Information Infrastructure II

INFO I211 – Spring 2014 – Sections 18530 & 22719

Lecture 17 – 2014.03.26 & 2014.03.27

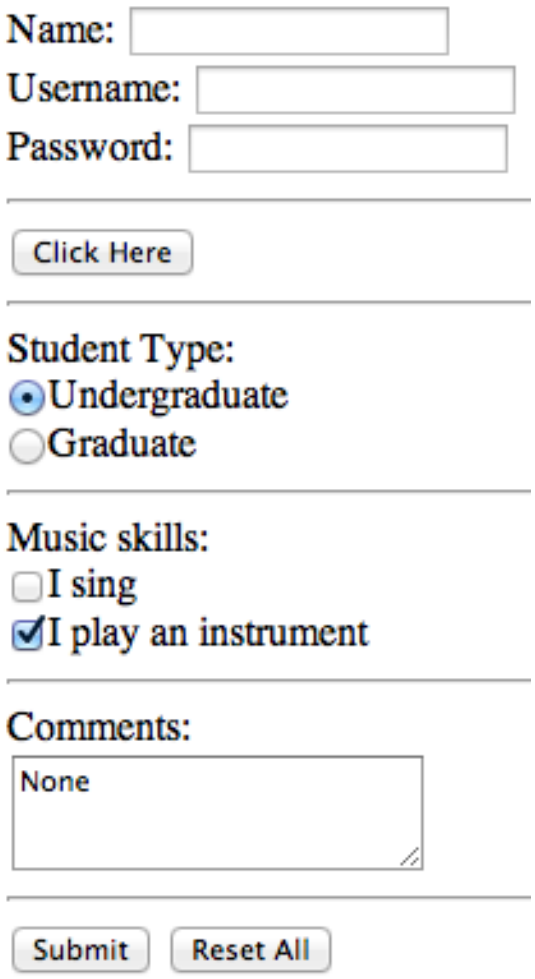
Instructor:
Mitja Hmeljak,
http://mypage.iu.edu/~mitja
mitja@indiana.edu

Introduction to HTML *user input* for CGI scripts

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

These have the same kind of elements
as other GUI interfaces
(e.g. such as Python's Tkinter) :

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



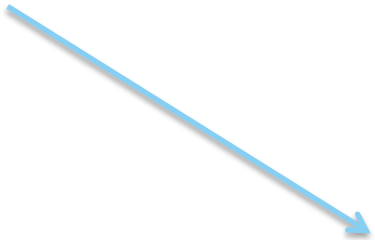
Introduction to CGI scripts, part III

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+rx*** on it!

Handling CGI from Python
a.k.a. HTML calls, Python responds

Python programs can use the *cgi* module (part of the standard library) to parse data arriving from HTML *forms* thus:

```
import cgi

form = cgi.FieldStorage()
```

← ***form*** now contains a ***dictionary-like object***

- where the form element's *name attribute* is the dictionary's **key**,
- and the form element's data (user-typed or *value attribute*) is the dictionary **value** associated to that key.

For example, if you were to print the value of the ***form*** variable in the *name.cgi* script on the previous page, this is what you'd get:

```
FieldStorage(None, None, [MiniFieldStorage('name', 'Mitja')])
```

Introduction to CGI scripts, part III

```
<html><head><title>I211 - Form One</title></head><body>
<form action="form-processor.cgi" method="post">
   Name: <input type="text" name="name"><br>
   Username: <input type="text" name="user"><br>
   Password: <input type="password" name="pwd"><br>
   <hr>
   <input type="button" value="Click Here"><br>
   <hr>
   Student Type: <br>
   <input type="radio" name="studenttype" value="ugrad" checked>Undergraduate<br>
   <input type="radio" name="studenttype" value="grad">Graduate<br>
   <hr>
   Music skills: <br>
   <input type="checkbox" name="music" value="sing">I sing<br>
   <input type="checkbox" name="music" value="play">I play an instrument<br>
   <hr>
   Comments: <br>
   <textarea name="comments" rows="3">None</textarea><br>
   <hr>
   <input type="submit" value="Submit">
   <input type="reset" value="Reset All">
</form>
</body></html>
```

Form 2 (Group Work)

Save the **HTML form** from previous page in these lecture notes to a file named "form-GUI.html", place the file in your "lecture17" cgi-pub directory on silo.soic.indiana.edu.

Write a **Python CGI script** that receives all the data from the form, and prints it out in an HTML table, place the Python script in a file named "form-processor.py" in your your "lecture17" cgi-pub directory on silo.soic.indiana.edu.

Test that the two files work, i.e. the input provided in the HTML page has to show up in a page generated by the Python script you just wrote.

HTML and CGI

HTML Forms:

http://www.w3.org/TR/html4/interact/forms.html

http://www.w3schools.com/html/html_forms.asp

Python CGI:

http://docs.python.org/2.6/library/cgi.html

http://webpython.codepoint.net/cgi_tutorial

Calculate (Group Work)

Write an **HTML form** that allows users to input numbers on separate lines of a text field. Use radio buttons to give them the option of multiplying or adding these numbers.

Write a **Python CGI script** that receives this data and shows the result.

Your program should behave like this:

http://cgi.soic.indiana.edu/~mitja/lecture17/calculate.html

Calculate (HTML Solution)

```
<html>
<head><title>Calculate</title></head>
<body>

<form action="calculate.cgi" method="post">
  Please enter as many numbers as you'd like on separate lines: <br>
  <small><em>(one number per line, no empty lines, no non-numeric
  strings please)</em></small><br>
  <textarea name="numbers" rows="10" cols="80"></textarea><br>
  Choose an operation: <br>
  <input type="radio" name="operation" value="add" checked>Add Them
  <input type="radio" name="operation" value="mult">Multiply Them<br>
  <input type="submit" value="Submit">
</form>

</body>
</html>
```

Calculate (Python Solution)

```
#!/usr/bin/python

import cgi

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

def add_mult(nums):
    total_sum = 0
    product = 1
    for num in nums:
        total_sum += float(num)
        product *= float(num)
    return total_sum, product

def calculate():
    form = cgi.FieldStorage()

    html = """
    <html>
        <head><title>Calculate</title></head>
        <body>
            %s
        </body>
    </html>
    """
# continues...
```

```
# ... continues:

    if not 'numbers' in form or not 'operation' in form:
        print html % 'No data entered or no operation chosen!'
    else:
        numbers = form['numbers'].value
        numbers = numbers.split('\r\n')
        for num in numbers:
            num = num.strip()
        the_sum, the_prod = add_mult(numbers)
        if form['operation'].value == 'add':
            print html % ("The sum is: " + str(the_sum))
        else:
            print html % ("The product is: " + str(the_prod))

calculate()
```