我们正确的选择语言, 不需要我们来选取。上面的例子显示的就是简体中文, 如果想要显示繁体中文, 将 GB2312 替换为 BIG5 就可以了。

## 刷新页面

```
<meta http-equiv="refresh" content="60",URL="new.htm">
```

该语句可以让我们的浏览器按照 content 属性中制定的时间来跳转到 URL 属性中设定的 URL 地址。上面的例子就是在打开页面 60 秒后调用一个新的页面 new.htm。

如果没有能够找到 new.htm, 浏览器就返回一下错误:

```
404 Not Found
The requested URL /new.html was not found on this server.
```

## link

这个元素用来指定当前文档和其他文档之间的联接关系。 语法格式:

```
<link rel="描述" href="URL 地址">
```

rel 说明两个文档之间的关系; href 说明目标文档名。以下是一个指定外联样式表文件的例子:

```
<link rel="stylesheet" href="style.css">
```

## base

用于定义文档提交是默认的外部资源。 语法格式:

```
<base href="原始地址" target="目标窗口名称">
```

href 指定文档中链接到的所有文件默认的 URL 地址。在这里指定 href 的属性, 所有的相对路径的前面都会加上 href 属性中的值。

target 指定文档中所有链接的默认打开窗口。最常见的应用是在框架页 (frame) 中。我们要把 Menu 框架中的联接显示到 content 框架中, 就可以在 Menu 的网页中加上 `<base target="content">`。这样, 就省去了在 Menu 网页上所有链接的属性上加上 target 属性了。

## script

标记用来在页面中加入脚本程序。 语法格式:

```
<script language="脚本语言"> ..... </script>
```

在 language 中一定要指定脚本语言的种类。如 VBScript 等。

## style

用来指定当前文档的 css 层叠样式表。css 对于网页的字体样式、背景、边界等都有很大的应用。详细部分请大家参阅 css 有关内容。

## 5. body

---

body 是由开始, 由结束。

下面我们看一下 body 部分的元素:

元素	描述
bbgcolor	背景色
background	背景图案
text	文本颜色
link	链接文字颜色
alink	活动链接文字颜色
vlink	已访问链接文字颜色
leftmargin	页面左侧的留白距离
topmargin	页面顶部的留白距离

- bgcolor bgcolor 用于指定页面的背景颜色。在 Microsoft 公司的 IE 浏览器中, 默认情况下是白色, Netscape 公司的 Navigator 浏览器的默认情况是灰色。

语法格式:

```
<body bgcolor="颜色值">
```

- background: background 属性用于指定页面的背景图案。 语法格式:

```
<body background="URL">
```

示例:

```
<html>
```

```
<head><title>使用背景颜色</title></head>
```

```
<body bgcolor="#ff0000" text="#ffff00" leftmargin="100" background="linus.jpg">
```

```
<h4>热爱生命</h4>
```

```
<p> 我不去想是否能够
```

```
成功<br> 既然选择了远方
```

```
<br>
```

```
便只顾风雨兼程
```

```
</p>
```

```
<p> 我不去想能否赢得
```

```
爱情<br> 既然钟情于玫瑰
```

```
<br> 就勇敢地吐露真诚
```

```
</p>
```

```
<p> 我不去想身后会不会袭来寒风
```

```
冷雨<br> 既然目标是地平线<br> 留给
```

```
世界的只能是背影
```

```
</p>
```

```
<p> 我不去想未来是平坦还是泥泞
```

```
<br> 只要热爱生命<br> 一切, 都
```

```
在意料之中
```

```
</p>
```

```
</body>
```

```
</html>
```

执行结果:



下面介绍几种文字的属性, 可以改变文字的颜色, 以及在页面上留出空白:

```
text link alink vlink leftmargin topmargin
```

他们的作用分别是:

非链接文字的颜色; 可连接文字的颜色; 正被点击的可链接文字的颜色; 已被点击的可链接的文字的颜色; 页面左侧的留白; 页面顶部的留白。

语法格式:

```
<body text="color" link="color" alink="color" vlink="color">
```

```
<body leftmargin="value" topmargin="value">
```

实例:

```
<html>
```

```
<head><title>改变链接的颜色</title></head>
```

```
<body link="green" vlink="#800000" alink="#FF00FF">
```

```
<h4>如何找到我</h4>
```

```
<a href="https://space.bilibili.com/661326452">一口 Linux 的 B 站</a><br><br>
```

```
<a href="https://blog.csdn.net/daocaokafei">一口 Linux 的 CSDN</a><br><br>
<a href="https://gitee.com/yikoulinux">一口 Linux 的 Gitee 仓库</a><br><br>
<a href="https://www.zhihu.com/people/tudoujushi">一口 Linux 的知乎</a><br><br>
<a href="https://www.cnblogs.com/yikoulinux/">一口 Linux 的博客园</a><br><br>
<a href="https://www.toutiao.com/c/user/token/MS4wLjABAAAACqvbGwM1ITRA0LH07IzYz81D
cMYWmdITrNJHTAcBZU/?source=mine_home&tab=article">一口 Linux 的今日头条</a><br><br>
<body>
</html>
```

执行结果:



### 如何找到我

[一口Linux的B站](#)

[一口Linux的CSDN](#)

[一口Linux的Gitee仓库](#)

[一口Linux的知乎](#)

[一口Linux的博客园](#)

[一口Linux的今日头条](#)

说明: 对于 link 这样的属性, 大多数 html 的页面已经不使用这样的用法了。现在多数是在 head 中加入相应的代码。要达到的效果是鼠标放到有链接的字体上时, 字体变色, 点击后字体变为灰色。具体方法如下:

```
A:link {color:#0000ff;}
A:visited {color:#65038e;}
A:active {color:#ff0000;}
A:hover {color:#ff0000;}
```

将这段代码加入到 head 中, 在 body 中加入有链接的文字, 就可以看出效果了。这一部分是属于 css 中的知识。

在 body 中加入有链接的文字: 在 Dreamweaver 中, 选定文字后, 在下面的 Properties 中的 link 中加入想要链接到的页面的地址。在 FrontPage 中, 选定文字后, 点击右键就会有相应的选项。

## 6. 定制字体

font 标签允许你定制文字的大小, 颜色, 字体。

语法

```
<font></font>
```

属性有:

- color: 控制文字的颜色,使用颜色代码或 RGB 码,如生活的理想,就是 理想的生活!
- size: 控制文字的大小,html 设定了 1-7 共七种字号,1 号最小,7 号最大(比 H1 还要大),如要 显示比默认字体小一号可以设定 size="-1",比默认字体大一号可以设定 size="+1",我们也可 以使用比默认字体大一号或比默认字体小一号这两个标签是嵌 套在标签中的子标签.
- face: 控制文本的字体,设置文字的字体,如设置文字的字体使 用它的缺点是,你不能保证你的浏览器支持你指定的字体

实例

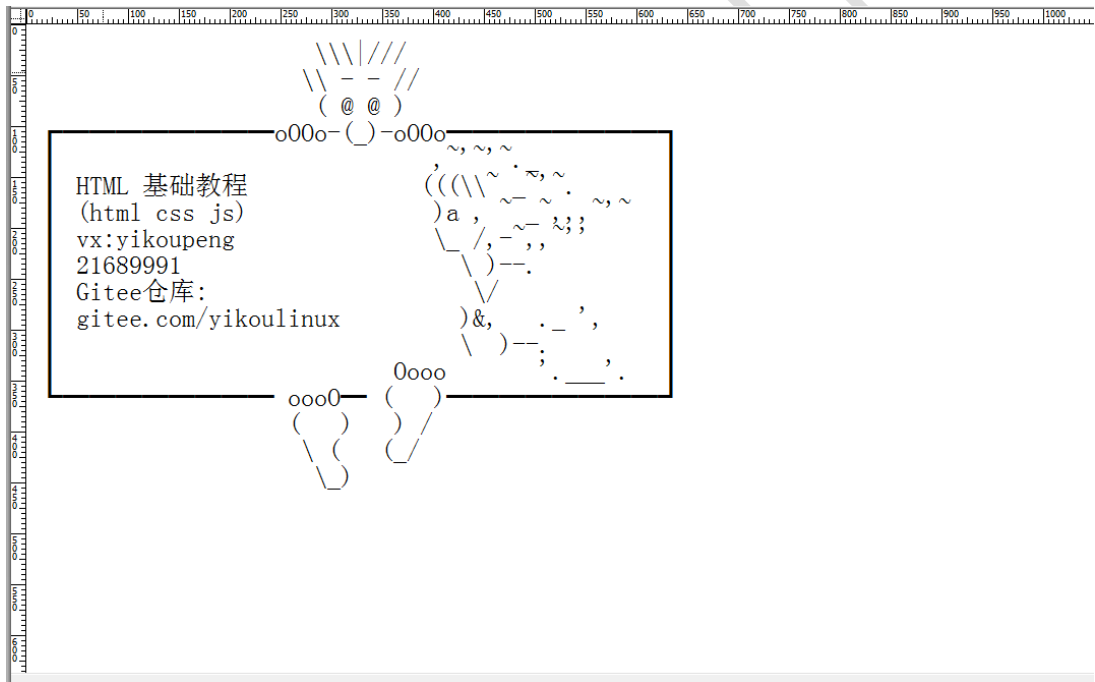
```
<html>
<head><title>定制我们的文字样式</title></head>
<body>
<font color="blue" face="Arial">既然选择了远方,便只顾风雨兼程!(正常大小)
english</font><br>
<font color="blue" size="+1" face="Comic Sans MS">既然选择了远方,便只顾风雨兼
程!(大一号) english</font><br>
<font color="blue" size="-1" face="Courier">既然选择了远方,便只顾风雨兼程!(小一
号) english</font><br>
<font color="blue" size="1" face="Courier New">既然选择了远方,便只顾风雨兼
程!(1 号) english</font><br><br><br>
<font color="blue" size="2" face="Modern">既然选择了远方,便只顾风雨兼程!(2 号)
english</font><br><br>
<font color="blue" size="3" face="MS Sans Serif">既然选择了远方,便只顾风雨兼
程!(3 号) english</font><br>
<font color="blue" size="4" face="MS Serif">既然选择了远方,便只顾风雨兼程!(4 号)
english</font><br>
<font color="blue" size="5" face="MS-DOS CP 932">既然选择了远方,便只顾风
雨!(5 号) english</font><br>
<font color="blue" size="6" face="Roman">既然选择了远方,便只顾风雨兼程!(6 号)
english</font><br>
<font color="blue" size="7" face="Times Roman">既然选择了远方,便只顾风雨兼
程!(7 号) eng</font><br>
<font color="blue" face="Times New Roman"><big>既然选择了远方,便只顾风雨
```





```
| HTML 基础教程      (((\\~_~'~.|
| (html css js)      )a , ~_~,,~'~
| vx:yikoupeng        \_ /,-~,~' '
| 21689991            \ )--.
| Gitee 仓库:         \ /
| gitee.com/yikoulinux )&, . _ ' ,
|                     \ )--.
|                     0ooo   ' . _ _ ' .
|
| ooo0- ( ) -
|          ( ) ) /
|          \ ( ( _ /
|          \_)
|
|</pre>
|</body>
|</html>
```

运行效果



## 9. SPAN

<span> 用于对文档中的行内元素进行组合。

<span> 标签没有固定的格式表现。当对它应用样式时, 它才会产生视觉上的变化。如果不对 <span> 应用样式, 那么 <span> 元素中的文本与其他文本不会任何视觉上的差异。

<span> 标签提供了一种将文本的一部分或者文档的一部分独立出来的方式。

```
<p><span>some text.</span>some other text.</p>
```

如果不对 span 应用样式, 那么 span 元素中的文本与其他文本不会任何视觉上的差异。尽管如此, 上例中的 span 元素仍然为 p 元素增加了额外的结构。可以为 span 应用 id 或 class 属性, 这样既可以增加适当的语义, 又便于对 span 应用样式。

可以对同一个<span>元素应用 class 或 id 属性, 但是更常见的情况是只应用其中一种。这两者的主要差异是, class 用于元素组(类似的元素, 或者可以理解为某一类元素), 而 id 用于标识单独的唯一元素。

## 10、UL

---

表标签分为:

- 无序列表:用黑点分隔,可以是闭环,开环,方框等多种风格,使用 <ul></ul> 标签.
- 有序列表: 按编号排序,比如数字,罗马字母,或英文大小写字母,使用<ol></ol> 标签.
- 定义列表:包含术语和定义,使用<dl></dl> 标签,嵌套术语标签<dt>,定义标签<dd>

普通的显示数据的时候, ul 就是项目列表, li 就是列表项。可以用来显示数据。

如果用于 DIV+CSS 布局的话, ul+li 可以替换表格的作用, 具体的设置, 如果宽度高度、行间距、背景边框等需要配合 CSS 一起设置。

### 1) 无序列表使用格式:

```
<ul>
<li></li>
<li></li>
</ul>
```

<ul>是无序列表标签, <li>嵌套其中表示列表项标签.

### type 属性

它们具有相同的 type 属性:

`type="circle"` 空心的圆点

`type="disc"` 实心的圆点

`type="square"` 实心方框

表的默认值为实心圆点。列表的重要的特定是可以嵌套的, 嵌套的列表为子列表, 子列表的缺省外观和主列表的外观是不一样的, 默认的形式是主列表是实心圆点, 第一级子列表使用空心的, 第二级子列表使用方框。

- 设置 div 水平横向排列的方法除了为所有 div 添加浮动外, 还可以通过设置 div 的属性 \* `display:inline-block;` 来实现, `display:block` 属性亦可以撑起整个块级 div ;
- ul 的无需裂变部分的圆圈可以通过设置 ul 的 `font-size` 实现, 同理还可以通过 `font-color` 改变圆圈的颜色;
- 去掉列表前面的小圆点: 设置 ul 属性"`list-style-type:none;`";
- 列表横向排列: 设置 li 属性"`float:left;`";
- 设置列表高度: 设置 ul 属性"`line-height;`";

## 2) 有序列表

有序列表和无序列表类似, 使用 `<ol>` 标签表示有序列表, 嵌套使用 `<li>` 表示列表项, 指定 `type` 属性为有序的数字或字母, 也可以对有序列表进行嵌套。

`type="1"` 使用数字, 缺省的情况

`type="a"` 使用小写字母

`type="A"` 使用大写字母

`type="i"` 使用小写罗马字母

`type="I"` 使用大写罗马字母

`type="X"` 开始的编号

`value ""` 可以改变当前值

### 实例 1

```
<html>
```

```
<head><title>使用无序列表</title></head>
```

```
<body>
```

```
<h3>有序列表的样式</h3>
```

```
<h5>默认样式</h5>
```

```
<ol>
```

```
<li>鼠标</li>
```

```
<li>键盘</li>
```

```
<li value="10">主机</li>
```

```
<li>显示器</li>
```

</ol>

<h5>使用字母</h5>

<ol type="a">

<li>个人 PC 机</li>

<li>笔记本</li>

<li>小型服务器</li>

<li>掌上电脑</li>

</ol>

<h5>使用大写字母</h5>

<ol type="A">

<li>无序列表</li>

<li>有序列表</li>

<li>定义列表</li>

</ol>

<h5>使用小写罗马字母</h5>

<ol type="i">

<li>中国</li>

<li>美国</li>

<li>日本</li>

<li>韩国</li>

</ol>

<h5>使用大写罗马字母</h5>

<ol type="I">

<li>萝卜</li>

<li>白菜</li>

<li>西红柿</li>

<li>南瓜</li>

</ol>

<h5>设定开始值</h5>

<ol start="5">

<li>罗湖区</li>

<li>宝安区</li>

<li>南山区</li>

<li>福田区</li>

<li>盐田区</li>

<li>龙岗区</li>

</ol>

```
<h5>有序列表和无序列表的嵌套</h5>
```

```
<ul>
```

```
<li>人体</li>
```

```
<ul>
```

```
<li>头</li>
```

```
<li>手</li>
```

```
<ul>
```

```
<li>手指</li>
```

```
<li>手掌</li>
```

```
</ul>
```

```
<li>腿</li>
```

```
</ul>
```

```
<li type="square">季节</li>
```

```
<ol type="I">
```

```
<li>春</li>
```

```
<li>夏</li>
```

```
<li>秋</li>
```

```
<li>冬</li>
```

```
</ol>
```

```
</ul>
```

```
</body>
```

```
</html>
```

## 实例 2:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta>
```

```
<title>ul、li 测试实例</title>
```

```
</head>
```

```
<body>
```

```
<ul>
```

```
<li><a href="http://www.baidu.com">Baidu</a></li>
```

```
<li><a href="https://gitee.com/yikoulinux">gitee</a></li>
```

```
<li><a href="https://blog.csdn.net/daocaokafei">CSDN</a></li>
```

```
<li><a href="https://www.cnblogs.com/yikoulinux/" class="last">博客园</a></li>
```

```
</ul>
```

```
</body>
```

```
</html>
```

- [Baidu](#)
- [gitee](#)
- [CSDN](#)
- [博客园](#)

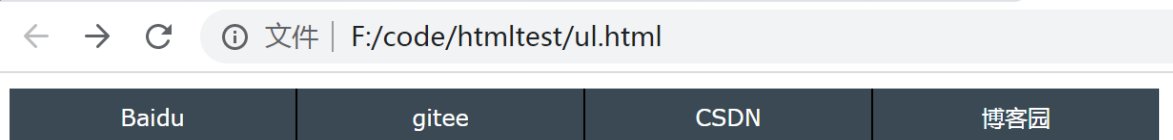
上述例子可以用 `style` 设置下格式:

```
<html>
<head>
<meta>
<title>图片提示效果</title>
<style type="text/css">
    #menu {
        font:12px verdana, arial, sans-serif; /* 设置文字大小和字
体样式 */
        width: 100%;
    }
    #menu, #menu li {
        list-style:none; /* 将默认列表符号去掉 */
        padding:0; /* 将默认的内边距去掉 */
        margin:0; /* 将默认的外边距去掉 */
        float: left; /* 往左浮动 */
        display: block;
    }
    #menu li a {
        display:inline-block; /* 将链接设为块级元素 */
        width:150px; /* 设置宽度 */
        height:30px; /* 设置高度 */
        line-height:30px; /* 设置行高, 将行高和高度设置同一个值, 可以让单行
文本垂直居中 */
        text-align:center; /* 居中对齐文字 */
        background:#3A4953; /* 设置背景色 */
        color:#fff; /* 设置文字颜色 */
        text-decoration:none; /* 去掉下划线 */
        border-right:1px solid #000; /* 在左侧加上分隔线 */
    }
    #menu li a:hover { /* 鼠标放在对应文字图标上 */
        background:#146C9C; /* 变换背景色 */
        color:#fff; /* 变换文字颜色 */
        font-weight: bold;
```



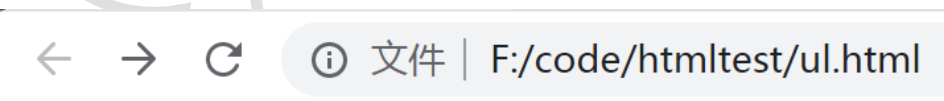
```
}
#menu li a.last {
border-right:0; /* 去掉右侧边框 */
}
</style>
</head>
<body>
<ul id="menu">
<li><a href="http://www.baidu.com">Baidu</a></li>
<li><a href="https://gitee.com/yikoulinux">Gitee</a></li>
<li><a href="https://blog.csdn.net/daocaokafei">CSDN</a></li>
<li><a href="https://www.cnblogs.com/yikoulinux/" class="last">博客园
</a></li>
</ul>
</body>
</html>
```

运行结果:



如果要成竖排排列: 修改如下参数:

```
#menu {
font:12px verdana, arial, sans-serif; /* 设置文字大小和字体样式 */
width: 10%;
}
#menu li a.last {
border-right:1; /* 去掉右侧边框 */
}
```



## 三、table 表格

### 1. 表格的一般标签和属性

`<table>`: 创建表格

`<caption>`: 表的标题(可选的), 表格内容的描述信息

`<tr>`: 表的行, 包括表头和表的数据行

`<td>`: 表的列

`<th>`: 表头(可选的)会用黑体显示

#### 1) 设定表格的边框

上面的表格是没有边框的, 我们使用 `border` 属性可以为表格增加边框`<table border>` (默认 宽度为 1) 或设定边框的值`<table border="10">`, 使用 `border="0"` 相当于没有边框.

#### 2) 设定表格的对齐方式

使用 `align` 可以设定表格的对齐方式, 缺省是左对齐的.

`align="left"` 左对齐

`align="right"` 右对齐

`align="center"` 居中

#### 3) 设定表格和单元格的宽度和高度

使用 `width` 可以设定表格的宽度, 使用像素值 `width="80"`, 也可以使用百分比,

如 `width="100%"` (屏幕宽度), `width="50%"` (屏幕一半的宽度), `width="150%"` (屏幕一倍半的宽度, 会出现滚动条)

使用 `height` 可以设定表格的高度, `height="150"`

在`<td>``<th>`中使用 `width` 和 `height` 属性可以设置单元格的宽度和高度.

#### 4) 控制单元格距离 `cellspacing`、`cellpadding`

`cellspacing`:设置单元格间的间距

`cellpadding`:设置单元格内的间距

间距使用像素值.

## 5) 表格中数据的对齐方式 `align`、`valign`

对于`<tr><td><th>`标签可以使用 `align` 确定文本的水平对齐方式:

`align="left"` 左对齐 (默认)

`align="right"` 右对齐

`align="center"` 水平居中

使用 `valign` 设定垂直方向上的对齐方式

`valign="top"` 顶端对齐

`valign="middle"` 垂直居中

`valign="bottom"` 底部对齐

`valign = "baseline"` 基线对齐(和顶端对齐方式类似,但要求当字体大小不一致时要保持 底线是对齐的. 当单元格内的对齐方式和表格行的对齐方式有冲突时,单元个内的对齐方式优先.

## 6 ) 为表格设置背景颜色

在`<table><tr><td><th>`中可以使用 `bgcolor` 设置表格或单元格的背景颜色

```
<table bgcolor="red">,<tr bgcolor="#00FF00">
```

## 7) 禁止单元格中的文本折行

在`<td>`和`<th>`中使用 `nowrap` 可以是单元格内的文本在一行显示,不会折行显示.

不要期望在`<tr>`中加入 `nowrap` 使一行单元个不折行,`nowrap` 对`<tr>`是不起作用的.

## 8) 合并单元格

在和中使用 `colspan` 可以合并列,使用 `rowspan` 可以合并行.

合并 2 列,2 行.### 9) 表格的嵌套 表格的嵌套是把一张表插入到另外一张表中.

```
<table>
<tr>
  <td>
    <table>
      <tr>
        <td></td>
      <tr>
    </table>
  </td>
  <td>
  </td>
</tr>
</table>
```

## 10) 为表格和单元格加入背景

在<table>中使用 background 属性可以设置表格的背景图片,  
在<td>或<th>中使用 background 属性也可以为单元格设置背景图片

## 11) 使用在 width 中使用”\*”

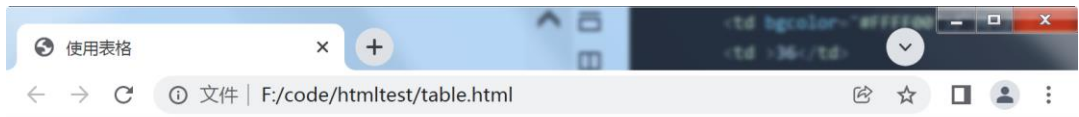
在 width 属性中我们可以指定像素值, 使用百分比,  
也可以使用”\*”来表示宽度, <col width=”\*”>表示占满全部可用的空间,  
如果两列都使用”\*”, 表示各占 50%宽度.  
使用”0\*”表示占用最小的宽度, 使用”1\*”和”2\*”等表示,  
如果有两列, 一列用”1\*”, 另外一列用”2\*”表示第一列占 1/3, 第二列占 2/3

## 2. 实例

---

```
<html>
<head><title>使用表格</title></head>
<h4>显示简单的表格</h4>
<table border="1" width="600" cellspacing="1" cellpadding="22" background="image/
backGround1.jpg">
  <!-- 这里是表的标题信息, 表的标题信息是可选的 -->
  <caption>全国各大城市 8 月 8 日气温</caption>
  <!-- 表的第一行, 表头也是可选的 -->
  <tr>
    <th width="100">城市</th>
```

```
<th width="200">最高气温</th>
<th width="300">最低气温</th>
</tr>
<!-- 表的数据信息 -->
<tr>
<td>深圳</td>
<td>35</td>
<td>29</td>
</tr>
<tr>
<td bgcolor="#FFFF00">广州</td>
<td>36</td>
<td>28</td>
</tr>
<tr bgcolor="#00FF00">
<td>北京</td>
<td>29</td>
<td>25</td>
</tr>
<tr>
<td>长春</td>
<td>27</td>
<td align="center">19</td>
</tr>
</table>
</html>
```



显示简单的表格

全国各大城市8月8日气温

城市	最高气温	最低气温
深圳	35	29
广州	36	28
北京	29	25
长春	27	19

四、创建有声有色的网页

1、img 网页中使用图片

用错误!未指定文件名。标签可以在你的网页中包含图片。语法：

```

```

使用 alt 属性, 可以在浏览器不能显示图片时显示替代的文本信息, 比如：

```

```

2、图像的对齐方式

对齐方式上区分图像有两种方式：行间图像:在文本行中的图片, 如果图片很大, 文本行会很高, 而且有一大片空白。浮动图像: 文本在图像边会换行, 图像很大 时, 文字会围绕图片进行自动换行。行间图片的对齐方式：

`align="top"` 图像和文本的顶部对齐.

`align="middle"` 图像和文本的中部对齐.

`align="bottom"`, 缺省值, 图像和文本的底部对齐.

浮动图像的对齐方式:

`align="left"` 图片在文本的左边

`align="right"` 图片在文本的右边

使用 `<br clear="left">` 可以在图片的右侧产生空白.

`<br clear="right">` 可以在图片的左侧产生空白.

`<br clear="right">` 可以在图片的两侧产生空白

### 3. 使用 `width,height` 设置图片的尺寸

用 `width`(像素)宽度, `height`(像素)高度可以设置图像的显示尺寸, 设定图像的尺寸, 浏览器会很容易的显示图片. 显示速度会加快.

```

```

### 4. 设置边框 `border`

们使用 `border` 属性为图片增加边框, `border="0"` 是没有边框, 我们没有必要设置, 系统默认是没有边框的. 我们上面设置图片的尺寸是图片的实际尺寸, 是不包括边框的尺寸的. 设定图片的边框为 1 可以使用 `border="1"`。

### 5. `hspace, vspace` 调整图片的位置

`space`: 在图片的左侧和右侧添加空白, 使用的是像素值. `vspace`: 在图片的上下方向上添加空白, 使用的是像素值.

```

```

### 6. 使用图像作为链接

可以使用图片作为链接: `<a`

```
<a href="http://www.google.com"></a>
```

### 7. 实例



```
<HTML>
```

```
<head><title>在网页中使用图片</title></head>
```

```
<body>
```

```
<h4>在网页中使用图片</h4>
```

```
<img src ="linus.jpg" alt="linux 大佬"><br>
```

```
<hr>
```

行间图片的对齐方式<br>

和文字的顶部对齐:

```
<img src ="linus.jpg" alt="linux 大佬" align="top"><br>
```

和文字的中部对齐:<img src ="girl.jpg" alt=" linux 大佬 "

```
align="middle"><br>
```

和文字的底部对齐:<img src ="linus.jpg" alt="linux 大佬" align="bottom">

```
<hr>
```

浮动图片的对齐方式<br>

```
<img src ="linus.jpg" alt="linux 大佬" align="left">左对齐:左对齐:左对齐:
```

左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:

对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:

使用 clear 属性

```
<br clear="left">
```

```
<hr>
```

```
<img src ="linus.jpg" alt="linux 大佬" align="left">左对齐:左对齐:左对齐:
```

左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:

左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:左对齐:

没有使用 clear 属性

```
<hr>
```

在此处使用 clear 属性:

```
<br clear="left">
```

```
<img src ="linus.jpg" alt="linux 大佬" align="right">
```

图片在右侧:图片在右侧:图片在右侧:图片在右侧:

图片在右侧:图片在右侧:图片在右侧:图片在右侧:

图片在右侧:图片在右侧:图片在右侧:图片在右侧:

图片在右侧:图片在右侧:图片在右侧:图片在右侧:

图片在右侧:图片在右侧:图片在右侧:图片在右侧:

图片在右侧:图片在右侧:图片在右侧:图片在右侧:

图片在右侧:图片在右侧:图片在右侧:图片在右侧:

在此处使用 clear 属性

```
<br clear ="right">
```

```
<hr>
```

设置图片的尺寸为 50X40:

```

```

```
<hr>
```

带边框的图片:

```
<hr>
```

调整图片的位置:<br>

```

```

```
<hr>
```

使用图像作为链接:

```
<a href="http://www.sohu.com"></a>
```

```
</body>
```

```
</html>
```

## 五、frameset 框架页面

frameset 可以在某一个页面包含多个 html 页面。frameset 文档和普通的 html 文档稍有不同, 普通的 html 文档包含<head>和<body>元素,

### 1. rows 和 cols 水平、竖直分割

- 1) <frameset>包含一个或多个<frame>标签, frameset 使用 rows 和 cols 来水平和竖直分割页面。
- 2) <frame>标签使用 src 属性指定在此框架中要显示的文档。
- 3) 如果浏览器不支持框架会显示<noframes>标签中的内容。
- 4) <frameset>标签中的 cols 属性可以垂直分割窗口, 定义两个或多个垂直框架的尺寸, 各框架的尺寸值用", " 分隔. cols 的缺省值是 100%, 若没有指定任何值, 就会只有一个框架占据整个水平窗口。

每个框架的宽度可用是绝对宽度(使用像素或百分比), 也可以是相对宽度(使用数字和" ", 如 2)。

分隔原则是绝对值具有较高的优先级, 最后按相对宽度划分剩余的空间。

```
<frameset cols="25%,75%">
```

```
<frameset cols="1*,3*">
```

```
<frameset cols="100,25%,2*,3*">
```

第一个占用 100 个像素, 第二个占用整个的 25%, 第三个占剩下的 1/4, 第四个占剩下的 3/4

- 5) rows 的用法和 cols 类似, 它进行水平分割窗口, 也可以指定绝对和相对高度, 这里的高度为整个窗口的高度
- 6) frame 标签中使用 src 属性插入要在该框架中显示的文档. <frame src="插入的页面.html">

- 7) 用 `frameborder` 设置是否显示边框 `<frame>` 标签中设置 `frameborder` 属性可以决定是否显示框架与相邻的框架之间的边框是否显示, 默认值为 1, 显示, 设为 0 表示不显示. 如果要设置边框的尺寸可以在 `<frameset>` 中的 `border` 属性中设置

不显示边框

```
<frame frameborder="0" src="first.html">
```

- 8) `marginheight`, `marginwidth` 设置页面在框架中的位置

`<frame>` 标签中使用 `marginheight`, 和 `marginwidth` 属性可以设定显示的页面距离框架的上下距离和左右距离.

```
<frame src="first.html" marginheight="100" marginwidth="200">
```

- 9) `name` 属性 用 `name` 属性来命名框架, 在其它框架中的链接可以使用这个名字来指向这个框架.

```
<frame src="first.html" name="first">
```

- 10) `bordercolor` `frameset` 和 `frame` 中使用 `bordercolor` 属性可以设定边框的颜色, 缺省的颜色是灰色.

```
<frameset bordercolor="#FF0000">,<frame bordercolor="#00FF00">
```

•

11. `<noframes>` 为不支持框架的浏览器提供替代内容 `<noframes>` 标签为不支持框架的浏览器提供了替代的内容, 当浏览器不支持框架时会显示 `<noframes>` 标签内的内容:

## 2. 实例 1:

```
<html>
```

```
<head><title>最简单的框架</title></head>
```

```
<frameset rows="200,*" border="12" framespacing="0">
```

```
<frame src="first.html" />
```

```
<frameset cols="200,*" border="5" rows="612" >
```

```
<frame src="second.html"/>
```

```
<frame src="three.html"/>
```

```
<body>
```

```
你的浏览器不支持框架!<br>
```

```
<a href="first.html">文档 1</a> <br>
<a href="second.html">文档 2</a> <br>
<a href="three.html">文档 3</a> <br>
</body>
</noframes>
</frameset><noframes></noframes>
</html>
```

first.html

```
<html>
<head><title>first html</title></head>
<body><font color="red">第 1 个 Html 文档</font></body>
</html>
```

second.html

```
<html>
<head><title>second html</title></head>
<body><font color="blue">第 2 个 Html 文档</font></body>
</html>
```

three.html

```
<html>
<head><title>three html</title></head>
<body><i>第三个 Html 文档</i></body>
</html>
```

执行结果:



## 第1个 Html 文档

第2个 Html 文档

第三个 Html 文档

## 3. 预定义的目标框架名

在 html 中预定义了四个目标框架名, 它们都以下划线开头, 并且区分大小写.

### 1) blank

blank: 当 `target="_blank"` 时, 浏览器会新开窗口显示链接指定的信息.

```
<a href="first.html" target="_blank">new windows</a>
```

单击 new windows 链接时会在新窗口中显示内容.

### 2) parent

在父框架中显示, 如果没有父框架会在当前框架中显示信息.

```
<a href="first.html" target="_parent">display at parent windows</a>
```

### 3) \_self

在当前框架中显示, 它是配合使用

比如我们在<head>部分义了<base target="main">, 我们想让其中某个链接在当前窗口而不是 main 框架中打开, 我们可以使用:

```
<a href="first.html" target="_self">current windows</a>
```

## 4. 实例 2:

---

导航框架.html

```
<html>
<head><title>使用 name 属性制作导航框架</title></head>
<frameset rows="100,*" border="1" framespacing="0">
<frame src="top.html" />
<frameset cols="25%,85%" >
<frame src="navvar.HTML" >
<frame src="welcome.html" name="main">
</frameset>
</frameset>
</html>
```

top.html

```
<html>
<head>
<title>top</title>
</head>
<table width="100%" border="0">
<tr>
<td width="1300" height="84" background="image/top.png">
<table width="800" height="45" border="0" cellpadding="5" cellspacing="4" align="right">
<tr>
<td><font color="#999900" face="微软雅黑" size="+3">无线传感网 Web 管理系统!</font></td>
</tr>
</table></td>
</tr>
</table></td>
</body>
</html>
```

navvar.html

```
<html>
<head><title>导航条</title></head>
<body>
<b>一口 Linux</b><br>
<ul>
<li><a href="first.html" target="main">first</a></li>
<li><a href="second.html" target="main">second</a></li>
</ul>
</body>
</html>
```

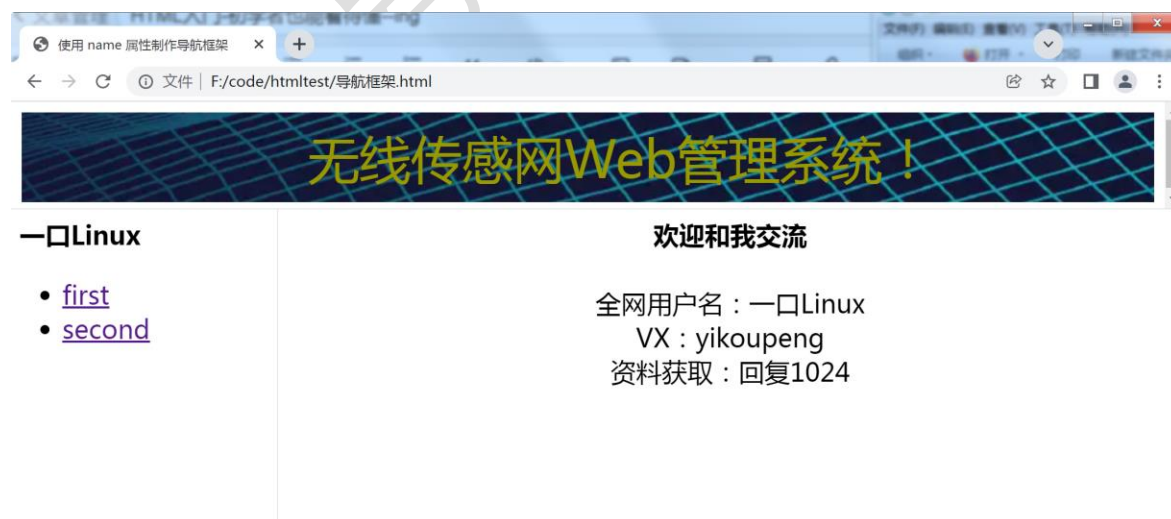
welcome.html

```
<html>
<head><title>欢迎页面</title></head>
<body>
<center><b>欢迎和我交流</b></center><br>
<center>全网用户名: 一口 Linux</center>
<center>VX: yikoupeng</center>
<center>资料获取: 回复 1024</center>
</body>
</html>
```

image/top.png



执行结果:

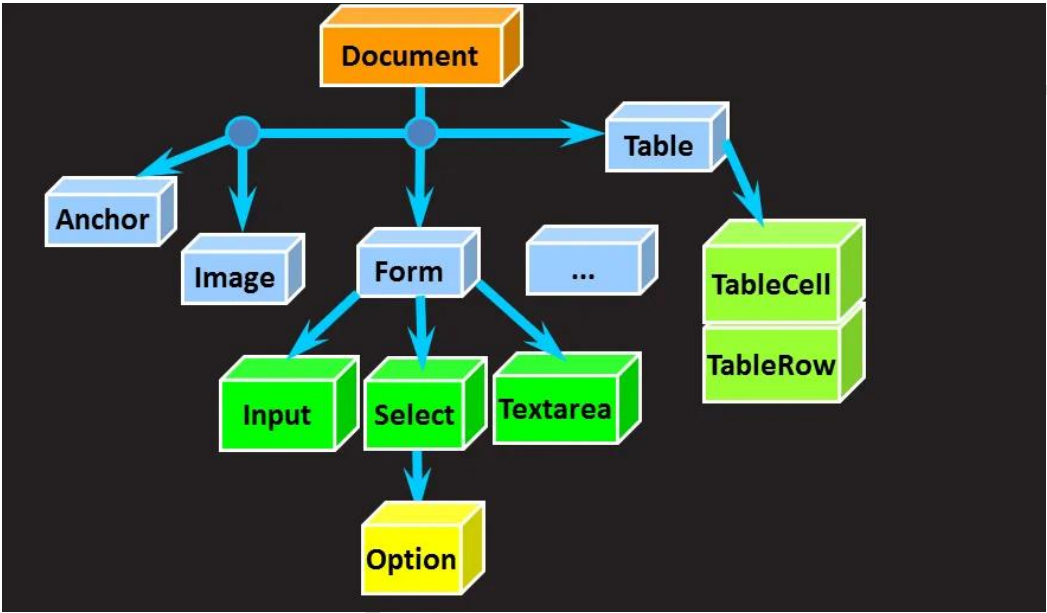




## 六、表单

web 应用程序经常要收集用户的信息, 如登陆名, 密码, Email 信息等, 在 Html 中使用表单来收集这些信息. 表单是信息域的集合, 这些信息域有着不同的表现形式, 比如, 文本框, 单选按钮, 下拉框等。

`<form>` 表单标签, 一个页面中可以含有多个表单标签, 被 `<form>` 包含的是一组要输入和提交给表单程序处理的信息.



### 1. 表单控件标签分类

类型	说明
input	文本框或按钮
textarea	多行文本区
select	下拉框

### 2. `<input>` 标签的使用

`<input>` 标签中的 `type` 属性指定表单控件的类型, `name` 属性用来标识当前的表单控件, 通过 `name` 引用表单控件, 获得表单控件的值. 没有 `name` 属性将无法提交任何值.

## 1) type

type 类型共有 10 种不同的类型:

类型	说明
text	创建单行文本框 用 value 指定默认值,size 指定文本框的大小,maxlength 指定文本框最多能输入的字符数.
password	密码框和 text 类似,但输入内容不直接显示在屏幕上用"*"屏蔽.
checkbox	复选框,可以同时选择多项.
radio	单选框,一次只能选择一项
submit	提交按钮,提交表单 image:图片形式的提交按钮
reset	重置按钮,把用户输入的信息置为初始值
button	创建一个按钮,但不能自动提交
hidden	隐藏域,不显示在屏幕上,可以用来收集不需要用户看到的信息
file	文件上载,发送客户端的文件

## value、size、maxlength

类型	说明
value	为文本框设定初始值.
size	指定文本框显示的长度,按英文字符个数为单位.
maxlength	允许输入的最大英文字符数
readonly /disable	我们可以使用 readonly 把单行文本框变成只读不允许用户改变初始值,使用 disable 可以把文本框变灰,处于不可用状态

## 3) checkbox 复选框

要使复选框在初始状态下选中, 使用 checked 属性:

```
<input type="checkbox" name="" value="" checked>
```

只有被选中的复选框在提交时, 其数据才被提交

#### 4) radio 单选按钮

```
<input type="radio" name="" value="" >
```

表示单选按钮, 在一组单选按钮中只能选中一个按钮, 如果设置按钮默认是选中状态使用 checked 属性, 和复选按钮一致

#### 5) submit 提交按钮

```
<input type="submit" value="" name="" >
```

value 属性是按钮上的显示值, 提交按钮的作用是提交表单数据, 提交表单可以放在表单的任何位置, 而且一张表单可以有多个提交按钮, 使用 size 属性可以控制按钮的尺寸

#### 6) image 图形提交按钮

```
<input type="image" src="XX.gif" alt="" align="">
```

图形提交按钮的功能和提交按钮的功能类似, 以图片代替灰色的按钮, 属性和我们前面讲的 img 的具有的属性类似. src 指定使用的图片, alt 显示提示文本, align 指定对齐方式

#### 7) reset 重置按钮

```
<input type="reset" value="" name="" size="">
```

重置按钮的作用是把表单中的数据置为初始值, 用户填入的数据会放弃, 我们在使用重置按钮时, 尽量不要把重置按钮放在提交按钮前, 以免用户不小心按了重置按钮丢失已经填入的数据

#### 8) hidden 隐藏域

```
<input type="hidden" name="" value="" >
```

隐藏域和单行文本框差不多, 但是隐藏域的信息是不显示给用户的, 用户没有机会改动隐藏域的信息, 隐藏域是为了收集程序需要的信息, 在提交时一起提交给程序处理

## 9) button 按钮

```
<input type="button" value="" >
```

创建一个按钮, 没有缺省的行为事件, 当使用 `onclick="XX()"` 时会在点击时执行 `XX` 事件, 我们在脚本中去实现 `XX()` 方法

## 10) file 文件上传对话框

```
<input type="file" name="">
```

显示文件上传对话框, 提交数据时会一起提交

## 11) select 下拉框

使用 `<select>` 标签可以创建下拉框, `<select>` 标签至少要嵌套一个 `<option>` 选项标签, 选项 标签会在下拉框中创建一个供选择的条目, 在 `<select>` 标签中指定 `size` 属性下拉框便成为列表

```
<select name="major" size="10" multiple>
<option value="1" >嵌入式
<option value="2" selected>JAVA
<option value="3" >大数据
</select>
```

## 12) button 按钮

`<button>` 的 `type` 属性可以是

```
button(普通按钮)
reset(重置按钮)
submit(提交按钮)
```

和 `input` 标签中的按钮类似, 但 `<button></button>` 中可以嵌套图片, 表格, 文本等作为按钮内容, 显示为一个按钮

## 13) readonly 和 disable 属性

我们可以给不希望用户修改的表单控件增加 `readonly` 属性, 对于不允许用户操作的表单控件增加 `disable` 属性. 比如我们可以把文本框, 密码框设置为 `readonly` 不允许用户更改, 可也 把某些按钮设置成 `disable` 不允许用户点击

## 14) `tabindex` 设置置表单顺序和快捷键

使用 `tabindex` 属性可以设定 `tab` 键的顺序, 使用该属性可以使表单控件按顺序响应, 使用 `accesskey` 属性可以为表单控件指定快捷键, 在 windows 使用 `alt+快捷键` 使特定的控件获得焦点

## 15) `action`

form 表单中的 `action` 属性的值是表单提交给的 url, 如果我们没有设置 `action` 属性, 那么默认的表单还是提交给当前页面,

- `action=""` 和 `action="#"`、没有 `action` 属性的作用相同, 都是提交到当前页面(也就是 `document.location.href`)
- `action="http://currentPage.xxx"` 如果 `http://currentPage.xxx` 表示当前页面, 则提交到当前页面, 同时跳转到当前页面, 当然就看不到 url 的变化; 如果 `http://currentPage.xxx` 是非当前页面, 则提交到非当前页面, 也跳转到非当前页面
- 表单提交到自身时 `action=""` 与 `action="http://currentPage.xxx"` 的区别:

通常为了省事, 我们一般将表单的 `action` 属性设为 `action=""` / `action="#"` 或者干脆不写, 但是这会造成一个问题, 就是其实将 `action` 设为 `action=""` 时, 表单提交时真正提交的页面是

`action="currentPage.xxx?arguments=values&..."`, 也就是如果当前页面带有参数, 参数也一并会提交, 而且参数的优先级会高于 `<form>` 控件提交的参数, 即如果 `action="currentPage.xxx?arguments=values&..."` 中的 `arguments` 有与 `<form>` 元素名相同的时候, 获取值时

`action=" currentPage.xxx?arguments=values&..."` 值会将 `<form>` 的值覆盖, 从页得到不正确的值.

## 15) `method` 属性

`method` 属性规定如何发送表单数据 (表单数据发送到 `action` 属性所规定的页面)。

最常用的两种方法是: GET 和 POST

- **get**

GET 方法, 这时浏览器会与表单处理服务器建立连接, 然后直接在一个传输步骤中发送所有的表单数据: 浏览器会将数据直接附在表单的 action URL 之后。这两者之间用问号进行分隔。

- **post**

POST 方法, 浏览器将会按照下面两步来发送数据。首先, 浏览器将与 action 属性中指定的表单处理服务器建立联系, 一旦建立连接之后, 浏览器就会按分段传输的方法将数据发送给服务器。在服务器端, 一旦 POST 样式的应用程序开始执行时, 就应该从一个标志位置读取参数, 而一旦读到参数, 在应用程序能够使用这些表单值以前, 必须对这些参数进行解码。用户特定的服务器会明确指定应用程序应该如何接受这些参数。

除此之外还可以是 cgi

- **cgi**

CGI 是 Web 服务器运行时, 调用外部应用程序 (即 CGI 程序, 如 nginx fast-cgi) 的规范, CGI 规范允许 Web 服务器执行外部程序, 并将它们的输出发送给 Web 浏览器, CGI 将 Web 的一组简单的静态超媒体文档变成一个完整的、新的交互式媒体, 按照 CGI 编写的程序可以扩展 Web 服务器功能。CGI 应用程序能够通过 Web 服务器与浏览器进行交互, 还可以通过数据库 API 与数据库服务器等外部数据源进行通信, 从数据库服务器中获取数据, 将数据格式化为 HTML 文档后, 发送给浏览器, 也可以将从浏览器获得的数据放到数据库中。

【如何使用 cgi, 后续会继续编写相应文章详细讨论】

一般浏览器通过上述任何一种方法都可以传输表单信息, 而有些服务器只接受其中一种方法提供的数据。可以在 `<form>` 标签的 method (方法) 属性中指明表单处理服务器要用方法来处理数据, 使 POST 还是 GET。

POST 还是 GET? 如果表单处理服务器既支持 POST 方法又支持 GET 方法, 那么该选择哪种方法呢? 下面是有关这方面的一些规律:

1. 如果希望获得最佳表单传输性能, 可以采用 GET 方法发送只有少数简短字段的小表单。
2. 一些服务器操作系统在处理可以立即传递给应用程序的命令行参数时, 会限制其数目和长度, 在这种情况下, 对那些有许多字段或是很长的文本域的表单来说, 就应该采用 POST 方法来发送。
3. 如果你在编写服务器端的表单处理应用程序方面经验不足, 应该选择 GET 方法。如果采用 POST 方法, 就要在读取和解码方法做些额外的工作, 也许这并不难, 但是也许你不太愿意去处理这些问题。

4. 如果安全性是个问题, 那么我们建议选用 POST 方法。GET 方法将表单参数直接放在应用程序的 URL 中, 这样网络窥探者可以很轻松地捕获它们, 还可以从服务器的日志文件中进行摘录。如果参数中包含了信用卡帐号这样的敏感信息, 就会在不知不觉中危及用户的安全。而 POST 应用程序就没有安全方面的漏洞, 在将参数作为单独的事务传输给服务器进行处理时, 至少还可以采用加密的方法。
5. 如果想在表单之外调用服务器端的应用程序, 而且包括向其传递参数的过程, 就要采用 GET 方法, 因为该方法允许把表单这样的参数包括进来作为 URL 的一部分。而另一方面, 使用 POST 样式的应用程序却希望在 URL 后还能有一个来自浏览器额外的传输过程, 其中传输的内容不能作为传统 <a> 标签的内容。

### 3. 综合实例

```
<html>
<head><title>input 标签的使用</title></head>
<body background="image/3.jpg">
<h4>用户注册</h4>
<form action="mailto:21689991@qq.com" method="post" name="usrform">
  <table width="50%">
    <tr>
      <td width="20%">用户名:</td>
      <td><input type="text" name="username" value="" size="30" maxlength="30"></td>
    </tr>
    <tr>
      <td width="20%">密码:</td>
      <td><input type="password" name="userpassword" size="30" maxlength="30"></td>
    </tr>
    <tr>
      <td width="20%">性别:</td>
      <td>
        <input type="radio" name="sex" value="0" checked>男
        <input type="radio" name="sex" value="1">女
      </td>
    </tr>
    <tr>
      <td width="20%">专业:</td>
      <td>
        <select name="major" size="10" multiple>
          <option value="1">嵌入式
          <option value="2" selected>JAVA
          <option value="3">大数据
          <option value="4">微电子
          <option value="5">自动化
        </select>
      </td>
    </tr>
  </table>
</form>
```

```
<option value="6">机械
</select>
</td>
</tr>
<tr>
    <td>
        <input type="checkbox" name="instresting" value="readBook"> 读书
        <input type="checkbox" name="instresting" value="web">上网
        <input type="checkbox" name="instresting" value="music"> 听音乐
    </td>
</tr>
<tr>
    <td width="20%">个人简介:</td>
    <td>
        <textarea name="yourword" rows="5" cols="40">性别:男 爱好:女</textarea>
    </td>
</tr>
<tr>
    <td width="20%">靓照:</td>
    <td><input type="file" name="photo" size="30"></td>
</tr>
<tr>
    <td colspan="2" height="48">
        <input type="submit" name="register" value="注册">
        <input type="button" name="checkuser" value="检测我的用户名">
        <input type="reset" value="重填">
        <button name="checkuser" ></button>
    </td>
</tr>
<tr>
    <td>
        <input type="image" src="image/backButton.png" width="80" height="40" name="goback" alt="返回上页">
    </td>
</tr>
</table>
```



```
<input type="hidden" name="option" value="register">
<input type="hidden" name="checkNumber" value="0121">
</form>
</body>
</html>
```



这段代码中

- action 属性指定处理表单的 url,
- method 指定发送表单使用的方法
- `<input type="text">`是文本输入框,`<input type="submit">` 是提交按钮,
- `<input type="reset">`是重置按钮.重置按钮会清除刚刚填入的内容,
- `<input type="submit">`注册按钮会提交填入的内容给指定的 url 去处理表单.