Chapter 6 – Introduction to the Common Gateway Interface (CGI)

Outline

6.1 Introduction
6.2 Client and Web Server Interaction
  6.2.1 System Architecture
  6.2.2 Accessing Web Servers
  6.2.3 HTTP Transactions
6.3 Simple CGI Script
6.4 Sending Input to a CGI Script
6.5 Using XHTML Forms to Send Input and Using Module `cgi` to Retrieve Form Data
6.6 Using `cgi.FieldStorage` to Read Input
6.7 Other HTTP Headers
6.8 Example: Interactive Portal
6.9 Internet and World Wide Web Resources

6.1 Introduction

- Common Gateway Interface (CGI)
- Web-page generation
  - Dynamic : content generated each request
  - Static : never changes unless document edited

6.2 Client and Web Server Interaction

- Extensible Hypertext Markup Language (XHTML)
  - Documents contain markup, or tags
  - Requires syntactically correct documents
- Uniform Resource Locator (URL) directs browser to resource
- Hypertext Transfer Protocol (HTTP) for transferring requests and files over the Internet
- Domain name system (DNS) server translates hostname into Internet Protocol (IP) address

6.2.1 System Architecture

- Multi-tier applications
  - Information tier
    - Also called data tier or bottom tier
    - Maintains data for application
  - Middle tier
    - Implements presentation logic and enforces business rules
    - Controller logic processes client requests and retrieves data
  - Client tier
    - Also called top tier
    - Application’s user interface

6.2.1 System Architecture

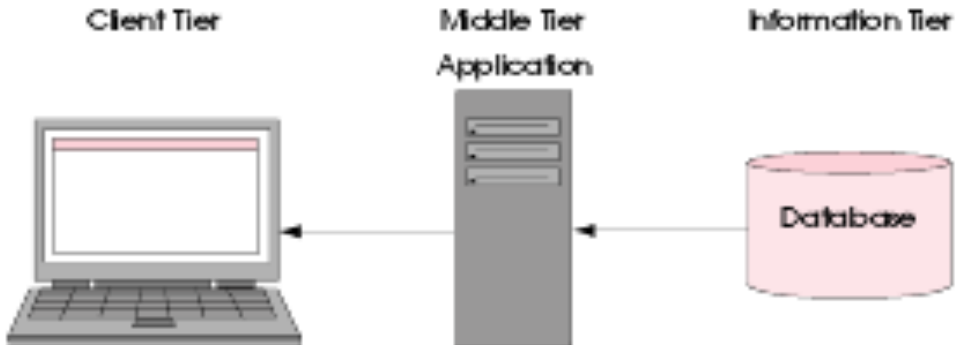


Fig. 6.1 Three-tier application model.

6.2.2 Accessing Web Servers

- Request documents from local or remote Web servers
- Ways to request a document
  - Local Web server : machine name or **`localhost`**
  - Remote Web server: specify server’s domain name or IP address
- Domain name
  - Represents groups of hosts on the Internet
  - combines with top-level domain (TLD) to form fully qualified hostname

6.2.2 HTTP Transactions

- HTTP request types
  - Get : sends form content as part of URL
  - Post : users cannot see sent data
- HTTP headers
  - provide additional information about data sent in response to request
  - Multipurpose Internet Mail Extensions (MIME) : Internet standard that specifies how messages should be formatted

6.2.2 HTTP Transactions

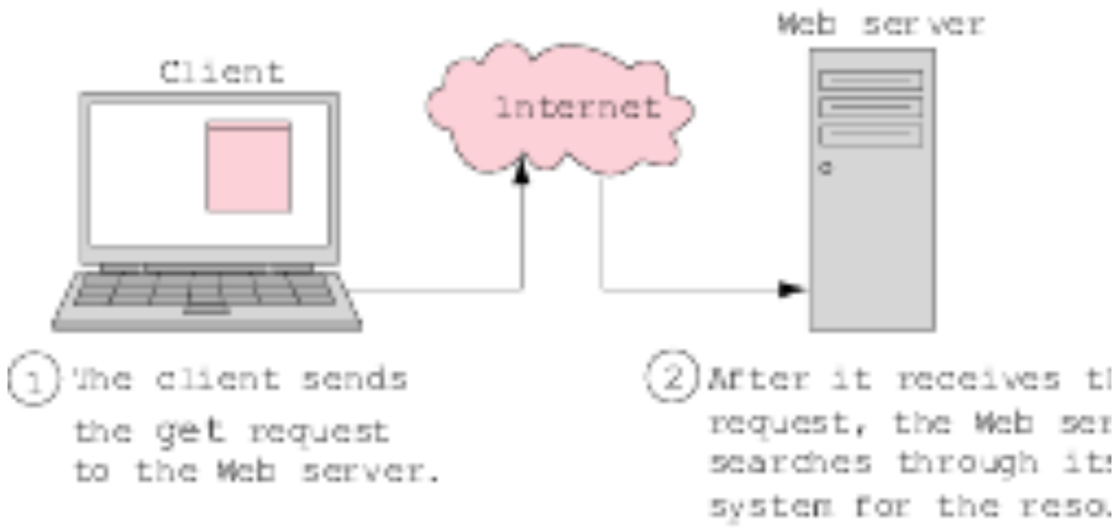

Web server
Client
Internet
1 The client sends the get request to the Web server.
2 After it receives the request, the Web server searches through its system for the reso

Fig. 6.2 Client interacting with server and Web server. Step 1: The request, **`GET /books/downloads.html HTTP/1.1`**.

6.2.2 HTTP Transactions

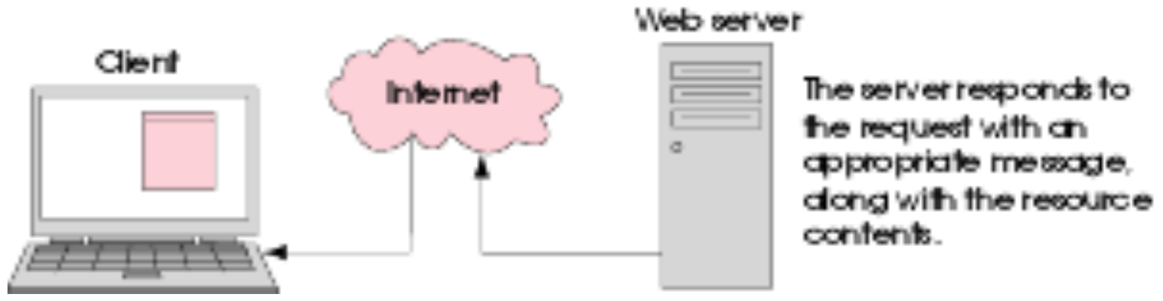

Client
Internet
Web server
The server responds to the request with an appropriate message, along with the resource contents.

Fig. 6.2 Client interacting with server and Web server. Step 2: The HTTP response, `HTTP/1.1 200 OK`.

6.2.2 Simple CGI Script

- Two types of scripting
  - Server-side scripting (i.e., CGI scripts) manipulate server resources
  - Client-side scripting (i.e, JavaScript) accesses browser features, manipulates browser documents, validates user input, etc.
- Server-side scripts
  - Execute on server
  - Usually generate custom responses for clients
  - Wider range of programmatic capabilities than client-side scripts

Outline

fig06_03.py

```
#!c:\Python\python.exe
# Fig. 6.3: fig06_03.py
# Displays current date and time in Web browser.

import time

def printHeader( title ):
   print """Content-type: text/html

<?xml version = "1.0" encoding = "UTF-8"?>
<!DOCTYPE html PUBLIC
  "-//W3C//DTD XHTML 1.0 Strict//EN"
  "DTD/xhtml1-strict.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title>%s</title></head>

<body>""" % title

printHeader( "Current date and time" )
print time.ctime( time.time() )
print "</body></html>"
```

Directive specifies location of server's Python interpreter

Print HTTP header

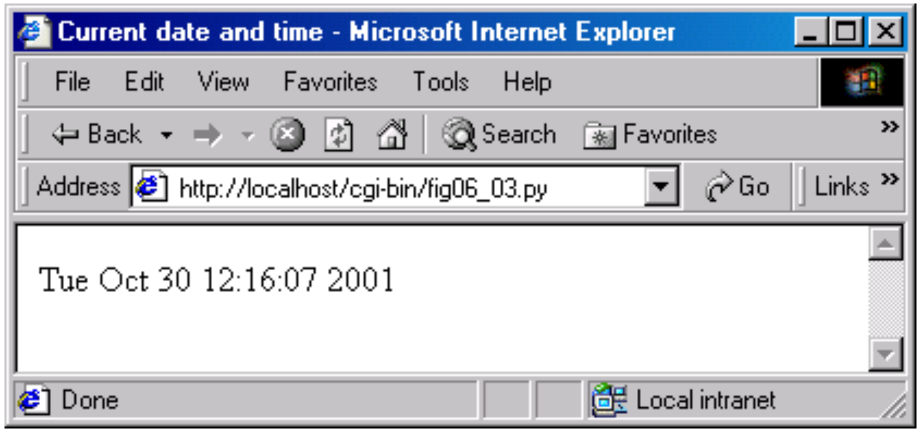
Blank line signifies end of HTTP header

Print XML declaration

Print XHTML document header

Returns seconds since epoch

Formats seconds to human-readable time


Current date and time - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/cgi-bin/fig06_03.py
Go Links
Done
Local intranet

Tue Oct 30 12:16:07 2001

6.3 Simple CGI Script

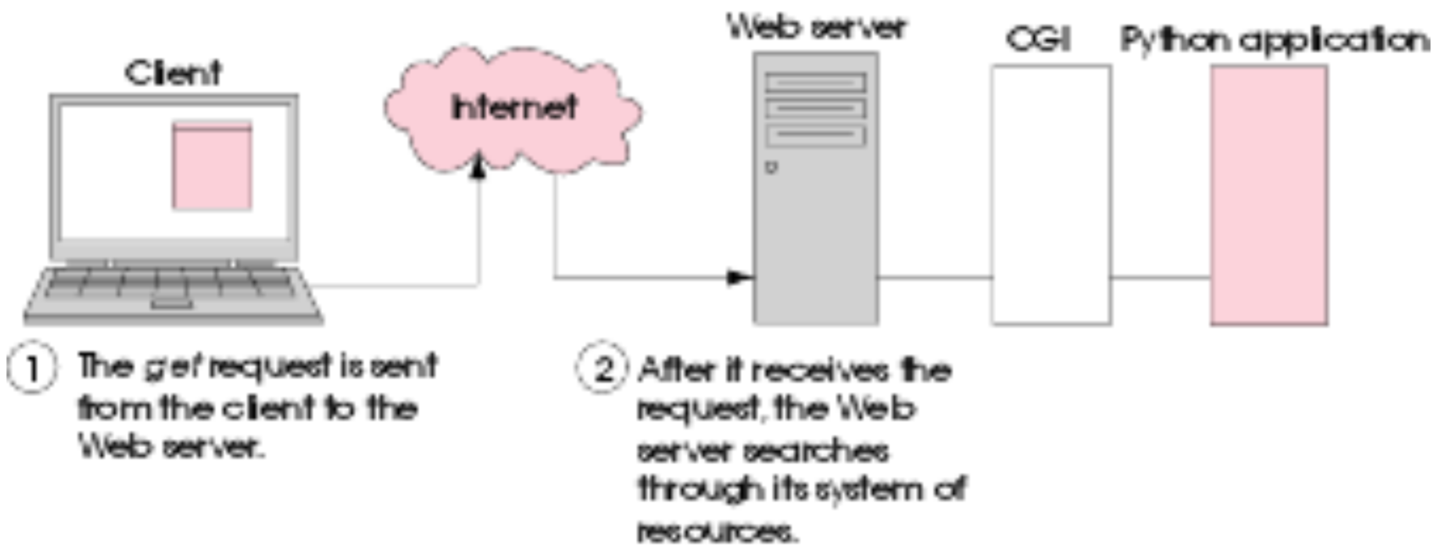

Client
Internet
Web server
CGI
Python application
1 The *get* request is sent from the client to the Web server.
2 After it receives the request, the Web server searches through its system of resources.

Fig. 6.4 Step 1: The **GET** request, **GET /cgi-bin/fig06_02.py HTTP/1.1**. (Part 1 of 4.)

6.3 Simple CGI Script

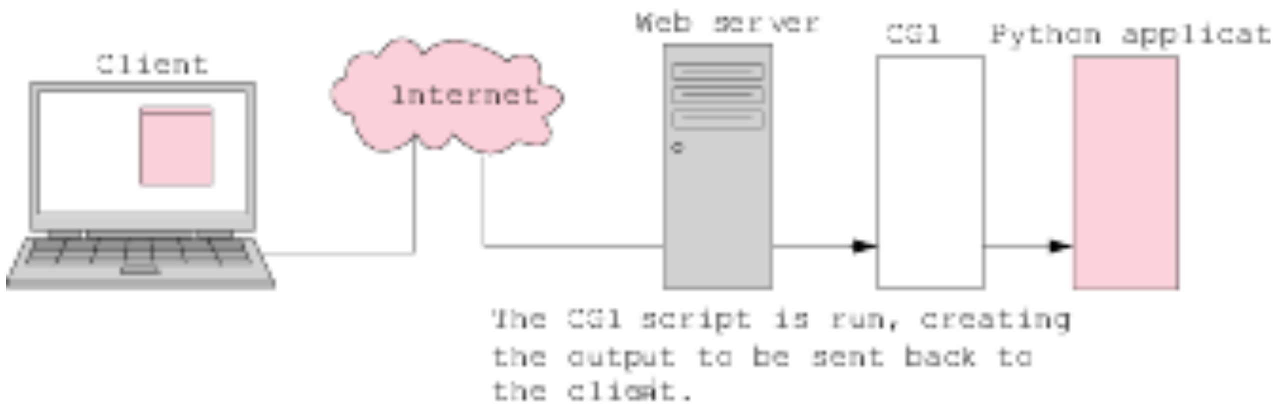

Client
Internet
Web server
CGI
Python applicat
The CGI script is run, creating the output to be sent back to the client.

Fig. 6.4 Step 2: The Web server starts the CGI script. (Part 2 of 4.)

6.3 Simple CGI Script

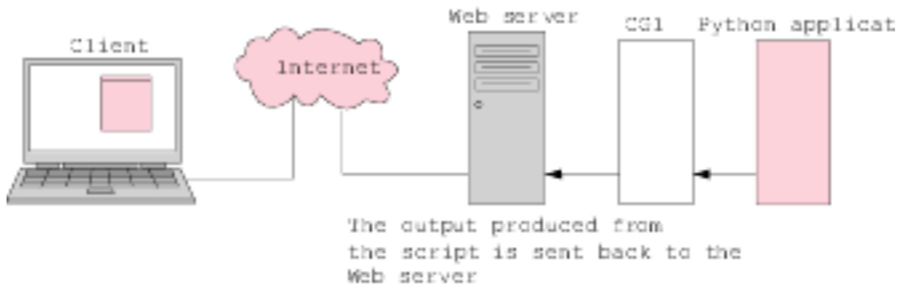

Client
Internet
Web server
CGI
Python applicat
The output produced from the script is sent back to the Web server

Fig. 6.4 Step 3: The output of the script is sent to the Web server. (Part 3 of 4.)

6.3 Simple CGI Script

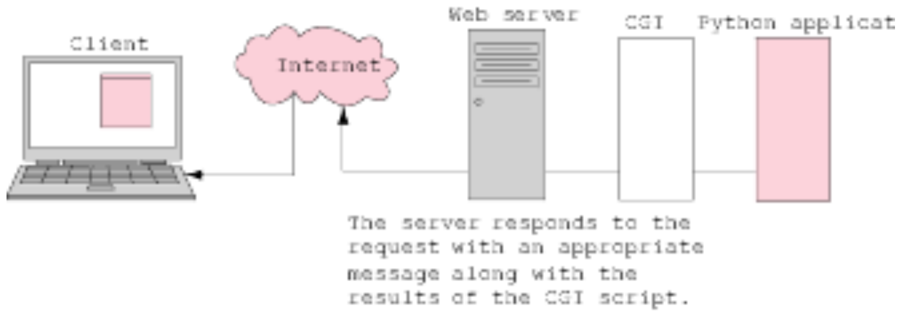

Client
Internet
Web server
CGI
Python applicat
The server responds to the request with an appropriate message along with the results of the CGI script.

Fig. 6.4 Step 4: The HTTP response, **`HTTP/1.1 200 OK`**. (Part 4 of 4.)

Outline

ig06_05.py

```
#!c:\Python\python.exe
# Fig. 6.5: fig06_05.py
# Program displaying CGI environment variables.

import os
import cgi

def printHeader( title ):
   print """Content-type: text/html

<?xml version = "1.0" encoding = "UTF-8"?>
<!DOCTYPE html PUBLIC
  "-//W3C//DTD XHTML 1.0 Strict//EN"
  "DTD/xhtml1-strict.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title>%s</title></head>

<body>""" % title

rowNumber = 0
backgroundColor = "white"

printHeader( "Environment Variables" )
print """<table style = "border: 0">"""

# print table of cgi variables and values
for item in os.environ.keys():
  rowNumber += 1

  if rowNumber % 2 == 0
     backgroundColor = "white"
  else:                          # odd row numbers are grey
     backgroundColor = "lightgrey"

```

Module **os** provides access to environment variables

Module **cgi** provides form processing and text formatting capabilities

Start table with **<table>** tag

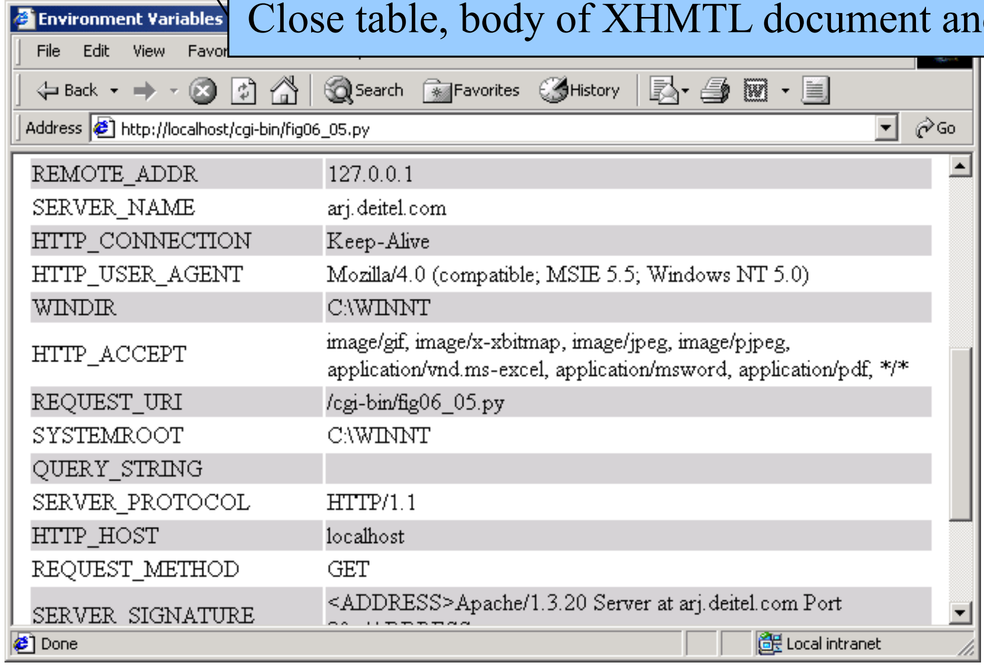
os.environ acts like a dictionary of environment variables and values

```
    print """<tr style = "background-color: %s">
    <td>%s</td><td>%s</td></tr>""" % ( backgroundColor,
       cgi.escape( item ), cgi.escape( os.environ[ item ] ) )

 print """</table></body></html>"""
```

Table row for each environment variables and its value

Function **`cgi.escape`** takes a string and returns a properly formatted XHMTL string

Close table, body of XHMTL document and XHTML document


Environment Variables
File Edit View Favor
Back Search Favorites History
Address http://localhost/cgi-bin/fig06_05.py
Go
Done
Local intranet

REMOTE_ADDR	127.0.0.1
SERVER_NAME	arj.deitel.com
HTTP_CONNECTION	Keep-Alive
HTTP_USER_AGENT	Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)
WINDIR	C:\WINNT
HTTP_ACCEPT	image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application/msword, application/pdf, */*
REQUEST_URI	/cgi-bin/fig06_05.py
SYSTEMROOT	C:\WINNT
QUERY_STRING	
SERVER_PROTOCOL	HTTP/1.1
HTTP_HOST	localhost
REQUEST_METHOD	GET
SERVER_SIGNATURE	<ADDRESS>Apache/1.3.20 Server at arj.deitel.com Port

6.4 Sending Input to a CGI Script

- **`QUERY_STRING`** variable contains extra information appended to a URL in a **`GET`** request, following a question mark (**`?`**)
- Question mark (**`?`**) serves as delimiter between source and query string
- Name-value pairs in the query string separated by ampersands (**`&`**)

Outline

fig06_06.py

```
#!c:\Python\python.exe
# Fig. 6.6: fig06_06.py
# Example using QUERY_STRING.

import os
import cgi

def printHeader( title ):
   print """Content-type: text/html

<?xml version = "1.0" encoding = "UTF-8"?>
<!DOCTYPE html PUBLIC
  "-//W3C//DTD XHTML 1.0 Strict//EN"
  "DTD/xhtml1-strict.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title>%s</title></head>

<body>""" % title

printHeader( "QUERY_STRING example" )
print "<h1>Name/Value Pairs</h1>"

query = os.environ[ "QUERY_STRING" ]

if len( query ) == 0:
  print """<p><br />
     Please add some name-value pairs to the URL above.
     Or try
     <a href = "fig06_06.py?name=Veronica&amp;age=23">this</a>.
     </p>"""
else:
  print """<p style = "font-style: italic">
     The query string is '%s'.</p>""" % cgi.escape( query )
  pairs = cgi.parse_qs( query )

```

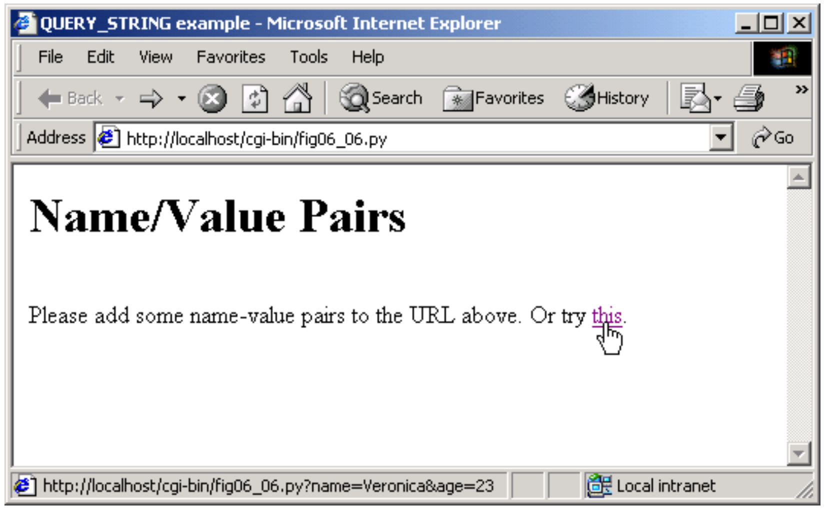
Contains information appended to URL in **`GET`** request

Parses query string and returns dictionary of its name-value pairs

Outline

```
   for key, value in pairs.items():
      print "<p>You set '%s' to value %s</p>"" % \
         ( key, value )

print "</body></html>"
```

fig06_06.py

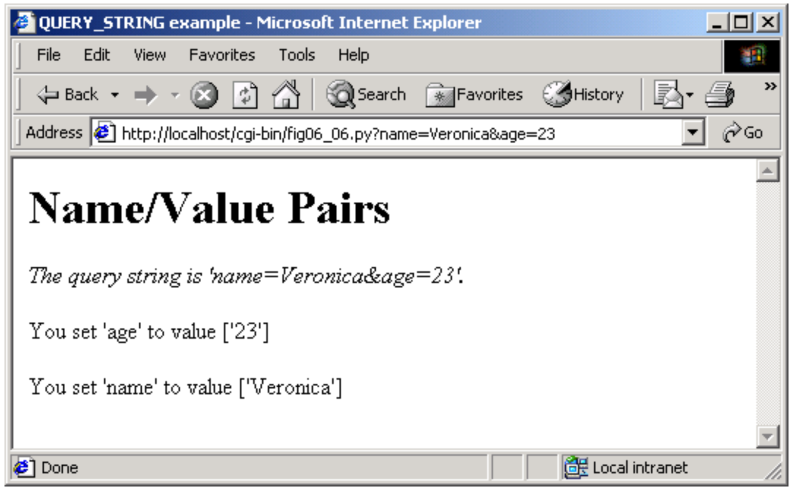

QUERY_STRING example - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://localhost/cgi-bin/fig06_06.py
Go
Local intranet

Name/Value Pairs

Please add some name-value pairs to the URL above. Or try this.

http://localhost/cgi-bin/fig06_06.py?name=Veronica&age=23

fig06_06.py

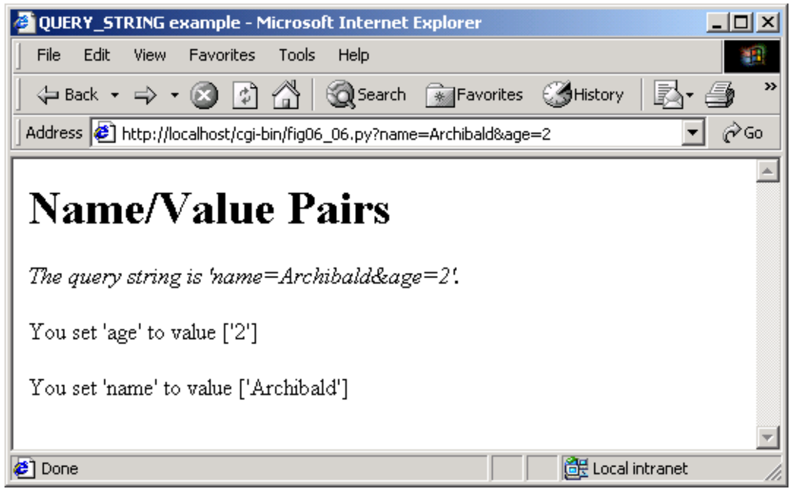

QUERY_STRING example - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://localhost/cgi-bin/fig06_06.py?name=Veronica&age=23
Go
Done
Local intranet

Name/Value Pairs

The query string is 'name=Veronica&age=23'.

You set 'age' to value ['23']

You set 'name' to value ['Veronica']


QUERY_STRING example - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://localhost/cgi-bin/fig06_06.py?name=Archibald&age=2
Go
Done
Local intranet

Name/Value Pairs

The query string is 'name=Archibald&age=2'.

You set 'age' to value ['2']

You set 'name' to value ['Archibald']

6.5 Using XHMTL Forms to Send Input and Using Module `cgi` to Retrieve Form Data

- Web page forms provide an intuitive way for users to input information to CGI scripts
- **`<form>`** and **`</form>`** tags surround an XHTML form
  - Attribute **`action`** specifies operation to perform when users submit the form
  - Attribute **`method`** : either get or post
    - Get sends data to CGI script via **`QUERY_STRING`** environment variable
    - Post sends data to CGI script via standard input and sets environment variable **`CONTENT_LENGTH`** to number of characters that were posted

6.5 Using XHMTL Forms to Send Input and Using Module `cgi` to Retrieve Form Data

Tag name	`type` attribute (for `<input>` tags)	Description
<input>	`button`	A standard push button.
	`checkbox`	Displays a checkbox that can be checked (true) or unchecked (false).
	`file`	Displays a text field and button so the user can specify a file to upload to a Web server. The button displays a file dialog that allows the user to select a file.
	`hidden`	Hides data information from clients so that hidden form data can be used only by the form handler on the server.
	`image`	The same as **`submit`**, but displays an image rather than a button.
	`password`	Like **`text`**, but each character typed appears as an asterisk (**`*`**) to hide the input (for security).
	`radio`	Radio buttons are similar to checkboxes, except that only one radio button in a group of radio buttons can be selected at a time.
	`reset`	A button that resets form fields to their default values.

6.5 Using XHMTL Forms to Send Input and Using Module `cgi` to Retrieve Form Data

	`submit`	A push button that submits form data according to the form's **`action`**.
	`text`	Provides single-line text field for text input. This attribute is the default input type.
<select>		Drop-down menu or selection box. When used with the **`<option>`** tag, **`<select>`** specifies items to select.
<textarea>		Multiline area in which text can be input or displayed.

Fig. 6.7 XHTML form elements.

Outline

fig06_08.py

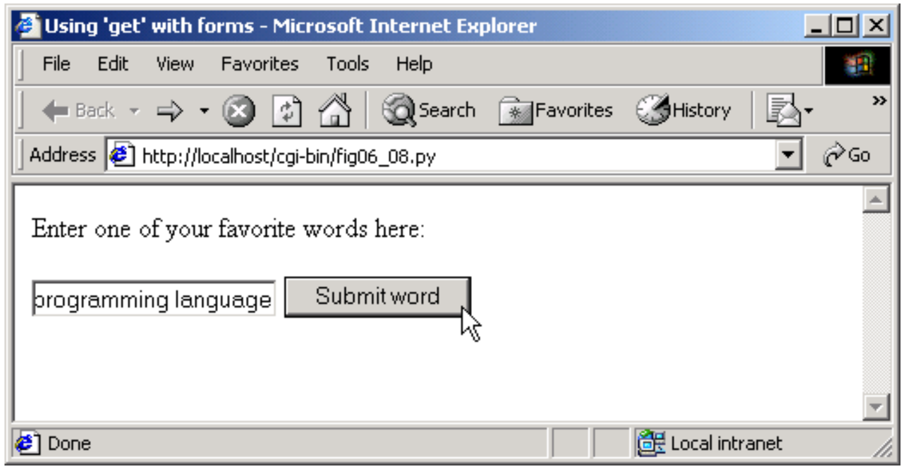
```
#!c:\Python\python.exe
# Fig. 6.8: fig06_08.py
# Demonstrates get method with an XHTML form.

import cgi

def printHeader( title ):
   print """Content-type: text/html

<?xml version = "1.0" encoding = "UTF-8"?>
<!DOCTYPE html PUBLIC
  "-//W3C//DTD XHTML 1.0 Strict//EN"
  "DTD/xhtml1-strict.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title>%s</title></head>

<body>""" % title

printHeader( "Using 'get' with forms" )
print """<p>Enter one of your favorite words here:<br /></p>
  <form method = "get" action = "fig06_08.py">
     <p>
     <input type = "text" name = "word" />
     <input type = "submit" value = "Submit word" />
     </p>
  </form>"""

pairs = cgi.parse()

if pairs.has_key( "word" ):
  print """<p>Your word is:
     <span style = "font-weigh
     % cgi.escape( pairs[ "word" ][ 0 ] )

print "</body></html>"
```

Parse form data and return a dictionary

Input data keyed by `name` attribute of `input` element

Called in case form data includes special characters

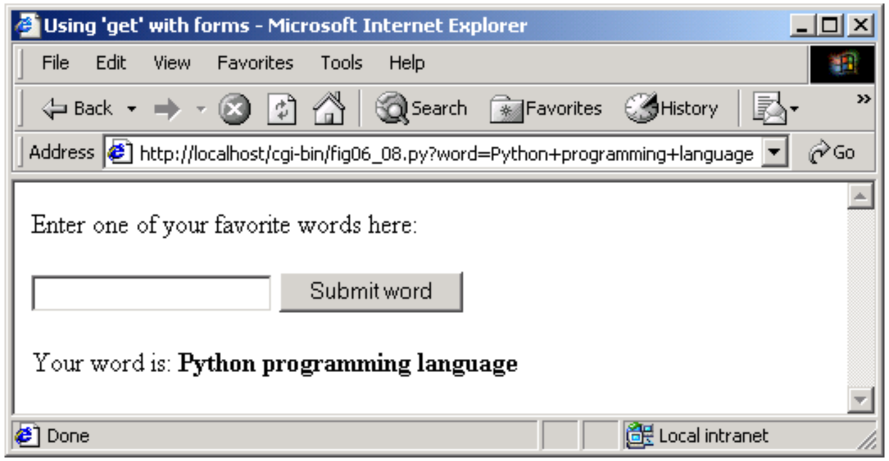
fig06_08.py



Using 'get' with forms - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://localhost/cgi-bin/fig06_08.py
Go
Done
Local intranet

Enter one of your favorite words here:

programming language [Submit word]



Using 'get' with forms - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://localhost/cgi-bin/fig06_08.py?word=Python+programming+language
Go
Done
Local intranet

Enter one of your favorite words here:

[Submit word]

Your word is: **Python programming language**

Outline

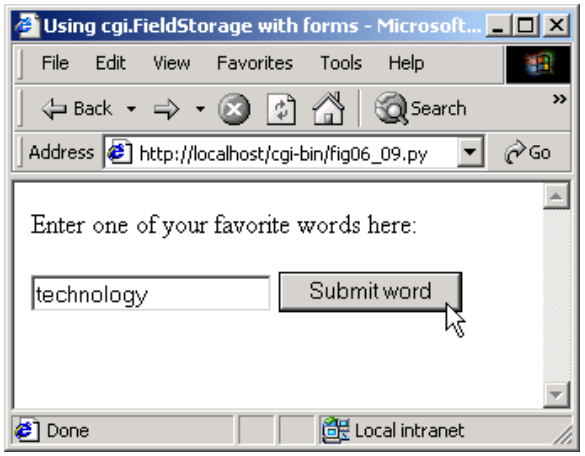
fig06_09.py

```
#!c:\Python\python.exe
# Fig. 6.9: fig06_09.py
# Demonstrates post method with an XHTML form.

import cgi

def printHeader( title ):
   print """Content-type: text/html

<?xml version = "1.0" encoding = "UTF-8"?>
<!DOCTYPE html PUBLIC
  "-//W3C//DTD XHTML 1.0 Strict//EN"
  "DTD/xhtml1-strict.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title>%s</title></head>

<body>""" % title

printHeader( "Using 'pos
print """<p>Enter one of your favorite words here:<br /></p>
  <form method = "post" action = "fig06_09.py">
     <p>
     <input type = "text" name = "word" />
     <input type = "submit" value = "Submit word" />
     </p>
  </form>"""

pairs = cgi.parse()

if pairs.has_key( "word" ):
  print """<p>Your word is:
     <span style = "font-weight: bold">%s</span></p>""" \
     % cgi.escape( pairs[ "word" ][ 0 ] )

print "</body></html>"
```

Post request attaches form content to the end of a HTTP request

Outline

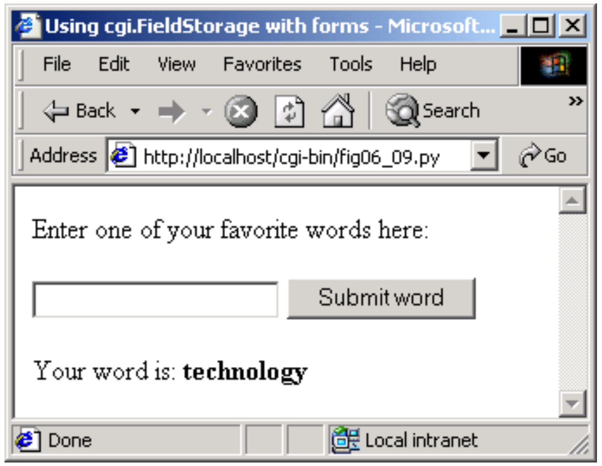
fig06_09.py


Using cgi.FieldStorage with forms - Microsoft...
File Edit View Favorites Tools Help
Back
Search
Address
http://localhost/cgi-bin/fig06_09.py
Go
Done
Local intranet

Enter one of your favorite words here:

technology

Submit word


Using cgi.FieldStorage with forms - Microsoft...
File Edit View Favorites Tools Help
Back
Search
Address
http://localhost/cgi-bin/fig06_09.py
Go
Done
Local intranet

Enter one of your favorite words here:

Submit word

Your word is: **technology**

6.5 Using `cgi.FieldStorage` to Read Input

- Module **`cgi`** provides class FieldStorage to parse forms

Outline

fig06_10.py

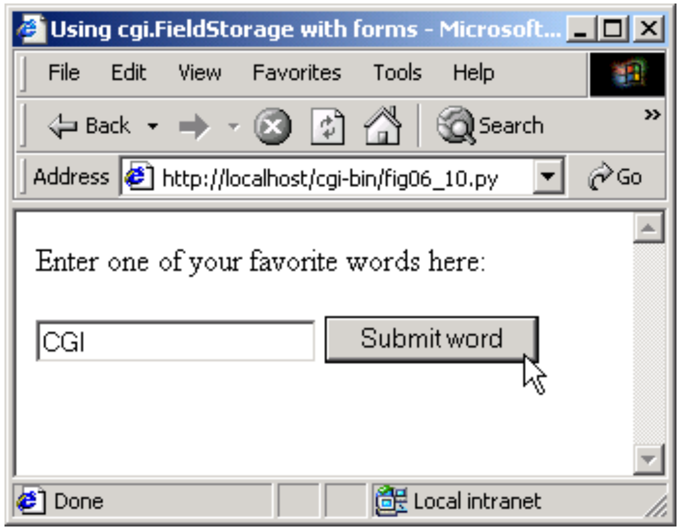
```
#!c:\Python\python.exe
# Fig. 6.10: fig06_10.py
# Demonstrates use of cgi.FieldStorage an with XHTML form.

import cgi

def printHeader( title ):
   print """Content-type: text/html

<?xml version = "1.0" encoding = "UTF-8"?>
<!DOCTYPE html PUBLIC
  "-//W3C//DTD XHTML 1.0 Strict//EN"
  "DTD/xhtml1-strict.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title>%s</title></head>

<body>""" % title

printHeader( "Using cgi.FieldStorage with forms" )
print """<p>Enter one of your favorite words here:<br /></p>
  <form method = "post" action = "fig06_10.py">
     <p>
     <input type = "text" name = "word" />
     <input type = "submit" value = "Submit word" />
     </p>
  </form>"""

form = cgi.FieldStorage()

if form.has_key( "word" ):
  print """<p>Your word is:
     <span style = "font-weight: bold">%s</span></p>""" \
     % cgi.escape( form[ "word" ].value )

print "</body></html>"
```

`cgi.FieldStorage` object stores form data in a dictionary

Form data keyed by the value of each **`input`** element's **`name`** attribute

Outline

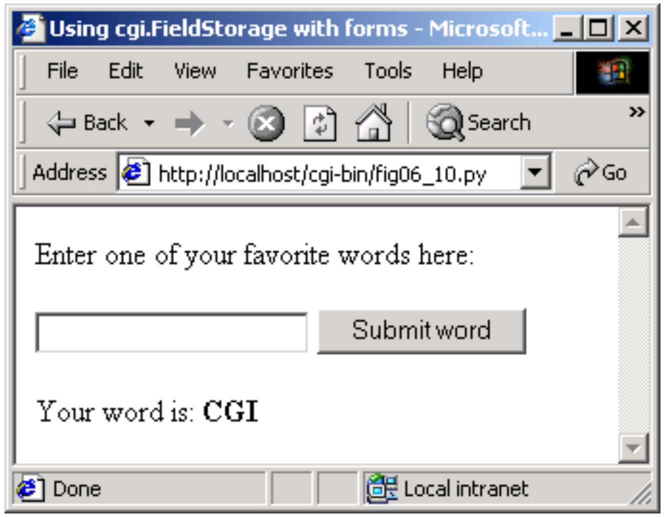
fig06_10.py


Using cgi.FieldStorage with forms - Microsoft...
File Edit View Favorites Tools Help
Back Search
Address http://localhost/cgi-bin/fig06_10.py
Go
Done
Local intranet

Enter one of your favorite words here:

CGI

Submit word


Using cgi.FieldStorage with forms - Microsoft...
File Edit View Favorites Tools Help
Back Search
Address http://localhost/cgi-bin/fig06_10.py
Go
Done
Local intranet

Enter one of your favorite words here:

Submit word

Your word is: **CGI**

6.5 Other HTTP Headers

- **`Content-type`** header specifies how document is processed
- **`Refresh`** header redirects client to new location or refreshes current page
- **`Location`** header indicates redirection performed on server side
- **`Status`** header tells server to output status-header line

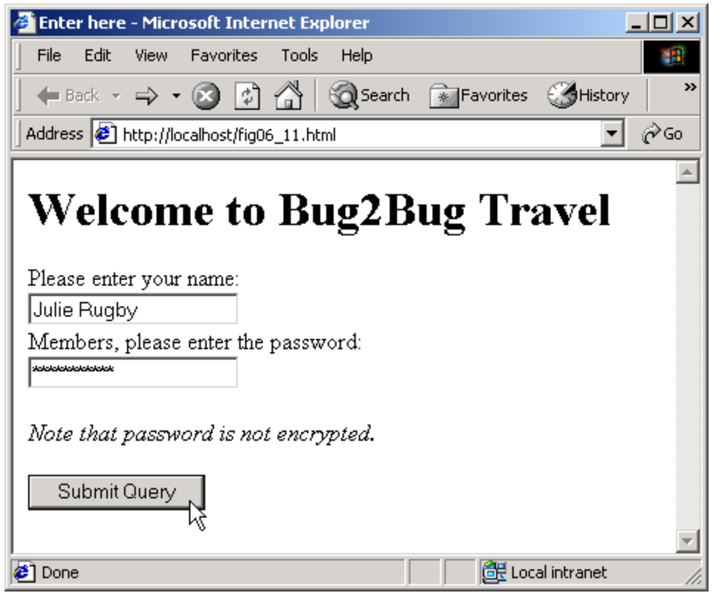
Outline

fig06_11.html

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!DOCTYPE html PUBLIC
   "-//W3C//DTD XHTML 1.0 Strict//EN"
   "DTD/xhtml1-strict.dtd">
<!-- Fig. 6.11: fig06_11.html    -->
<!-- Bug2Bug Travel log-in page. -->

<html xmlns = "http://www.w3.org/1999/xhtml">
   <head><title>Enter here</title></head>

  <body>
     <h1>Welcome to Bug2Bug Travel</h1>

     <form method = "post" action = "/cgi-bin/fig06_12.py">

        <p>Please enter your name:<br />
        <input type = "text" name = "name" /><br />

        Members, please enter the password:<br />
        <input type = "password" name = "password" /><br />
        </p>

        <p style = "font-size: em - 1; font-style: italic" >
        Note that password is not encrypted.<br /><br />
        <input type = "submit" />
        </p>

     </form>
  </body>
</html>
```

CGI script called when user submits form

fig06_11.html


Enter here - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://localhost/fig06_11.html
Go
Done
Local intranet

Welcome to Bug2Bug Travel

Please enter your name:

Julie Rugby

Members, please enter the password:

****

Note that password is not encrypted.

Submit Query

Outline

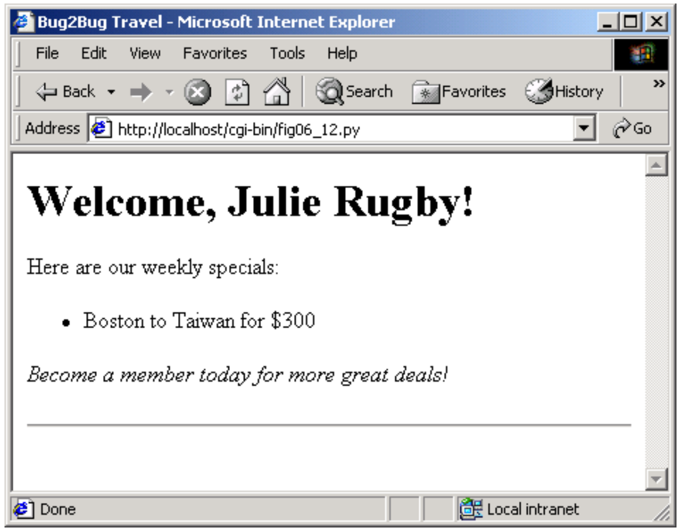
fig06_12.py

```
#!c:\Python\python.exe
# Fig. 6.12: fig06_12.py
# Handles entry to Bug2Bug Travel.

import cgi

def printHeader( title ):
   print """Content-type: text/html

<?xml version = "1.0" encoding = "UTF-8"?>
<!DOCTYPE html PUBLIC
  "-//W3C//DTD XHTML 1.0 Strict//EN"
  "DTD/xhtml1-strict.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title>%s</title></head>

<body>""" % title

form = cgi.FieldStorage()

if not form.has_key( "name" ):
  print "Location: /fig06_11.html\n"
else:
  printHeader( "Bug2Bug Travel" )
  print "<h1>Welcome, %s!</h1>" % form[ "name" ].value
  print """<p>Here are our weekly specials:<br /></p>
     <ul><li>Boston to Taiwan for $300</li></ul>"""

  if not form.has_key( "password" ):
     print """<p style = "font-style: ital
        Become a member today for more great deals!</p>"""
  elif form[ "password" ].value == "Coast2Coast":
     print """<hr />
        <p>Current specials just for members:<br /></p>
        <ul><li>San Diego to Hong Kong for $250</li></ul>"""
```

User did not enter login name

User entered password

fig06_12.py

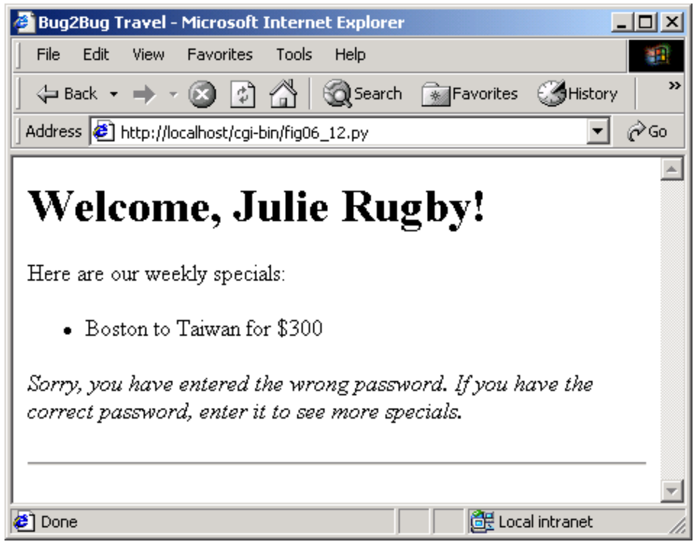

Bug2Bug Travel - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://localhost/cgi-bin/fig06_12.py
Go
Done
Local intranet

Welcome, Julie Rugby!

Here are our weekly specials:

- Boston to Taiwan for $300

Become a member today for more great deals!

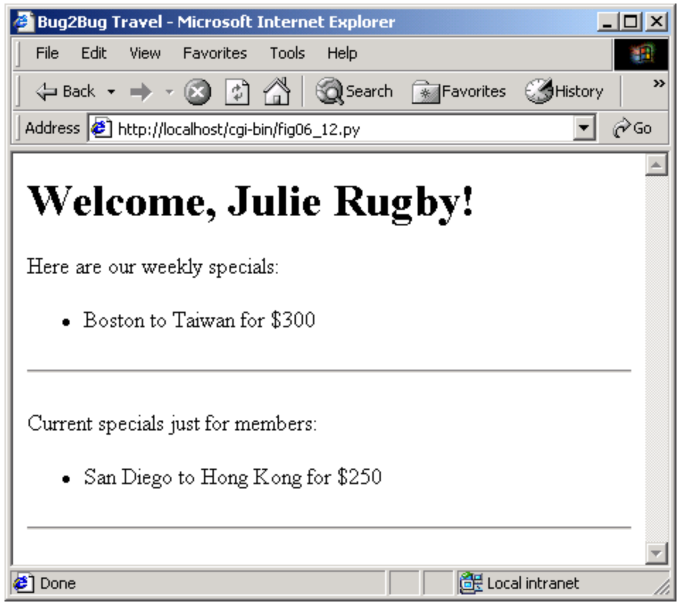

Bug2Bug Travel - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://localhost/cgi-bin/fig06_12.py
Go
Done
Local intranet

Welcome, Julie Rugby!

Here are our weekly specials:

- Boston to Taiwan for $300

Sorry, you have entered the wrong password. If you have the correct password, enter it to see more specials.


Bug2Bug Travel - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://localhost/cgi-bin/fig06_12.py
Go
Done
Local intranet

Welcome, Julie Rugby!

Here are our weekly specials:

- Boston to Taiwan for $300

Current specials just for members:

- San Diego to Hong Kong for $250