

ECS 89

4/23

Server program – hw2.wsgi

```
import os, path, sys
sys.path.append(os.path.dirname(__file__))
import htmlClass
```

- First two lines figure out the directory we are in and tells Python to import things from this directory.
- Then we can do the usual import of the module. With this syntax, everything in the module is available but has to be preceded with htmlClass., eg.

```
htmlClass.HTML_Page()
```

```
def application(environ, start_response):
    status = '200 OK'
    pageObj = htmlClass.HTML_Page(environ) # make object
    output = pageObj.returnHTML() # use object to make output string
    response_headers = [('Content-type', 'text/html'),
                        ('Content-Length', str(len(output)))]
    start_response(status, response_headers)
    return [output]
```

- Typical wsgi file. pageObj is the Web page object, and then we call its one method (other than init).

htmlClass - Object initialization

```
class HTML_Page:
    def __init__(self, environ):
        self.environ = environ
        self.pageType = environ['PATH_INFO']
```

- Stores the environ dictionary
- Pulls out the url called from the dictionary

Initialization, cont.

```
self.head = self.make_head()
if pageType == '/index.html':
    self.body = self.get_index_body()
elif pageType == '/add_data.html':
    self.add_data() # puts data into file
    self.body = self.get_add_data_body()
elif pageType == '/report.html':
    self.body = self.get_report_body()
else:
    self.body = ""
```

A function each, to write body strings of each of the three Web pages.

add_data() actually adds the data to the spreadsheet.

Writing out the file

```
def returnHTML(self):
    theHTML = "<!doctype html> \n"
    theHTML += "<html> \n"
    theHTML += self.head
    theHTML += self.body
    theHTML += "</html>"
    return theHTML
```

Making the html heads

- All the heads are the same:

```
def make_head(self):
    return "<head></head> \n"
```

- You'll want to link to your stylesheet here when the time comes.

Making the index page

```
def get_index_body(self):
    return """
    <body>
    <form name="input" action="add_data.html" method="get">
    <label for="steps">Steps</label><input type="number" name="steps"
    min="0">
    <input type="submit" value="Submit">
    </body>
    """
```

- Name in form is for style or Javascript
- Add a pattern to number field, also max...
- Simpler <label> syntax

Finding the directory

- This program is running in a directory
- We want to open the file steps.csv in this directory
- Usually this is done by:

```
f = open("steps.csv", "rU")
```

- Sadly, this crashes. Why?

Printing the current data

- This program is running in a directory
- We want to open the file steps.csv in this directory
- Usually this is done by:

```
f = open("steps.csv", "rU")
```

- Sadly, this crashes. Why?

The actual program being run is Apache (a C program). The version above is trying to open the file in whatever directory Apache is in, not the one the module is in.

Finding the directory of a module

- The `__file__` variable contains the name of the current module (*not* the current program using the module).
- The `os` module contains (among other things) functions for manipulating directory names.
- `os.path.realpath(__file__)` gets the "canonical" directory name; there might be others (due to symbolic links; don't worry about this).
- `os.path.dirname()` gets just the directory, not the name of the file

Adding to a file

- Open with "a", rather than "r" or "w".
- Let's you add onto the end of the file, without having to copy the whole thing.

Traceback module

- When you make a mistake in Python, it shows you the whole “stack” of functions you are in.
- Not only what was the mistake, but how did you get to the line where the mistake occurred.
- The traceback module lets you show these error messages without crashing! Handy in an “except” construction.
- This is a fancy detail...

Traceback

```
Traceback (most recent call last):
  File "/Users/amenta/classes/ecs89h/hw2/stubhw2/harness.py", line 59, in <module>
    main()
  File "/Users/amenta/classes/ecs89h/hw2/stubhw2/harness.py", line 47, in main
    htmlList = application(environ,start_response)
  File "/Users/amenta/classes/ecs89h/hw2/stubhw2/hw2.py", line 7, in application
    pageObj = HTML_Page(environ) # make object
  File "/Users/amenta/classes/ecs89h/hw2/stubhw2/htmlClass.py", line 10, in __init__
    self.head = self.make_head()
  File "/Users/amenta/classes/ecs89h/hw2/stubhw2/htmlClass.py", line 35, in
    make_head
    int("cat")
ValueError: invalid literal for int() with base 10: 'cat'
```