知识点总结

自定义web服务器

webserver.py

*"""  
用Python实现一个HTTP web服务器，它知道如何运行服务器端CGI脚本：  
从当前工作目录提供文件和脚本；Python脚本必须存储在 webdir\cgi-bin或  
webdir\htbin中；  
"""***import** os, sys  
**from** http.server **import** HTTPServer, CGIHTTPRequestHandler  
  
webdir = **'.' *# 存放HTML文件和cgi-bin脚本文件夹的所在***port = 80 ***# 缺省http://localhost/，也可以使用http://localhost:xxxx/***os.chdir(webdir) ***#进入运行目录***srvaddr = (**''**,port) ***#定义服务器地址和端口***srvobj = HTTPServer(srvaddr, CGIHTTPRequestHandler) ***#创建HTTPServer对象***srvobj.serve\_forever() ***# 以守护进程永久运行***

启动脚本后，可访问此目录下的文件

新建在此目录下新建html文件 cgi101.html

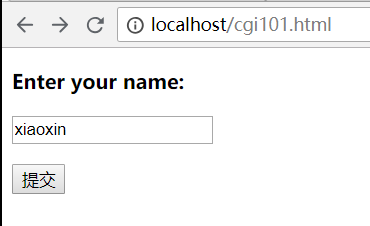
<!DOCTYPE **html**>  
<**html lang="en"**>  
<**head**>  
 <**meta charset="UTF-8"**>  
 <**title**>Interactive Page</**title**>  
</**head**>  
<**body**>  
<**form method="POST" action="cgi-bin/cgi101.py"**>  
 <**p**><**B**>Enter your name:</**B**></**p**>  
 <**p**><**input type="text" name="user"**></**p**>  
 <**p**><**input type="submit"**></**p**>  
</**form**>  
</**body**>  
</**html**>

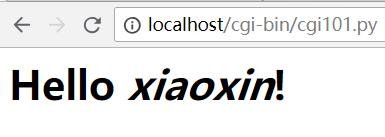
创建目录cgi-bin

创建文件cgi-bin/cgi101.py

**import** cgi  
form = cgi.FieldStorage() ***# 解析表单数据***print(**'Content-type: text/html\n'**) ***# 输出类型***print(**'<title>Reply Page</title>'**) ***# 输出title*if not 'user' in** form: ***# 若用户未输入内容*** print(**'<h1>Who are you?</h1>'**) ***#输出页面内容*else**:  
 print(**'<h1>Hello <i>%s</i>!</h1>'** % cgi.escape(form[**'user'**].value))

访问 localhost/cgi101.html





第二章：系统编程

python系统模块

glob

用于文件扩展名

socket

用于网络连接和进程间通信

threading, \_thread, queue

用于运行和同步化并发线程

time, timeit

用于获取系统时间相关细节

subprocess, multiprocessing

用于启动和控制并行进程

signal, select, shutil, tempfile

用于多种系统相关任务

import string

string.ascii\_lowercase 所有小写字母

string.whitespace 空白分隔符

eval() 将字符串转为对象

repr() 将对象转为字符串

PP4E\System\more.py

*"""  
分割字符串或文本文件并交互地进行分页  
"""***def** more(text, numlies=15):  
 lines = text.splitlines()  
 **while** lines:  
 chunk = lines[:numlies]  
 lines = lines[numlies:]  
 **for** line **in** chunk:print(line)  
 **if** input(**'More?'**) **not in** [**'y'**, **'Y'**]:**break  
  
if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 **import** sys  
 more(open(sys.argv[1]).read(),10)

sys.platform 系统名称

sys.maxsize 最大整型

sys.version 系统版本信息

常用OS模块工具

Shell变量 os.environ

运行程序 os.system, os.popen, os.execv, os.spawnv

派生进程 os.fork, os.pipe, os.waitpid, os.kill

文件描述符，文件锁 os.open, os.read, os.write

文件处理 os.remove, os.rename, os.mkfifo, os.mkdir, os.rmdir

管理工具 os.getcwd, os.chdir, os.chmod, os.getpid, os.listdir, os.access

移植工具 os.sep, os.pathset, os.curdir, os.path.split, os.path.join

路径名工具 os.path.exists(‘path’), os.path.isdir(‘path’), os.path.getsize(‘path’)

os.pathsep, os.sep, os.pardir, os.curdir, os.linesep

(‘;’, ‘\\’, ‘..’, ‘.’, ‘\r\n’)

os.path.isdir(‘dir’) #判断是否为目录

os.path.isfile(‘file’) #判断是否为文件

os.path.exists(‘file’) #判断文件或目录是否存在

os.path.getsize(‘file’) #获取文件大小

os.path.split(‘/opt/opt.txt’) #分割路径和文件

os.path.join(‘/opt’,’opt.txt’) #合并文件路径

os.path.dirname() #文件目录路径

os.path.basename() #文件名称

os.path.splitext() #后缀名与路径分开

os.path.normpath() #处理混合路径

os.system #在python中运行shell命令

os.popen #运行shell命令并与其输入流相连接

subprocess模块

subprocess.call(‘ls’) #类似os.system()