Information Infrastructure II

INFO I211 – Spring 2014 – Sections 18530 & 22719

Lecture 16 – 2014.03.24 & 2014.03.25

Instructor:
Mitja Hmeljak,
http://mypage.iu.edu/~mitja
mitja@indiana.edu

reminder: PLTL Sessions

PLTL sessions take place every week

Monday and Wednesday

7:30 – 8:45 PM

Informatics West 107

PLTL leader Jonathan Ruddell <ruddellj> :

"Voluntary, but proven successful worldwide"

1 extra credit point per session

(with a limit on max. number of possible points earned)

CGI scripts, reviewed

CGI is the **C**ommon **G**ateway **I**nterface protocol

CGI is an interface to pass requests from a web server to an executable program, and return the program's results back to a web browser.

The Yahoo finance page and the Boston Market locator we saw at previous lectures were both **CGI scripts**.

CGI scripts can be written in many different scripting languages. We're using Python to write CGI scripts.

CGI scripts, reviewed: CGI Hello World - *hello.cgi*

```
#!/usr/bin/python
# the above line starts the Python interpreter
print "Content-type: text/html\n"
# the above line is required by CGI
print "<html><head><title>First CGI</title></head>"
print "<body>Hello World!<br></body>"
print "</html>"
```

To run this program as CGI, you need to place this file into your **~/cgi-pub/** directory and then change the rights on it *to be executable* by running this command:
chmod ugo+rx hello.cgi

You'll then be able to see the HTML page resulting from this CGI program at this URL:
http://cgi.soic.indiana.edu/~yourusername/hello.cgi

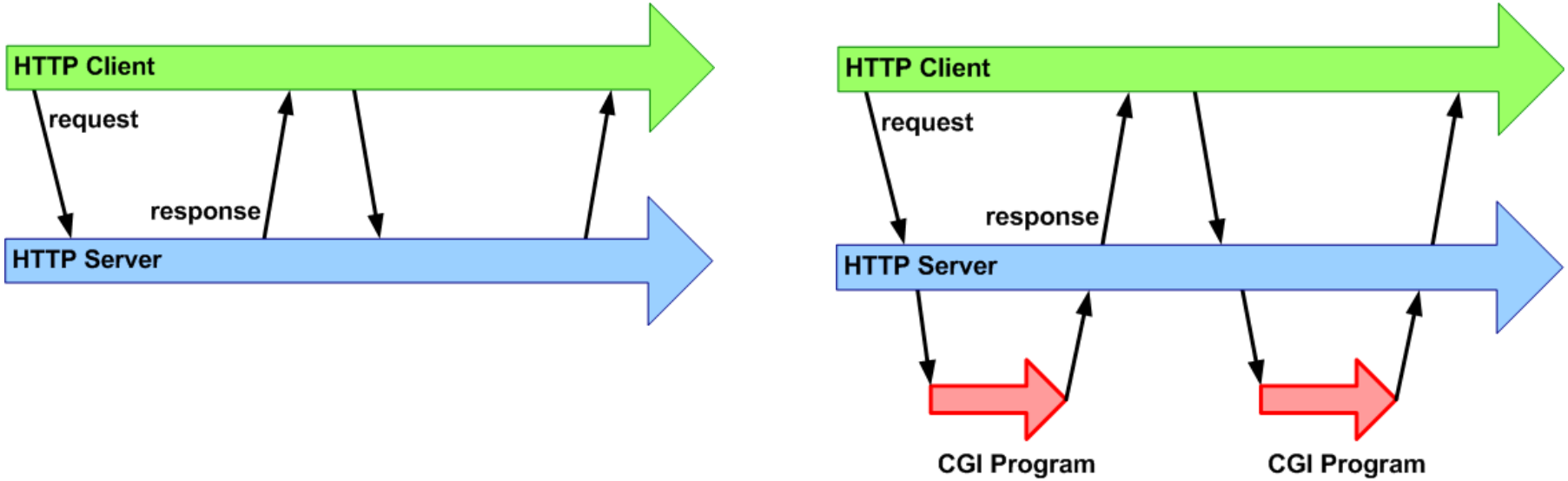
Content-type: text/html ← this first line tells the browser to expect a text-based HTML file content, as opposed to some other form of data. These are called **MIME types**. You can see a list of content types here:
http://en.wikipedia.org/wiki/Internet_media_type#List_of_common_media_types

The **\n** at the end of the MIME type line is needed to tell the browser that the actual content begins there!

Introduction to CGI scripts, part III

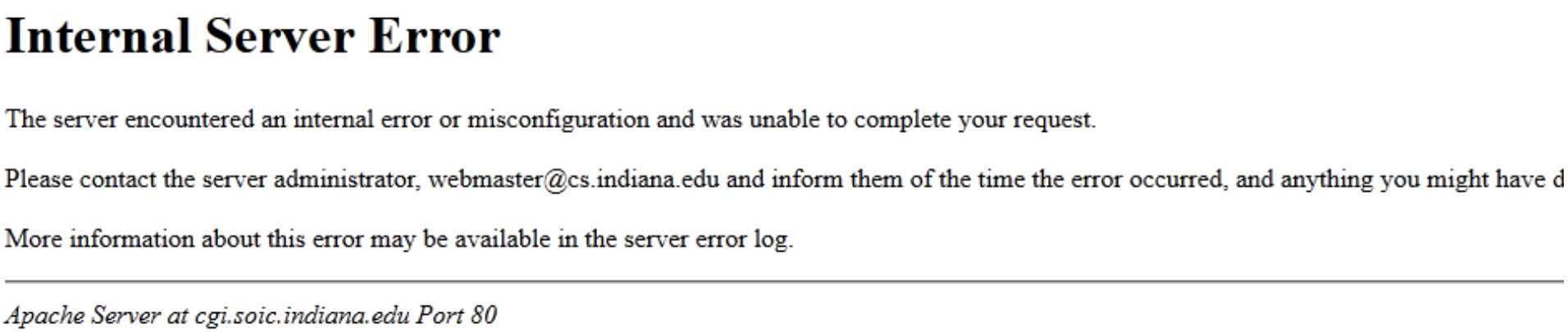
Common Gateway Interface

An interface to pass web requests off to an executable program and return results to a browser



And if we ~~bake~~ make a mistake?

If you do anything wrong, you'll see this:



Internal Server Error

The server encountered an internal error or misconfiguration and was unable to complete your request.

Please contact the server administrator, webmaster@cs.indiana.edu and inform them of the time the error occurred, and anything you might have d

More information about this error may be available in the server error log.

Apache Server at cgi.soic.indiana.edu Port 80

Unfortunately, it is not a very detailed error message...

(this – in rare circumstances – may also cause automatic resetting of the ***.py*** file permissions, so you may have to run the ***chmod*** command again)

Errors in CGI Python scripts?

catch any errors and display them in your browser with a try-except statement

```
#!/usr/bin/python

print 'Content-type: text/html\n'

def cgi_stuff():
    number += number
    # this line generates an error so we can see what happens

try:
    import traceback, sys, os, cgi
    sys.stderr = sys.stdout
    cgi_stuff()

except Exception as e:
    print '<html><head><title>Error in Script</title></head><body>'
    print '<h1>traceback printout</h1>'
    print '<pre>'
    print str(e)
    traceback.print_exc()
    print '</pre>'
    print '</body></html>'
```

Another CGI Script: *fancy.cgi*

Download **python-logo.png** from Oncourse **I211 Resources/sample-code** and try this:

```
#!/usr/bin/python

print 'Content-type: text/html\n'

text = """
<html>
   <head><title>A Second CGI Script</title></head>
   <body>
      <h1>A Second CGI Script</h1>
      <p>Hello, CGI World!</p>
      <img src="python-logo.png" border="1" alt="Python Logo">
      <hr>
   </body>
</html>
"""

print text
```

Then upload both files to your ***~/cgi-pub/*** directory, and ***chmod ugo+rx fancy.cgi*** !

Tables.cgi (Group Work)

Write a CGI script that uses Python to print out a multiplication table in HTML, 10 Rows and 8 Columns.

For an example of what yours should look like, go here:

http://cgi.soic.indiana.edu/~mitja/I211/lecture16/cgi_tables.cgi

Make sure that you could easily:

1. change the program to print out a different size of table,
2. print out an "addition table"

Tables.cgi (Solution)

```
#!/usr/bin/python

print 'Content-type: text/html\n'

text = """
<html>
   <head><title>Tables in CGI</title></head>
   <body>
      <h3>Your table should look like this:</h3>
      <table border="1">
"""

for i in range(10):
   text += "<tr>"
   for j in range(10):
      text += "<td>" + str((i+1)*(j+1)) + "</td>"
   text += "</tr>"

text += """
      </table>
   </body>
</html>
"""

print text
```

Introduction to HTML *user input* for CGI scripts

We can get ***input from users*** online by using **HTML forms**!

These have the same sorts of elements as other GUI interfaces (such as Python's Tkinter) :

- Text boxes
- Radio buttons
- Text areas
- Buttons
- Check boxes

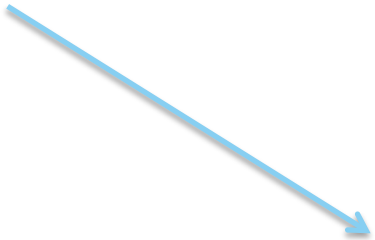
Introduction to CGI scripts, part III

HTML form elements must be enclosed in <form> tags.

The <form> tag has an

- ***action attribute*** that specifies...
- ...the ***URL*** that will ***receive*** the data when the HTML *posts* it.

For example:



```
<form action="send_to.cgi" method="post">
Name: <input type="text" name="name" /><br />
</form>
```

A simple HTML Form

```
<html>
<head><title>First Interactive Form</title></head>
<body>

<form action="name.cgi" method="post">
      Please enter your name:
      <input type="text" name="name" /><br />

      <input type="submit" value="Submit" />
</form>

</body>
</html>
```

Save this as
name.html and
chmod ugo+r

Simple Form CGI Handler

```
#!/usr/bin/python
print 'Content-type: text/html\n'
import cgi
form = cgi.FieldStorage()   # parses form data

html = """
<html>
   <head><title>Form in CGI</title></head>
   <body>
      <h1>Greetings!</h1>
      <hr>
      <p>%s</p> <!-- this says we'll insert this value later -->
      <hr>
   </body>
</html>
"""

if not 'name' in form:
   # this triggers if name was blank
   print html % "who are you?"
else:
   print html % ("hello, " + form['name'].value)
```

Save this as *name.cgi*, and don't forget to run ***chmod ugo+x*** on it