

分段: (两段之间有空行)

换行标签:(两行之间没有空行)

原样输出标签:

注释:

图片:

超链接

表单

浮动框架

```
1 <html>
2 <head>
3
4 <title>测试网页</title>
5
6 </head>
7 <body>
8 这是我的第一个html网页
9 </body>
10
11
12 </html>
13
```

另存为html格式;

本地磁盘 (C:) ▸ AppServ ▸ www ▸				
共享 ▾ 新建文件夹				
名称	修改日期	类型	大小	
appserv	2018/9/5 9:05	文件夹		
cgi-bin	2018/9/5 9:05	文件夹		
dwwa	2018/9/5 9:08	文件夹		
dwwa_all	2018/9/5 9:07	文件夹		
phpMyAdmin	2018/9/5 9:05	文件夹		
index.php	2008/5/8 18:52	PHP 文件	5 KB	
phpinfo.php	2005/6/13 10:19	PHP 文件	1 KB	
test1.html	2018/9/5 23:54	Chrome HTML Do...	1 KB	
新建文本文档.txt	2018/9/5 23:53	文本文档	1 KB	



右击查看源代码:



网页结构标签

```
1 <!DOCTYPE HTML>
2 <html lang="zh_CN">
3 <head>
4   <meta charset="utf-8">
5   <meta name="description" content="This is a HTML demo page">
6   <meta name="Keywords" content="HTML,demo">
7   <title> HTML demo </title>
8 </head>
9
10 <body>
11   HTML demo
12 </body>
13 </html>
```

- HTML主要用于标记网页结构及内容类型，便于搜索引擎识别。
- 网页的格式设置主要由CSS来完成。
- 网页的程序逻辑主要由PHP来完成

下面讲body里面的:

分段标签<p>

<p>这是段落。</p>

<p>这是另一个段落。</p>

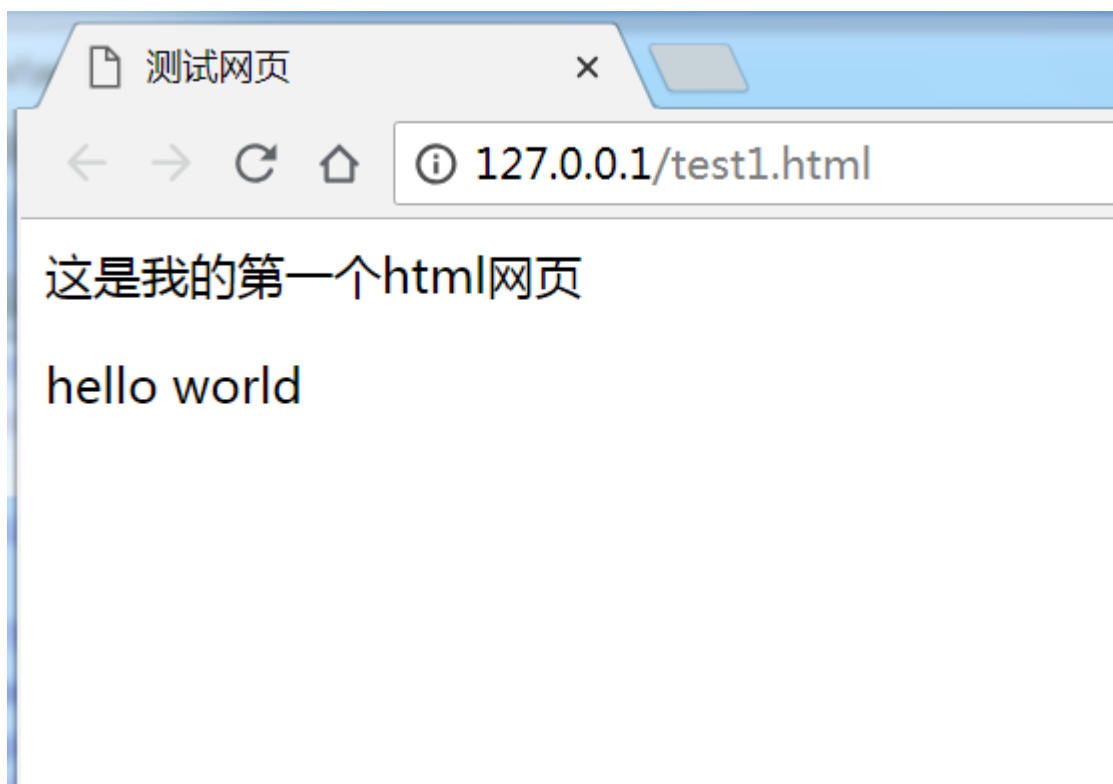
```
1 <html>
2 <head>
3   <title> 测试网页 </title>
4 </head>
5
6 <body>
7   这是我的第一个HTML网页
8   [
9
10  这是一个段落！
11 </body>
12 </html>
```

← → ↻ 127.0.0.1/test1.html

这是我的第一个HTML网页 这是一个段落!

分段: (两段之间有空行)

```
1  <html>
2  <head>
3
4    <title>测试网页</title>
5
6  </head>
7  <body>
8    <p>这是我的第一个html网页</p>
9    <p> hello world</p>
10 </body>
11
12
13 </html>
```



换行标签:(两行之间没有空行)

换行标签

<p>

To break
lines
in a
paragraph,
use the br tag.

</p>

To break
lines
in a
paragraph,
use the br tag.

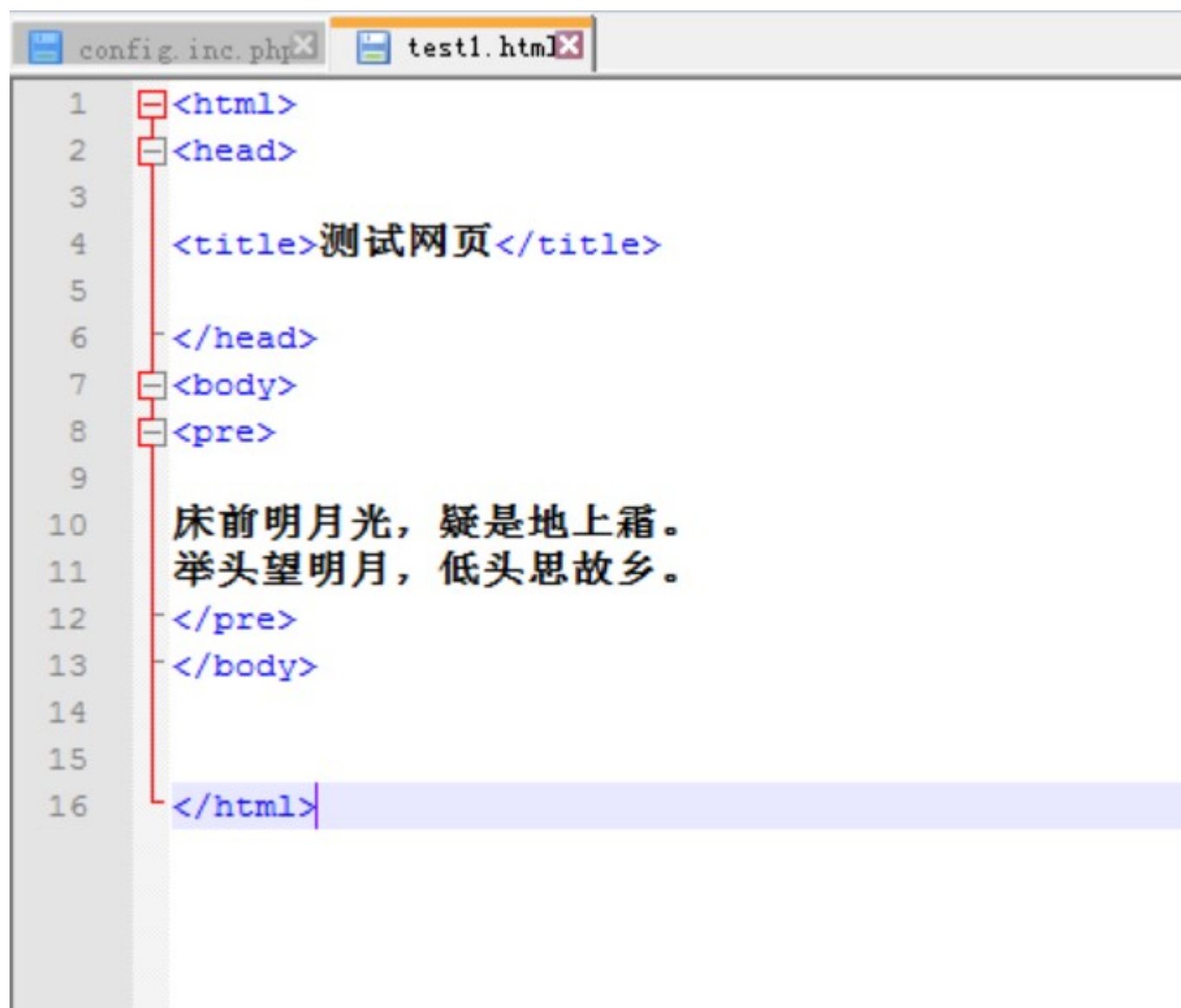
```
config.inc.php test1.html
1 <html>
2 <head>
3
4 <title>测试网页</title>
5
6 </head>
7 <body>
8 <p>这是我的第一个html网页</p>
9 <p> hello<br /> world <br /> you</p>
10 </body>
11
12
13 </html>
```



原样输出标签:

原样输出标签<pre>

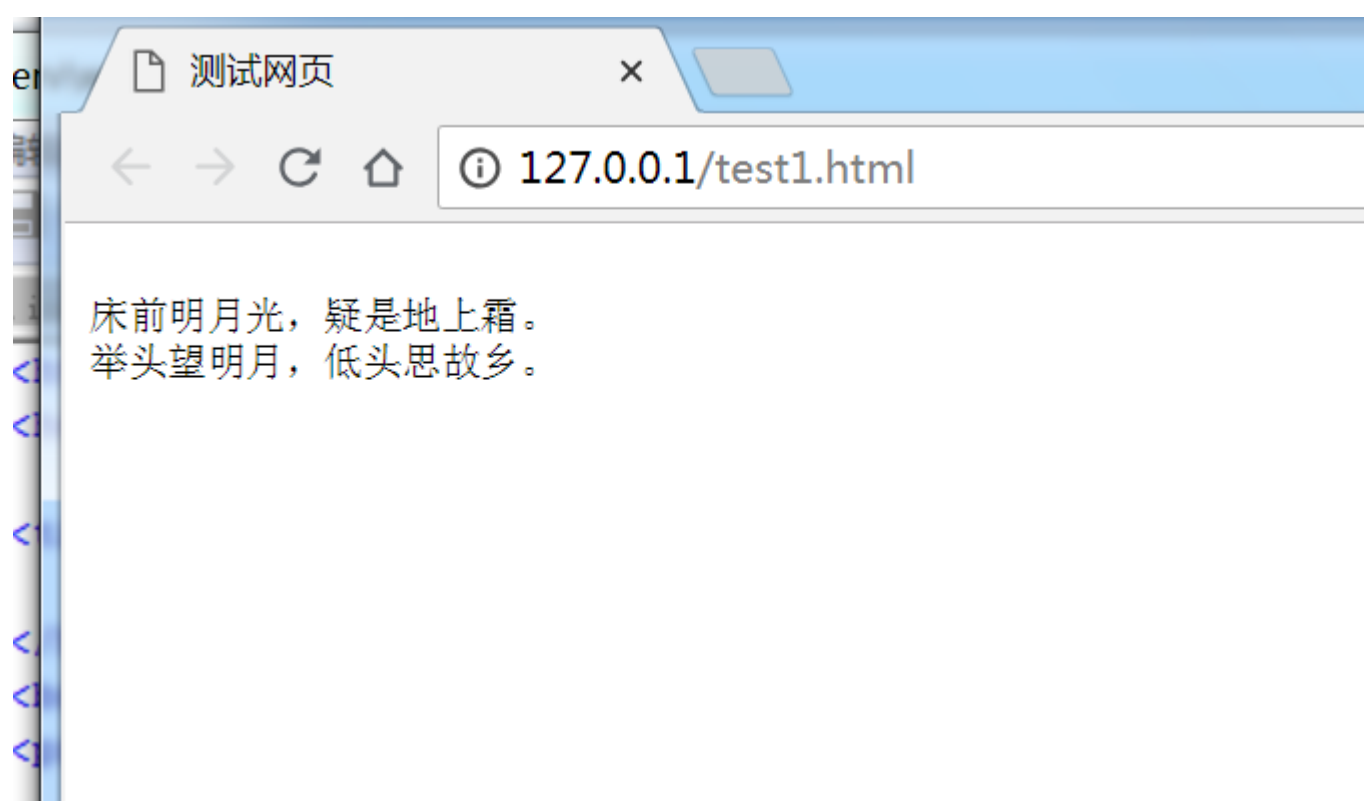
```
<pre>
床前明月光，
疑是地上霜。
举头望明月，
低头思故乡。
</pre>
```



The screenshot shows a web browser window with two tabs: 'config.inc.php' and 'test1.html'. The 'test1.html' tab is active, displaying a web page with the following HTML code:

```
1 <html>
2 <head>
3
4 <title>测试网页</title>
5
6 </head>
7 <body>
8 <pre>
9
10 床前明月光，疑是地上霜。
11 举头望明月，低头思故乡。
12 </pre>
13 </body>
14
15
16 </html>
```

The poem is displayed in a monospaced font, preserving the original line breaks and spacing. The text is: 床前明月光，疑是地上霜。 举头望明月，低头思故乡。



注释:

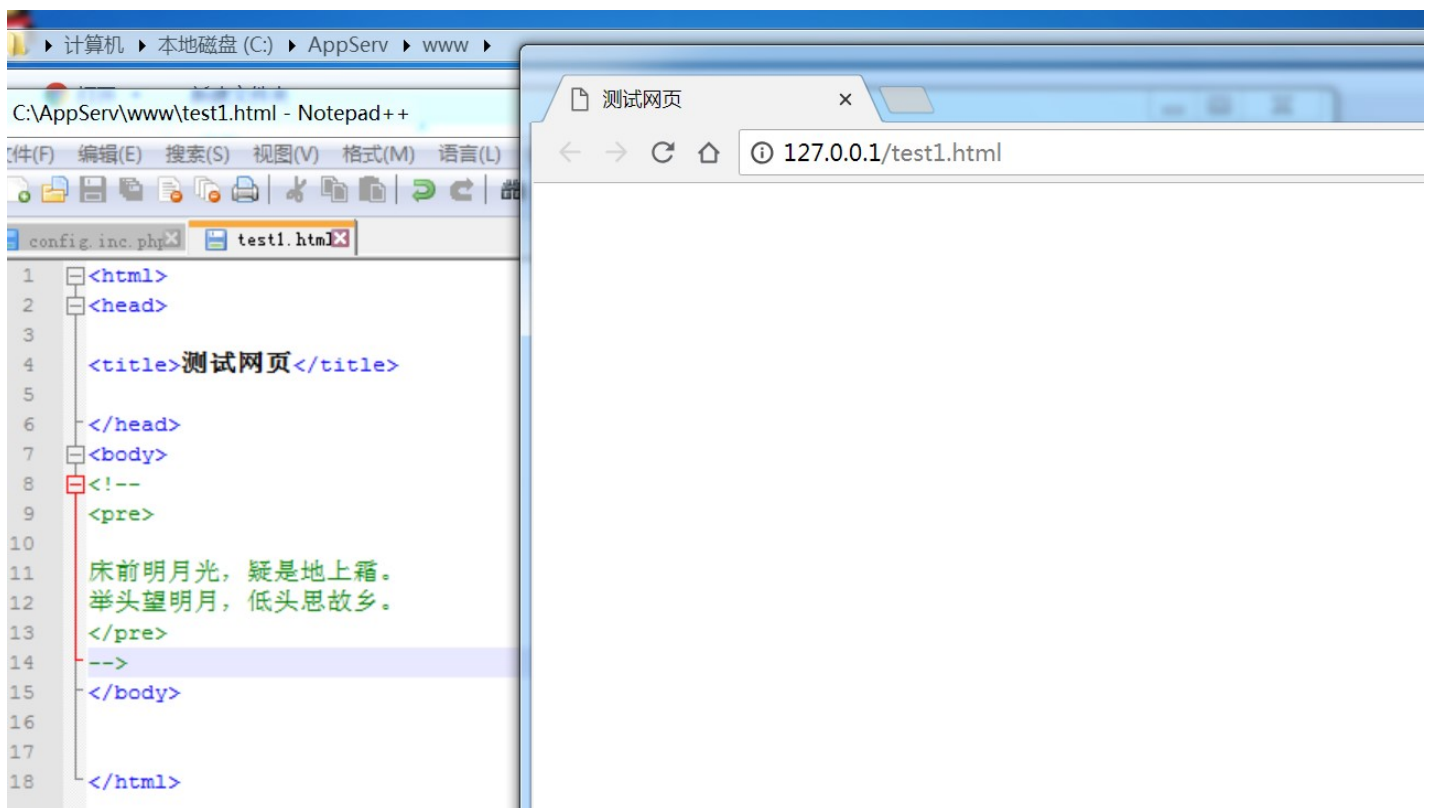
注释标签<!-->

语法格式：

```
<!--  
.....  
-->
```

如果不加结束标签-->，默认将开始标签至文档末尾的所有内容全部注释。


```
1 <html>
2 <head>
3   <title> 测试网页 </title>
4 </head>
5
6 <body>
7   <!--
8   <pre>
9     床前明月光，
10    疑是地上霜。
11    举头望明月，
12    低头思故乡。
13  </pre>
14  -->
15 </body>
16 </html>
```



图片：

图片标签

语法格式： ``

- `` 标签不需要闭合
- alt 描述信息主要是给搜索引擎用

`<p></p>`

`<p></p>`

计算机 > 本地磁盘 (C:) > AppServ > www >

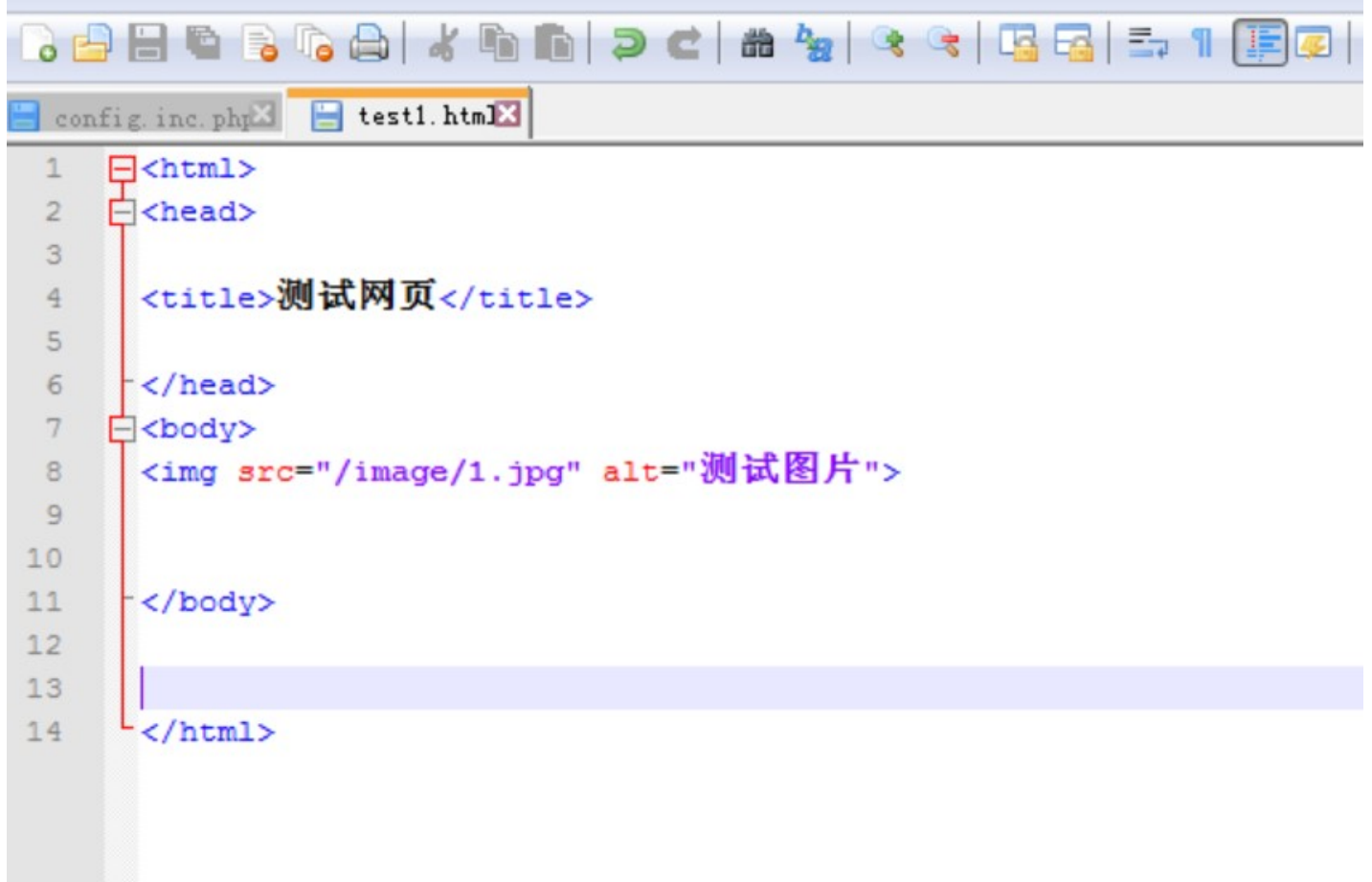
到库中 > 共享 > 新建文件夹

名称	修改日期	类型	大小
appserv	2018/9/5 9:05	文件夹	
cgi-bin	2018/9/5 9:05	文件夹	
dwva	2018/9/5 9:08	文件夹	
dwva_all	2018/9/5 9:07	文件夹	
phpMyAdmin	2018/9/5 9:05	文件夹	
1.jpg	2018/9/6 0:31	JPEG 图像	222 KB
index.php	2008/5/8 18:52	PHP 文件	5 KB
phpinfo.php	2005/6/13 10:19	PHP 文件	1 KB
test1.html	2018/9/6 0:29	Chrome HTML Do...	1 KB
新建文本文档.txt	2018/9/5 23:53	文本文档	1 KB

本地磁盘 (C:) > AppServ > www > image

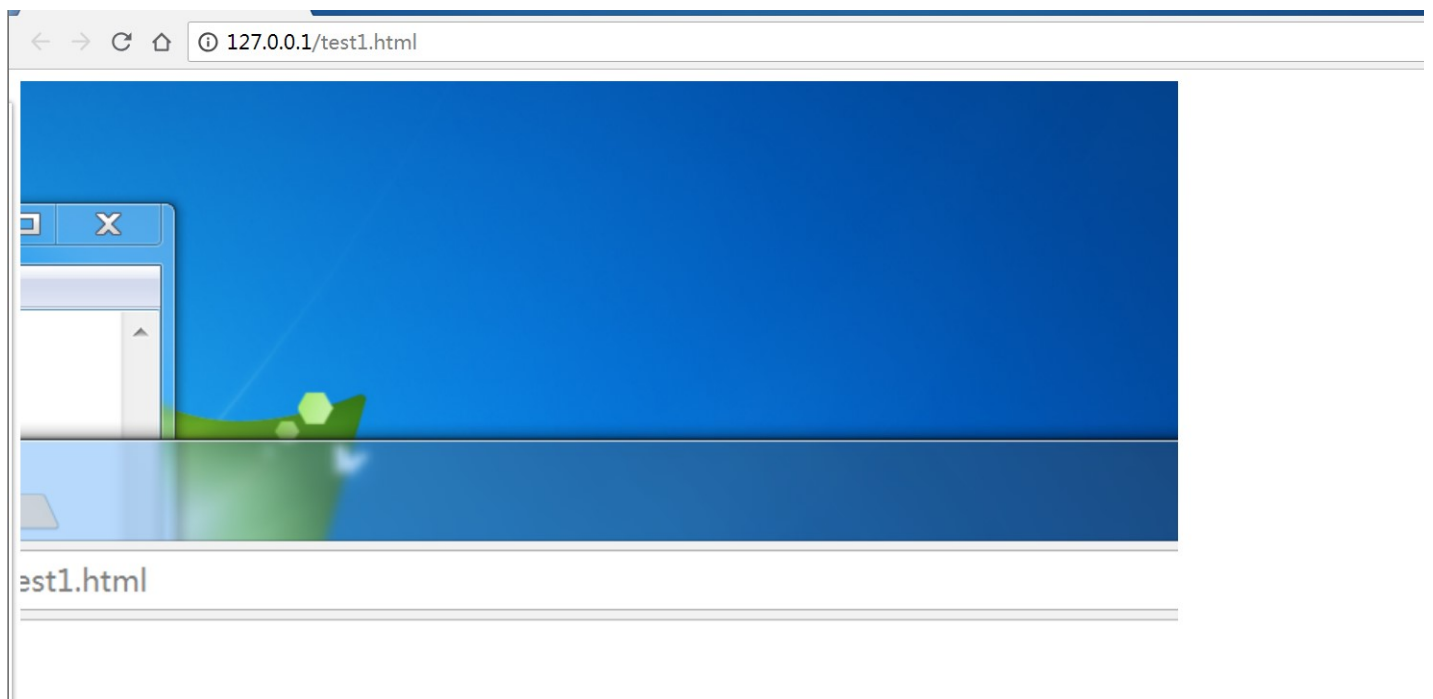
打印 新建文件夹

名称	修改日期	类型	大小
1.jpg	2018/9/6 0:31	JPEG 图像	



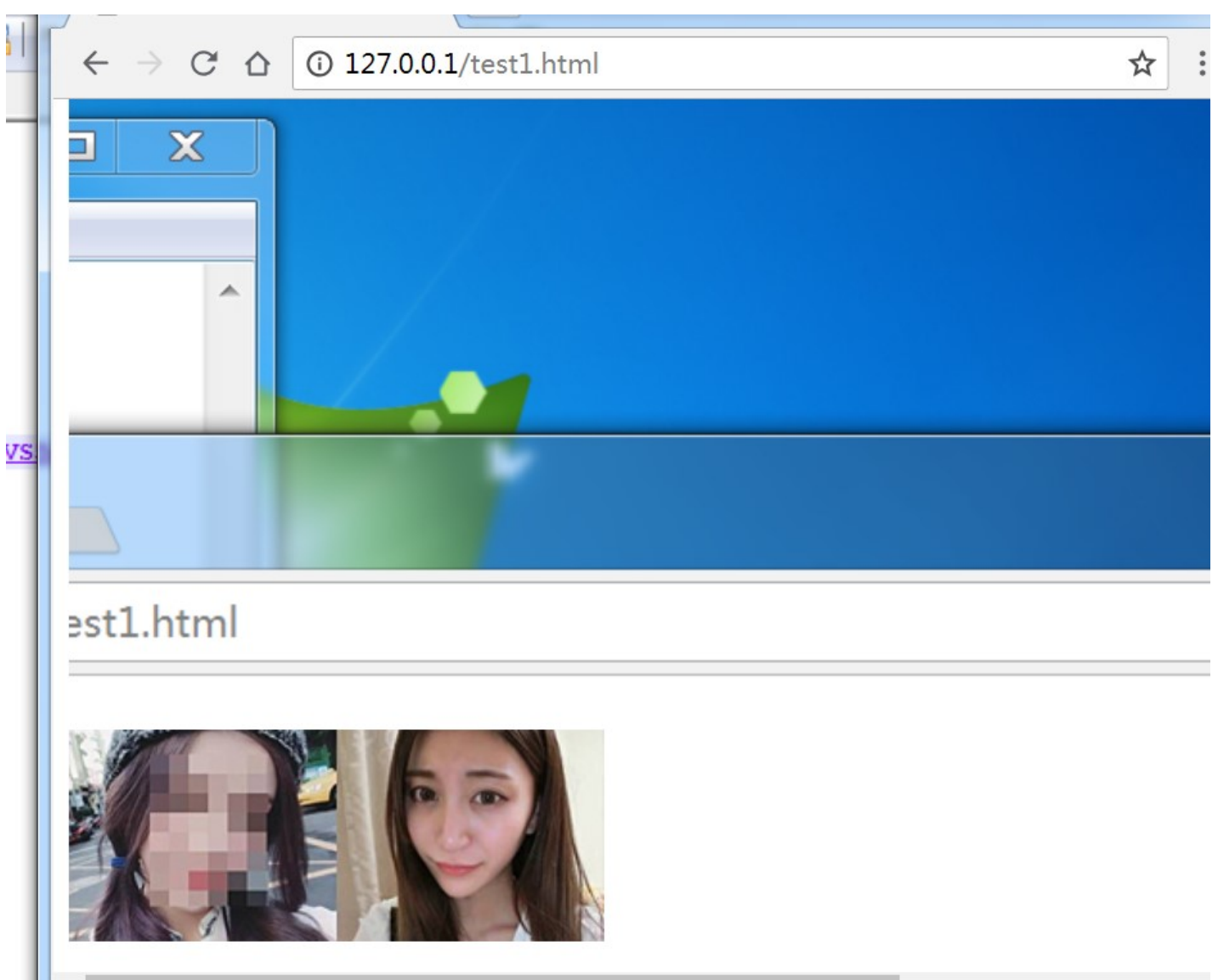
```
1 
```

这里面表示了一个相对路径 `/image/1.jpg` 表示根目录(这里指的是www这个文件夹)下面的image文件夹下面的1.jpg这个图片



```
1 <html>
2 <head>
3   <title> 测试网页 </title>
4 </head>
5
6 <body>
7   
8   
9 </body>
10 </html>
```

```
config.inc.php test1.html
1 <html>
2 <head>
3
4   <title>测试网页</title>
5
6 </head>
7 <body>
8   
9
10  <p></p>
11 </body>
12
13
14 </html>
```



超链接

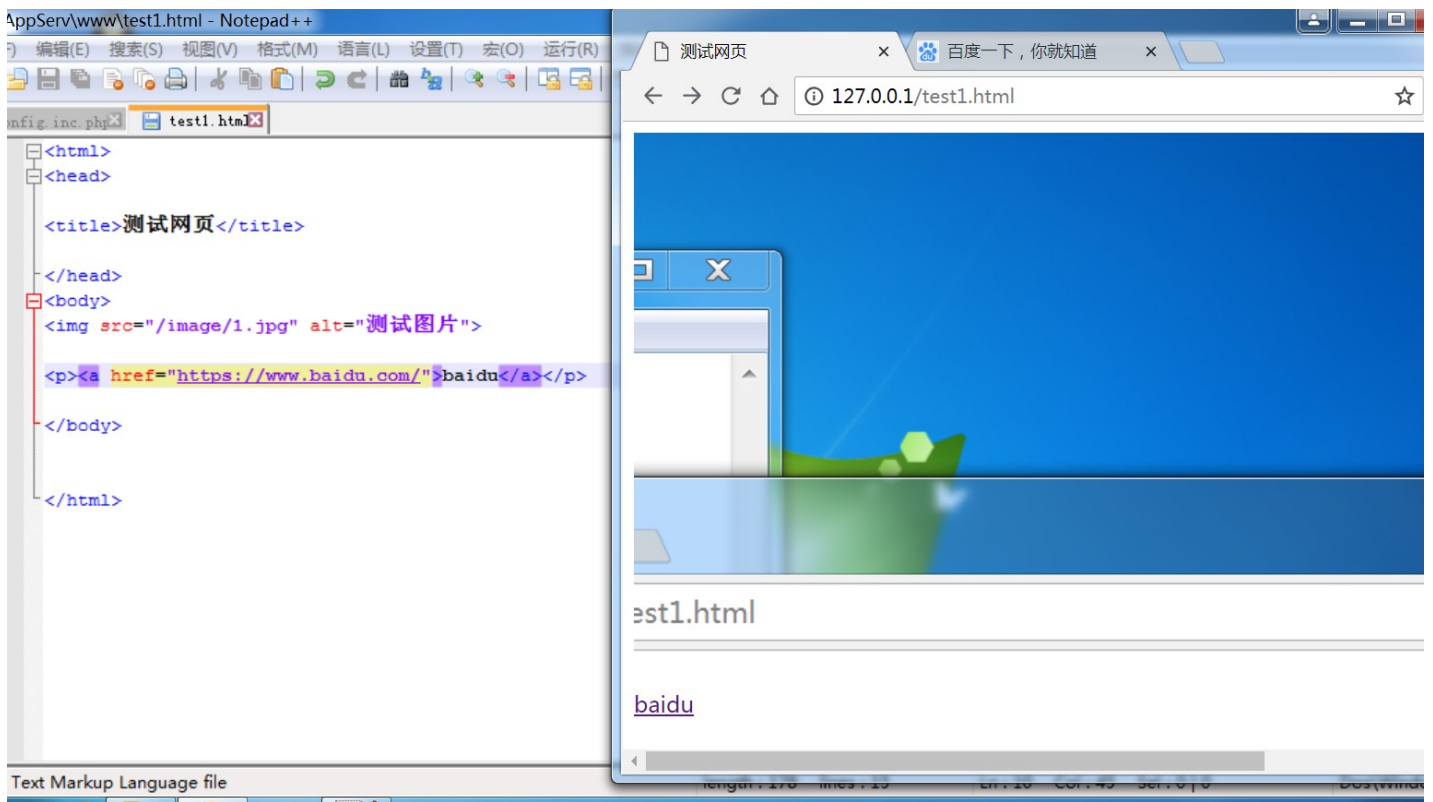
文本超链接

语法格式： `文字描述`

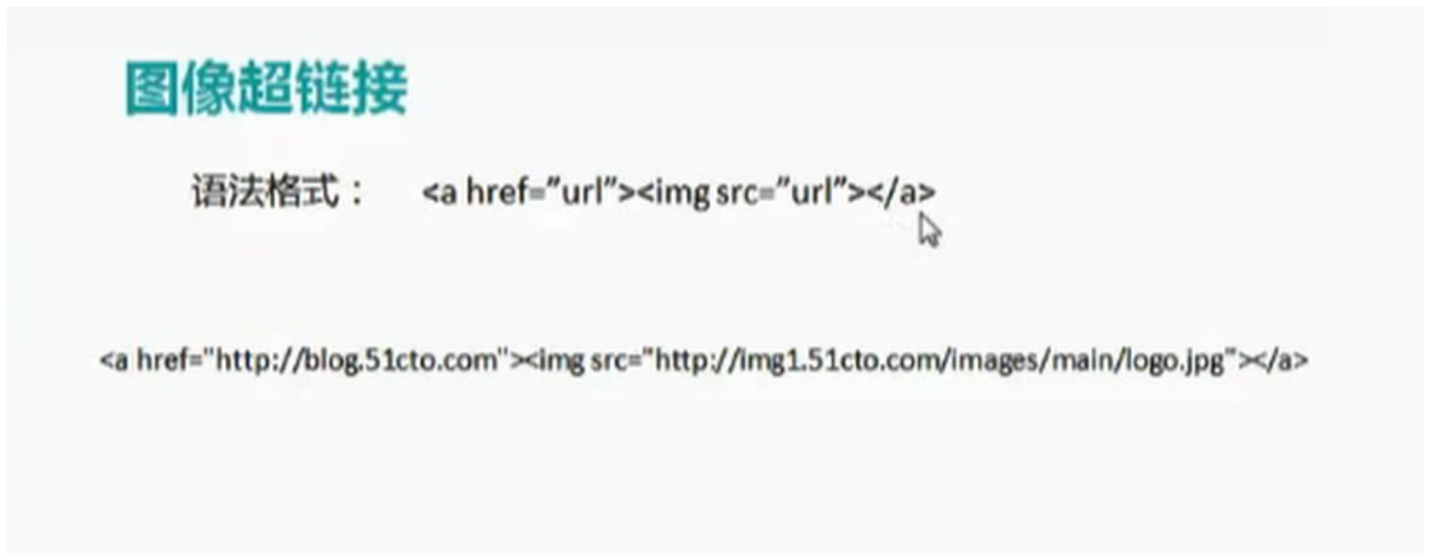
`跳转到51cto博客`

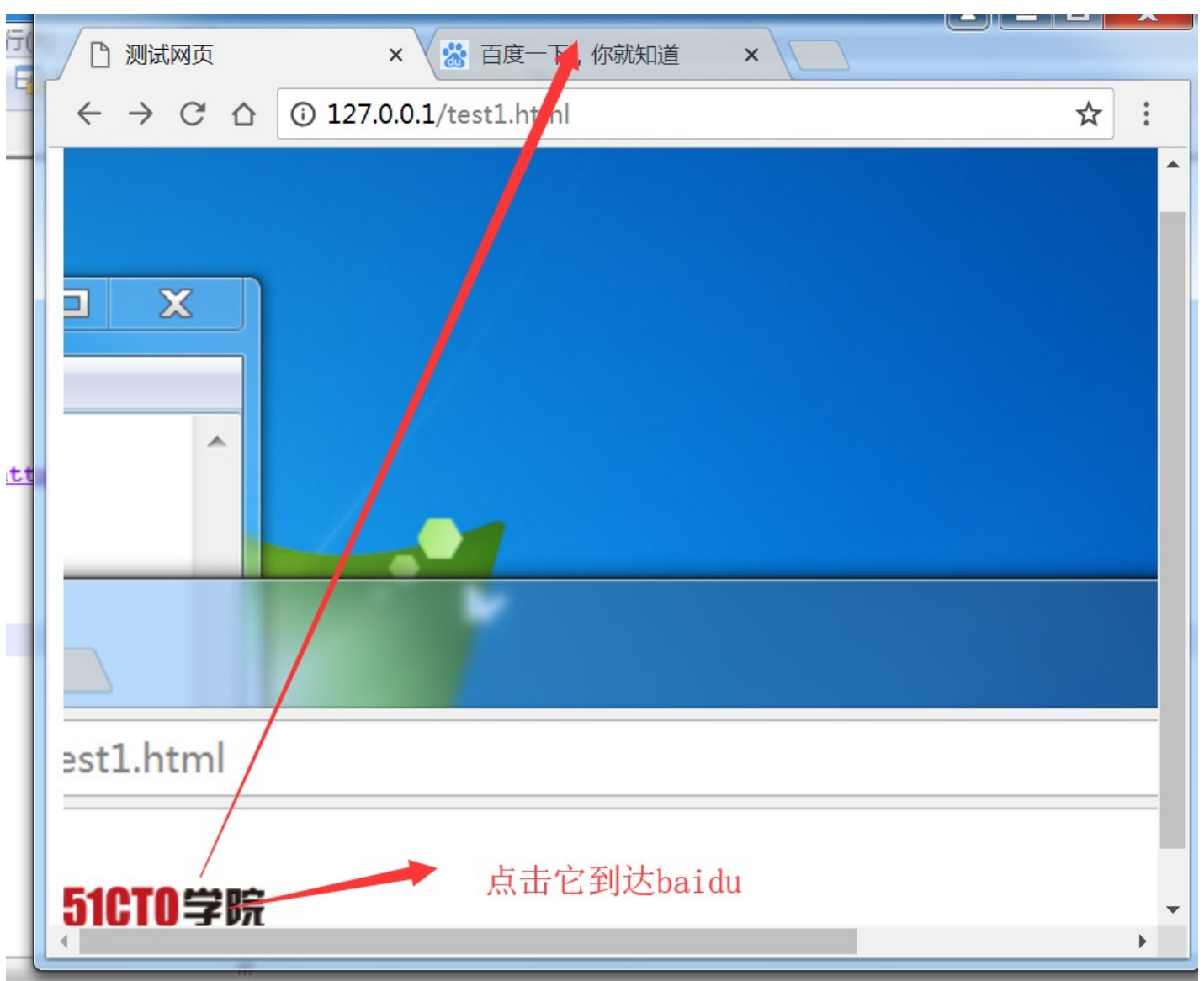
href属性指定url；

以上面图里面的例子为例:将文字描述这4个字嵌入网页里面,一点这四个字,就跳转至url所指定的页面;



图像超链接:





表单

场景:登录界面

表单

- 表单主要用于接收用户输入的信息，并向服务器提交。
- 表单使用表单标签`<form>.....</form>`定义。
- 表单中的子标签可用来实现某些具体功能，用到最多的是输入标签`<input>`

基本语法格式：

```
<form>
  <input type="type_name" name="field_name">
</form>
```

```
<form action="UserLogin.php" method="post">
  <p>用户名: <input type="text" name="username"></p>
  <p>密 码: <input type="password" name="password">
  <p><input type="submit" name="submit" value="确定"
</form>
```

属性

表单用于接收用户输入的数据,再将其发送给服务器,表单包括文本框和按钮;表单有一些子标签 (最常用的是input标签),每个子标签 又能实现一些功能;

action属性后面跟一个页面,“UserLogin.php”, 这个页面的作用是接收通过表单传给服务器的数据, (表单作用是接收数据传给服务器),数据传给服务器以后谁来处理这些数据,所以这里定义了一个动态的 php页面,由这个页面来验证用户名和密码是否正确(一般需要用户名和密码的,用户名和密码在服务端那边判断是否正确);

method是方法,你要通过什么方法把表单的数据传给服务器,客户端传数据给服务端的方法一般有两种, get和post(传输数据量大用post, 一般表单用post),一般传给服务器的数据在url显示出来的是get(会有一个?, 比如? username=...),而post传出的数据是在地址栏里面看不到的(5")

URL的详细格式

schema://host[:port#]/path/.../[?query-string][#anchor]

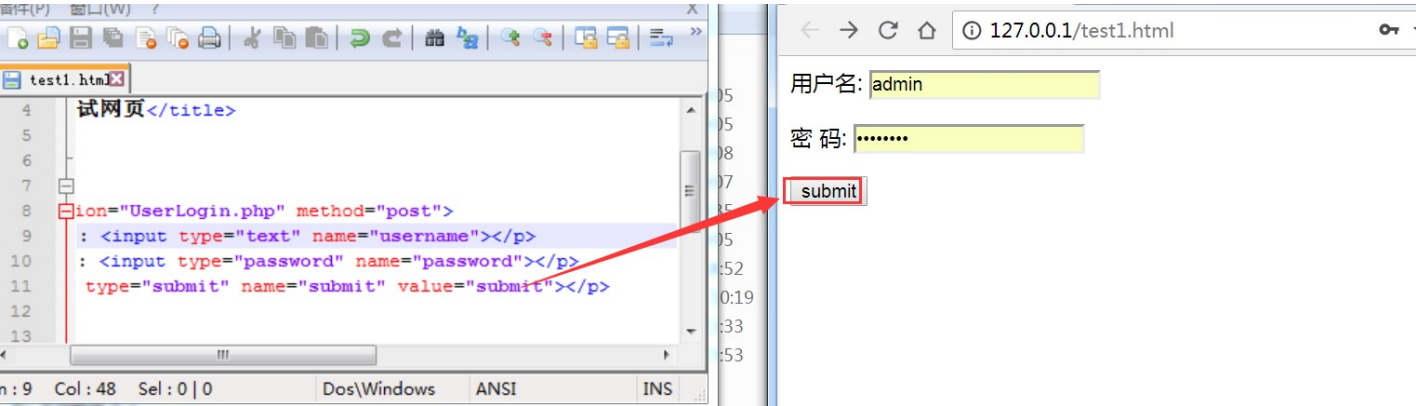
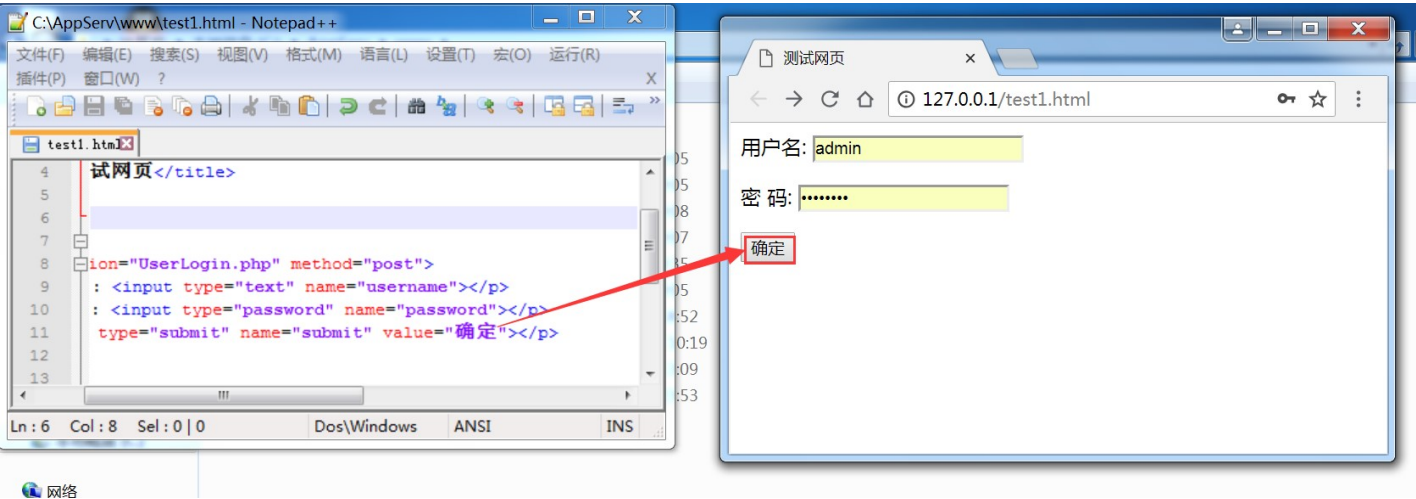


(表单主要是一些文本框和按钮)

input标签可以用来实现文本框,也可以实现按钮,是什么类型的要看后面的type, text类型的是一个文本框, username是给用户名冒号后面的文本框定义的一个名字; why? 我们输入的信息都需要传给UserLogin.php这个网页页面去处理(在这个例子里面); UserLogin.php这个网在一上来就要接受通过表单传给它的的数据, 接受的时候怎么知道那个数据是用户名, 密码, 那么从name是username文本框传过来的数据就是用户名, 从名字是password的文本框传过来的数据就是密码; 所以必须给每个文本框起个名字, 这样接收方在服务器那儿接受的时候就能区分!

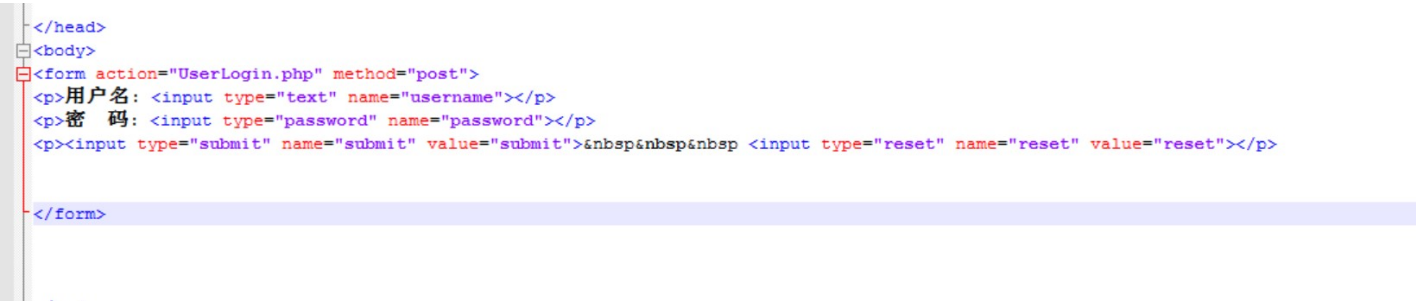
第三行是一个提交的按钮, 一点这个按钮, 数据就传出去了, 这里提交按钮的名字叫submit, 名字可以随便起, 但是类型的确定的(这里有text, password, submit三种类型)!

submit显示出来的文字就是value的值:(value后面是什么,后面按钮上面就显示什么)



 代表一个空格:

下图表示两个按钮之间有三个空格;



重置按钮功能是把名字和密码清0,服务器那里不接受他的数据,所以这里的名字有没有无所谓的!但是value必须有!

浮动框架

浮动框架iframe

可以在浏览器页面中嵌套其他子窗口，从而显示其它网页的内容。

```
<iframe src="http://www.51cto.com" height=500 width=1000 ></iframe>
```

```
<iframe src="http://www.51cto.com" height=0 width=0 ></iframe>
```

```
<iframe src="http://www.51cto.com" height=0 width=0 frameborder=0></iframe>
```

The image is a screenshot of a web browser window. The address bar shows '127.0.0.1/test1.html'. The page has a login form with '用户名: admin' and '密码:', and 'submit' and 'reset' buttons. Below the form, there's a sidebar with a QR code and contact information, and a main content area with a list of articles about D-Link router vulnerabilities.