

知识点1【B/S的概述】（了解）

- 1、b/s和c/s的区别

知识点2【boa的移植】（了解）

- 1、修改寻找boa.conf的路径
 - 2、编译源码
 - 3、配置boa.conf
 - 4、运行服务器boa
 - 5、测试服务器
-

知识点3【html的概述】（了解）

- 1、文字标签
 - 2、标题标签
 - 3、超链接标签
 - 4、表格标签
 - 5、表单标签
-

知识点4【JavaScript的概述】（了解）

知识点5【Document对象】（了解）

- 1、提供了从JS脚本中对 HTML 页面中的所有元素进行访问
-

登录页面

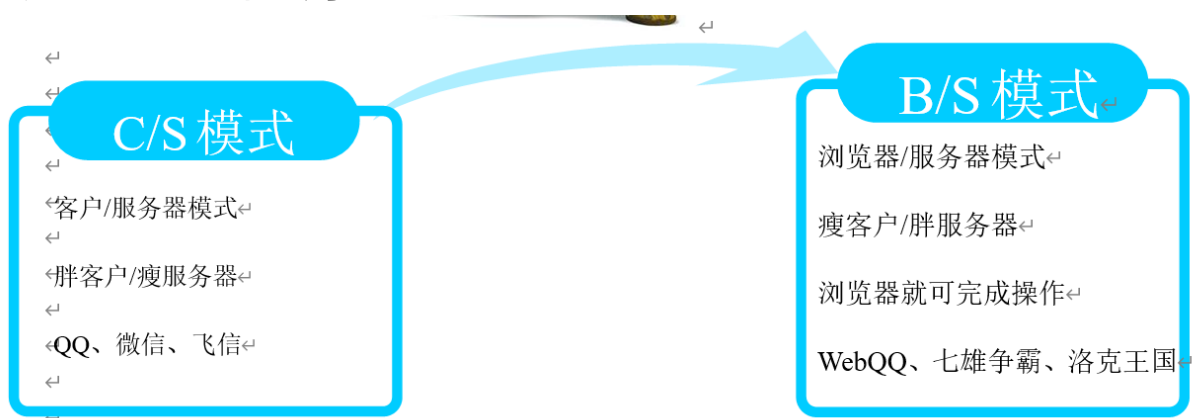
06_login.js

知识点6【windows对象】（了解）

更改label以及img的内容

知识点1【B/S的概述】（了解）

1、b/s和c/s的区别



| 角度 | C/S | B/S |
|------|---------------------------|-------------------------------------|
| 硬件环境 | 专用网络 | 广域网 |
| 安全要求 | 面向相对固定的用户群 信息安全的控制能力很强 | 面向是不可知的用户群 对安全的控制能力相对弱 |
| 程序架构 | 更加注重流程 系统运行速度可较少考虑 | 对安全以及访问速度要多重的考虑 B/S结构的程序架构是发展的趋势 |
| 软件重用 | 差 | 好 |
| 系统维护 | 升级难 | 开销小、方便升级 |
| 处理问题 | 集中 | 分散 |
| 用户接口 | 与操作系统关系密切 | 跨平台，与浏览器相关 |
| 信息流 | 交互性低 | 交互密集 |

知识点2 【boa的移植】（了解）

从工具中，将boa-0.94.13-src.tar.gz拷贝到虚拟机的任意目录

解压上述压缩包

```
1 tar xvf boa-0.94.13-src.tar.gz
```

进入boa-0.94.13-src

```
1 cd boa-0.94.13-src
```

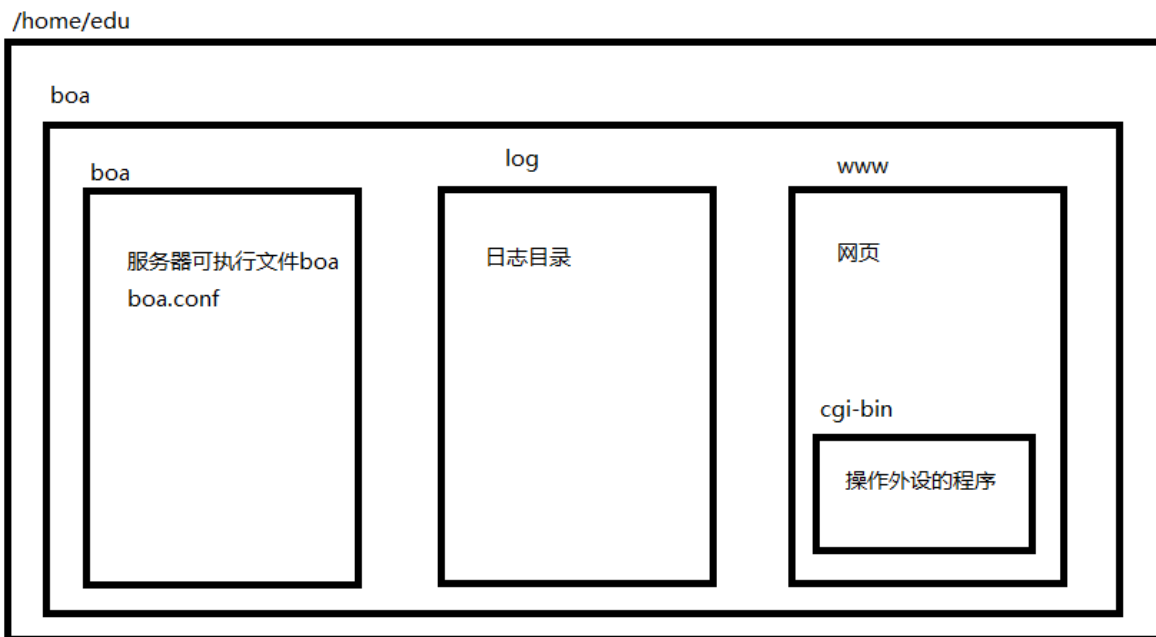
```
edu@edu:~/tools/boa-0.94.13-src$ ls
boa.conf ChangeLog contrib CREDITS docs examples extras Gnu_License README src
edu@edu:~/tools/boa-0.94.13-src$
```

配置文件

源码

```
1 cd src
```

boa服务器的安装目录分析:



```
1 mkdir /home/edu/boa/boa -p
2 mkdir /home/edu/boa/log
3 mkdir /home/edu/boa/www
4 mkdir /home/edu/boa/www/cgi-bin
```

1、修改寻找boa.conf的路径

```
edu@edu:~/tools/boa-0.94.13-src/src$ ls
acconfig.h  buffer.c  config.h  escape.c  log.c  request.c  webindex.pl
aclocal.m4  cgi.c     config.h.in  escape.h  Makefile.in  response.c
alias.c     cgi_header.c  config.log  get.c     mmap_cache.c  select.c
boa.c       check_struct_for.m4  config.status  globals.h  parse.h  signals.c
boa_grammar.y  compat.h  configure  hash.c     pipe.c  sublog.c
boa.h       config.c  configure.in  index_dir.c  queue.c  timestamp.c
boa_lexer.l  config.cache  defines.h  ip.c       read.c  util.c
edu@edu:~/tools/boa-0.94.13-src/src$
```

需要修改调用boa.conf的位置

```
1 vim defines.h
```

```
28
29 #ifndef SERVER_ROOT
30 #define SERVER_ROOT "/home/edu/boa/boa"
31 //#define SERVER_ROOT "/mnt/Smart_home"
32 #endif
33
```

自己确定的目录

保存退出: wq!

2、编译源码

```
1 ./configure
2 make
```

错误1: bison: 命令未找到

解决: sudo apt-get install bison

错误2: flex: 命令未找到

解决: sudo apt-get install flex

```
1 make 编译完成
```

```
edu@edu:~/tools/boa-0.94.13-src/src$ cp boa /home/edu/boa/boa
```

```
edu@edu:~/tools/boa-0.94.13-src/src$ tree ~/boa
```

```
/home/edu/boa
├── boa
│   └── boa
├── log
├── www
│   └── cgi-bin
```

4 directories, 1 file

```
edu@edu:~/tools/boa-0.94.13-src/src$
```

```
edu@edu:~/tools/boa-0.94.13-src/src$ cd ..
```

```
edu@edu:~/tools/boa-0.94.13-src$ ls
```

```
boa.conf  ChangeLog  contrib  CREDITS  docs  examples  extras  Gnu_License  README  src
```

```
edu@edu:~/tools/boa-0.94.13-src$ cp boa.conf ~/boa/boa
```

```
edu@edu:~/tools/boa-0.94.13-src$ tree ~/boa
```

```
/home/edu/boa
├── boa
│   ├── boa
│   └── boa.conf
├── log
├── www
│   └── cgi-bin
```

4 directories, 2 files

```
edu@edu:~/tools/boa-0.94.13-src$
```

3、配置boa.conf

```
1 cd ~/boa/boa
```

47

48 User 0

49 Group 0

50

61

62 ErrorLog /home/edu/boa/log/error_log

63 # Please NOTE: Sending the logs to a pipe ('|'), as shown below,

64 # is somewhat experimental and might fail under heavy load.

65 # "Usual libc implementations of printf will stall the whole

66 # process if the receiving end of a pipe stops reading."

67 #ErrorLog "|/usr/sbin/cronolog --symlink=/var/log/boa/error_log /var

68

69 # AccessLog: The location of the access log file. If this does not

70 # start with /, it is considered relative to the server root.

71 # Comment out or set to /dev/null (less effective) to disable

72 # Access logging.

73

74 AccessLog /home/edu/boa/log/access_log

75 # Please NOTE: Sending the logs to a pipe ('|'), as shown below,

76 # is somewhat experimental and might fail under heavy load.

77 # "Usual libc implementations of printf will stall the whole

78

109 # Comment out to disable server root user files.

110

111 DocumentRoot /home/edu/boa/www

112

113 # UserDir: The name of the directory which is appended

114 # directory if a ~user request is recieved.

115

116 UserDir public_html

122

123 DirectoryIndex index.html

124

125 # DirectoryMaker: Name of program used to c

126 # Comment out to disable directory listings

127 # DirectoryIndex are commented out. accessi

```
153 # Do *not* comment out (better use AddType!)
154
155 MimeType /home/edu/boa/boa/mime.types ←
156
157 # DefaultType: MIME type used if the file extension is unknc
158 # is no file extension.
159
160 DefaultType text/plain
161
```

```
192
193 ScriptAlias /cgi-bin/ /home/edu/boa/www/cgi-bin/
194
-- 插入 --
```

```
1 cd ..
```

```
edu@edu:~/boa$ sudo cp /etc/mime.types boa
[sudo] password for edu:
edu@edu:~/boa$ tree
```

```
.
├── boa
│   ├── boa
│   ├── boa.conf
│   └── mime.types ←
├── log
├── www
└── cgi-bin
```

```
4 directories, 3 files
edu@edu:~/boa$
```

```
edu@edu:~/boa$ touch log/error_log
edu@edu:~/boa$ touch log/access_log
edu@edu:~/boa$ tree
```

```
.
├── boa
│   ├── boa
│   ├── boa.conf
│   └── mime.types
├── log
│   ├── access_log
│   └── error_log
├── www
└── cgi-bin
```

```
4 directories, 5 files
edu@edu:~/boa$
```

4、运行服务器boa

```
edu@edu:~/boa$ ls
boa  log  www
edu@edu:~/boa$ sudo ./boa/boa
edu@edu:~/boa$ ps -A | grep boa
11794 pts/4    00:00:00 boa
edu@edu:~/boa$
```

成功

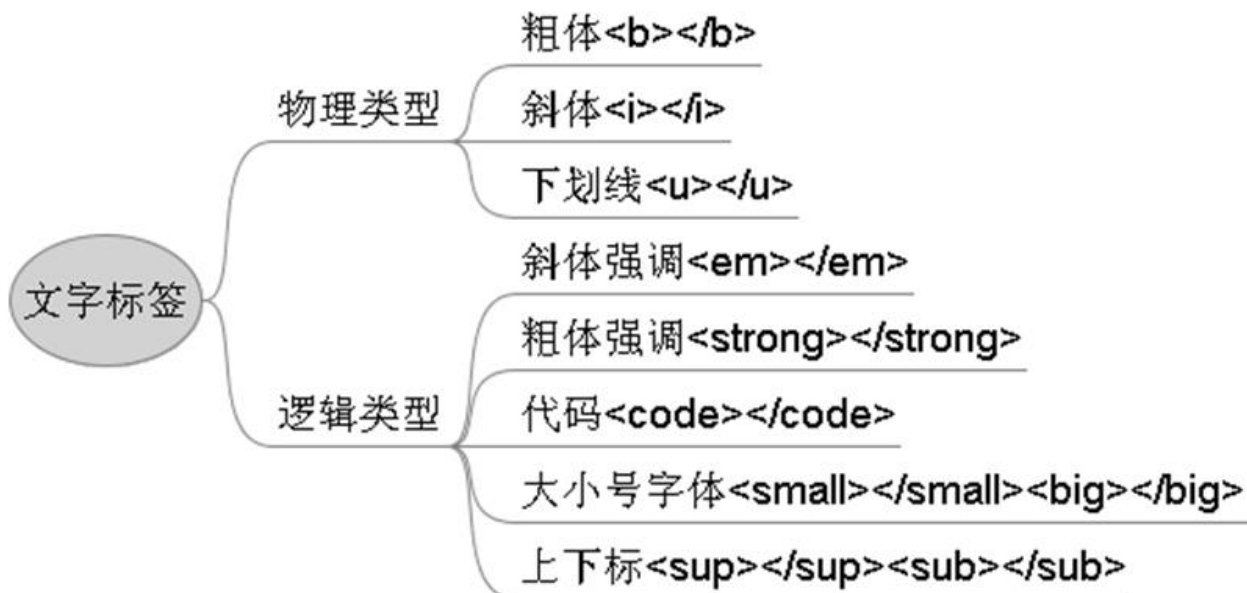
5、测试服务器

知识点3【html的概述】（了解）

HTML是超文本标记语言(Hyper Text Markup Language)

HTML由各种各样的标签(tag)组成，如<html> </html>、<body> </body>

1、文字标签



```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <title>Document</title>
7  </head>
8
9  <body>
10     <b>加粗的内容</b>
11     <br>
12     <i>斜体的内容</i>
13     <br>
14     <u>下划线</u>
15     <br>
16     <em>斜体强调</em>
17     <br>
18     <strong>粗体强调</strong>
19     <br>
20     <code>
21         printf("hello world\n");
22     </code>
23
24     <!-- 下标 H2O -->
25     <br>
26     水: H<sub>2</sub>O
27     <br>
```



```
28     平方: X<sup>2</sup>
29 </body>
30
31 </html>
```

加粗的内容

斜体的内容

下划线

斜体强调

粗体强调

```
printf("hello world\n");
```

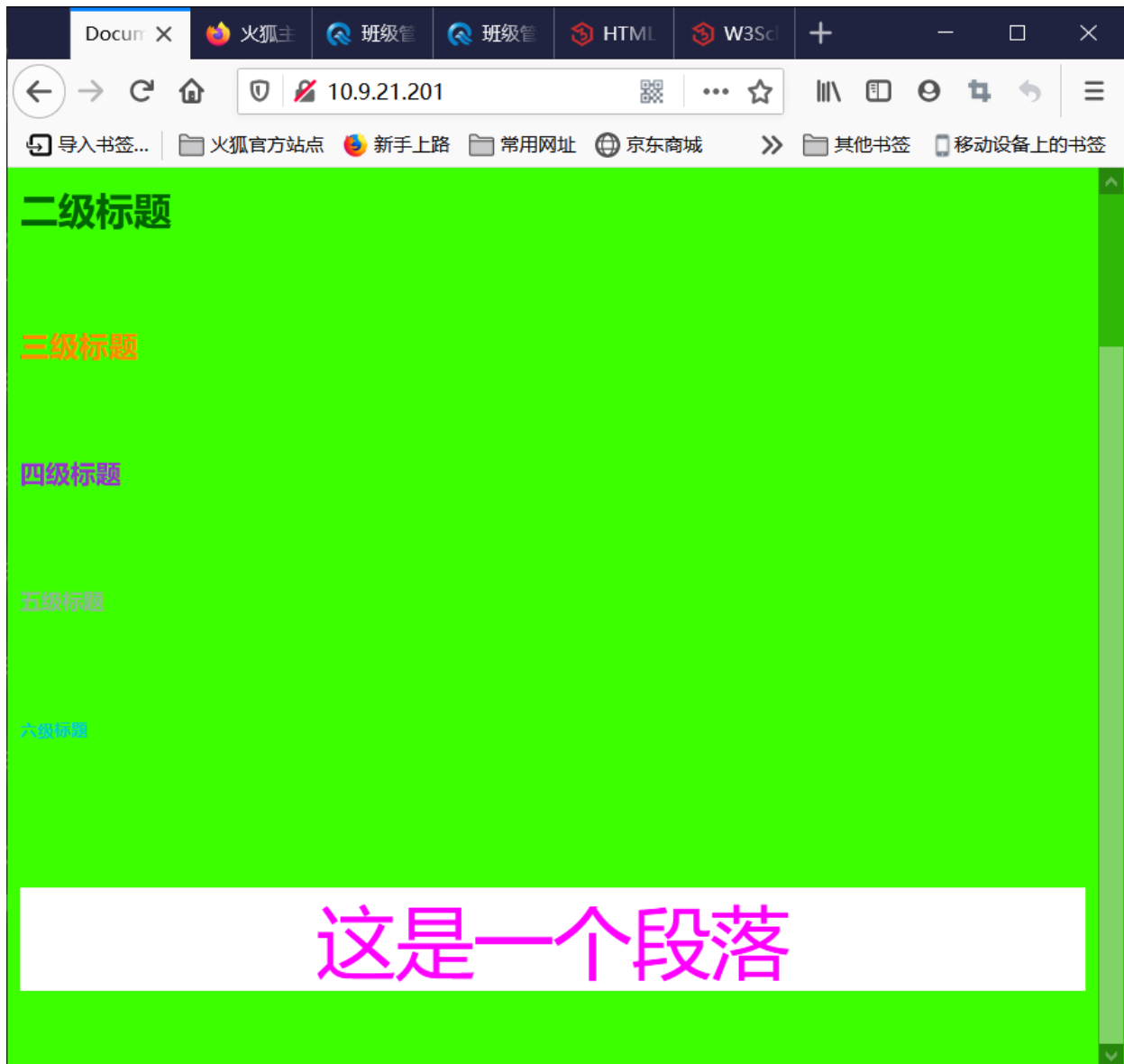
水: H₂O

平方: X²

2、标题标签

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <title>Document</title>
7 </head>
8
9 <body style="background-color: rgb(60, 255, 0);">
10     <h1 style="color: crimson;">一级标题</h1>
11     <br>
12     <h2 style="color: darkgreen;">二级标题</h2>
13     <br>
14     <h3 style="color: darkorange;">三级标题</h3>
15     <br>
16     <h4 style="color: darkorchid;">四级标题</h4>
17     <br>
18     <h5 style="color: darkseagreen;">五级标题</h5>
19     <br>
20     <h6 style="color: darkturquoise;">六级标题</h6>
21     <br>
```

```
22     <p style="background-color: white; font-  
family: Arial;color: rgb(255, 0, 255);  
23     font-size: 50px;text-align: center;">这是一个段落</p>  
24 </body>  
25  
26 </html>
```



3、超链接标签

```
1 <a></a>
```

1) 电子邮件超链接

` `

注: subject ---主题

(2) 页面内的超链接

回到顶部: ` `

回到某一位置: ` `

(3) 页面外的超链接

外网:

内网:

(4) 图片超链接

4、表格标签

```
1 <table></table>
```

表格标记

| 标 签 | 描 述 |
|----------------------------|---|
| <table>...</table> | 用于定义一个表格开始和结束 |
| <caption> ...</caption> | 定义表格的标题。在表格中也可以不用此标签。 |
| <th>...</th> | 定义表头单元格。表格中的文字将以粗体显示,在表格中也可以不用此标签,<th>标签必须放在<tr>标签内 |
| <tr>...</tr> | 定义一行标签,一组行标签内可以建立多组由<td>或<th>标签所定义的单元格 |
| <td>...</td> | 定义单元格标签,一组<td>标签将建立一个单元格,<td>标签必须放在<tr>标签内 |

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6
7     <title>Document</title>
8 </head>
```

```

9
10 <body>
11     <table border="1" style="width: 500px; text-align: center;" align="center">
12         <!-- 表格的标题 -->
13         <caption>学生信息</caption>
14         <!-- 行标签 -->
15         <tr>
16             <th colspan="3">学生信息</th>
17             <th colspan="2">成绩</th>
18             <th rowspan="3">照片</th>
19         </tr>
20         <tr>
21             <td>姓名</td>
22             <td>性别</td>
23             <td>专业</td>
24             <td>课程</td>
25             <td>分数</td>
26         </tr>
27         <tr>
28             <td>lucy</td>
29             <td>女</td>
30             <td>物联网</td>
31             <td>BS开发</td>
32             <td>99</td>
33         </tr>
34
35
36     </table>
37 </body>
38
39 </html>

```

学生信息

| 学生信息 | | | 成绩 | | 照片 |
|------|----|-----|------|----|----|
| 姓名 | 性别 | 专业 | 课程 | 分数 | |
| lucy | 女 | 物联网 | BS开发 | 99 | |

5、表单标签

(1) 属性

name: 表单的名称

method: 表单数据从浏览器传输到服务器的方法

get: 将表单数据附加在URL地址后面, 长度不超过8192个字符, 不具有保密性,
默认为get

post: 将表单数据包含在表单的主体中, 一起传输到服务器上。没有长度限制, 密
文传输

action: 用来定义表单处理程序

(2) <form></form>内的标签

a、<input> 表单输入标签

常用的文本域、按钮都是使用这个标签

属性:

name 域名称

type 域类型

value 元素值

type属性值:

text 文字域 **password** 密码域

file 文件域 **checkbox** 复选框

radio 单选框 **button** 普通按钮

submit 提交按钮 **reset** 重置按钮

hidden 隐藏域 **image** 图像域

b、选择列表 <select><option></option></select>

菜单和列表是为了节省网页的空间而产生的

属性

name 菜单和列表的名称

size 显示的选项数目

multiple 列表中的选项为多项

selected 默认被选中的选项(option中的属性)

c、文本域 <textarea></textarea>

用来制作多行文本输入域

属性

name 文字域的名称

rows 文字域的行数

cols 文字域的列数

知识点4 【JavaScript的概述】（了解）

连接文件:

```
1 <script type="text/javascript" src="./06_login.js"></script>
```

知识点5 【Document对象】（了解）

- 1、提供了从JS脚本中对 **HTML** 页面中的所有元素进行访问
- 2、可以通过**getElementById()**方法,来根据对应的ID号去访问、控制**HTML**页面中的标签元素
- 3、可以通过**title**，**URL**属性获取当前文档的标题，**URL**信息等
- 4、可以通过**write**方法在**HTML**页面中写入**HTML**表达式

常用方法

如果获取输入框使用value

```
1 document.getElementById("user").value="";
```

如果获取label使用innerHTML

```
1 document.getElementById("mylabel").innerHTML
```

如果获取img使用src

```
1 document.getElementById("img").src = "./image/butterfly.png";
```

登录页面

index.html

[illegible]

```
15     <input type="button" value="登录" onclick="my_fun(1)">  
16     &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
17     <input type="button" value="取消" onclick="my_fun(0)">  
18 </div>  
19 </body>  
20 </html>
```

06_login.js

```
1 // 函数的形参不需要var
2 function my_fun(arg) {
3     if (arg == 1) {
4         var user = document.getElementById("user").value;
5         var pwd = document.getElementById("pwd").value;
6         if (user == "admin" && pwd == "123456") {
7             alert("登录成功");
8             // window.open("./05__from.html");
9             window.location.href = "04_table.html";
10        }
11        else {
12            alert("用户名或密码错误");
13            document.getElementById("user").value = "";
14            document.getElementById("pwd").value = "";
15        }
16    }
17 }
18 else if (arg == 0) {
19     alert("我是取消");
20     document.getElementById("user").value = "";
21     document.getElementById("pwd").value = "";
22 }
23 }
```

知识点6 【windows对象】 (了解)

Window 对象表示浏览器中打开的窗口，打开一个 HTML 网页会创建一个 window 对象

Window 对象是全局对象←

window.open()打开一个新的窗口←

window.close()关闭当前窗口←

window.location.href: 返回完整的 URL;对其进行赋值, 则能够跳转到相应的网页

更改label以及img的内容

index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <script type="text/javascript" src="07_js.js"></script>
7     <title>Document</title>
8 </head>
9
10 <body>
11     <label id="mylabel">我会72变</label>
12     <input type="button" value="72变" onclick="my_change()">
13     <br>
14     
15 </body>
16
17 </html>
```

07_js.js

```
1 function my_change() {
2     var text = document.getElementById("mylabel").innerHTML;
3     alert(text);
4     document.getElementById("mylabel").innerHTML = "哈哈，我变了";
5 }
6 function change_image() {
7     document.getElementById("img").src = "./image/butterfly.png";
8 }
```