第二十章：web编程

目录

[20.1CGI 介绍 1](#_Toc350624135)

[20．2 CGI 应用程序 1](#_Toc350624136)

[1、cgi 模块 1](#_Toc350624137)

[2、建立cgi应用程序 2](#_Toc350624138)

[3、在cgi中使用unicode编码 2](#_Toc350624139)

[4、高级cgi 2](#_Toc350624140)

[5、用 Python 建立 Web 服务器 2](#_Toc350624141)

## 20.1CGI 介绍

CGI:帮助 Web 服务器处理客户端数据

类似于java中的jsp

这整个过程开始于 Web 服务器从客户端接到了请求（GET 或者 POST），并调用合适的程序。然后开始等待 HTML 页面—与此同时，客户端也在等待。一旦程序完成，会将生成的动态 HTML 页面返回到服务器端，然后服务器端再将这个最终结果返回给用户。服务器接到表单反馈，与外部应用程序交互，收到并返回新生成的 HTML 页面都发生在一个叫做 Web 服务器 CGI（Common Gateway Interface）的接口上.图 20-3 描述了 CGI 的工作原理，逐步展示了一个用户从提交表单到返回最终结果 Web 页面的整个执行过程和数据流。

客户端输入给 Web 服务器端的表单可能包括处理过程和一些存储在后台数据库中的表单。需要记住的是，在任何时候都可能有任何一个用户去填写这个字段，或者点击提交按钮或者图片，这更像激活了某种 CGI 活动。

创建 HTML 的 CGI 应用程序通常是用高级编程语言来实现的，可以接受、处理数据，向服务器端返回 HTML 页面。目前使用的编程语言有 Perl， PHP， C/C++,或者 Python。

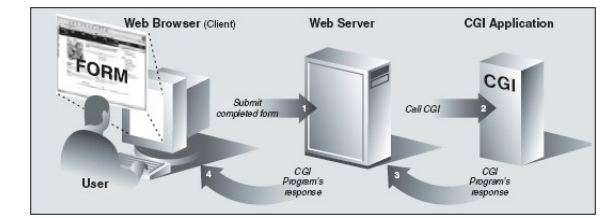


图 20-3 CGI 工作概要图。CGI 代表了在一个 Web 服务器和能够处理用户表单、生成并返回动态HTML 页的应用程序间的交互。

更进一步来说，有很多 Web 应用程序开发框架和内容管理系统，这些都弥补了过去 CGI 的不足。然而，在这些浓缩和升华下，它们仍旧遵循这 CGI 最初提供的模式，可以允许用户输入，根据输入执行拷贝，并提供了一个有效的 HTML 做为最终的客户端输出。因此，为了开发更加高效的 Web 服务有必要理解 CGI 实现的基本原理。

## 20．2 CGI 应用程序

### 1、cgi 模块

在 cgi 模块中有个主要类：FieldStorage 类，它完成了所有的工作。在 Python CGI 脚本开始时这个类将会被实例化，它会从 Web 客户端（具有 Web 服务器）读出有关的用户信息。一旦这个对象被实例化，它将会包含一个类似字典的对象，具有一系列的键-值对，键就是通过表单传入的表单条目的名字，而值则包含相应的数据。

这些值本身可以是以下三种对象之一。它们既可以是 FieldStorage 对象（实例）也可以是另一个类似的名为 MiniFieldStorage 类的实例，后者用在没有文件上传或 mulitple-part 格式数据的情况。MiniFieldStorage 实例只包含名字和数据的键-值对。最后，它们还可以是这些对象的列表。这发生在表单中的某个域有多个输入值的情况下。

### 2、建立cgi应用程序

首先是建立一个服务器。本书中代码简单，所以使用的是python自带的服务器。

如果你只是想建立一个基于 Web 的服务器，首先，你要切换到你想作为服务器目录的文件夹下，然后你可以直接执行下边的 Python 语句。

$ python -m CGIHTTPServer 8080

这将会在当前机器的当前目录下建立一个端口号为 8000 的 Web 服务器。然后可以在启动这个服务器的目录下建立一个 cgi–bin，将 Python CGI 脚本放到那里。将一些 HTML 文件放到那个目录下，然后就可以在地址栏中输入这些地址来访问 Web 站点啦。

例如：<http://localhost:8000/friends.htm> <http://localhost:8000/cgi-bin/friends2.py>

具体的实例参考代码。

### 3、在cgi中使用unicode编码

具体的参考代码

### 4、高级cgi

Mulitipart 表单提交和文件的上传

多值字段

Cookie

### 5、用 Python 建立 Web 服务器

由于已经打算建立这样的一个应用程序，你很自然的就需要创建个人素材，但是你将要用到的所有的基础代码都在 Python 的标准库中。要建立一个 Web 服务，一个基本的服务器和一个“处理器”是必备的。

基础的(Web)服务器是一个必备的模具。它的角色是在客户端和服务器端完成必要 HTTP 交互。在 BaseHTTPServer 模块中你可以找到一个名叫 HTTPServer 的服务器基本类。处理器是一些处理主要“Web 服务”的简单软件。它们处理客户端的请求，并返回适当的文件，

静态的文本或者由 CGI 生成的动态文件。处理器的复杂性决定了你的 Web 服务器的复杂程度。Python标准库提供了三种不同的处理器。

模块 描述

BaseHTTPServer 提供基本的 Web 服务和处理器类，分别是HTTPServer 和

BaseHTTPRequestHandler

SimpleHTTPServer 包含执行 GET 和 HEAD 请求的 SimpleHTTPRequestHandler 类

CGIHTTPServer 包含处理 POST 请求和执行 CGICGIHTTPRequestHandler 类

写例子中的收获：

1、写得cgi，用处理器最好使用cgihttpserver。

2、print 'Content-Type:text/plain\n\n'

注意加两个\n\n

3、其实不用加这个东西，#!usr/bin/env python

验证后，仍是正确的，不过呢，最好加上，指示了执行这个cgi脚本文件的程序，因为这里的是python代码，所以呢，指明了python的可执行程序的目录。

4\

## 启动python自带服务器

Python -m SimpleHTTPServer 8080

* BaseHTTPServer: 提供基本的Web服务和处理器类，分别是HTTPServer和BaseHTTPRequestHandler。
* SimpleHTTPServer: 包含执行GET和HEAD请求的SimpleHTTPRequestHandler类。
* CGIHTTPServer: 包含处理POST请求和执行CGIHTTPRequestHandler类。