

INTERNET PROGRAMMING IN PYTHON - WEEK 5

SMALL WEB FRAMEWORKS

A MOMENT TO REFLECT

```
[Fri Feb 04 13:34:12 2011] [error] [client 209.172.35.141] File does not exist: /var/www/phpMyAdmin-2.8.0-rc2
[Fri Feb 04 13:34:12 2011] [error] [client 209.172.35.141] File does not exist: /var/www/phpMyAdmin-2.8.0
[Fri Feb 04 13:34:13 2011] [error] [client 209.172.35.141] File does not exist: /var/www/phpMyAdmin-2.8.0.1
[Fri Feb 04 13:34:13 2011] [error] [client 209.172.35.141] File does not exist: /var/www/phpMyAdmin-2.8.0.2
[Fri Feb 04 13:34:13 2011] [error] [client 209.172.35.141] File does not exist: /var/www/phpMyAdmin-2.8.0.3
[Fri Feb 04 13:34:13 2011] [error] [client 209.172.35.141] File does not exist: /var/www/phpMyAdmin-2.8.0.4
[Fri Feb 04 13:34:13 2011] [error] [client 209.172.35.141] File does not exist: /var/www/phpMyAdmin-2.8.1-rc1
[Fri Feb 04 13:34:13 2011] [error] [client 209.172.35.141] File does not exist: /var/www/phpMyAdmin-2.8.1
[Fri Feb 04 13:34:13 2011] [error] [client 209.172.35.141] File does not exist: /var/www/phpMyAdmin-2.8.2
[Fri Feb 04 13:34:14 2011] [error] [client 209.172.35.141] File does not exist: /var/www/sqlmanager
[Fri Feb 04 13:34:14 2011] [error] [client 209.172.35.141] File does not exist: /var/www/mysqlmanager
[Fri Feb 04 13:34:14 2011] [error] [client 209.172.35.141] File does not exist: /var/www/p
[Fri Feb 04 13:34:14 2011] [error] [client 209.172.35.141] File does not exist: /var/www/PMA2005
[Fri Feb 04 13:34:14 2011] [error] [client 209.172.35.141] File does not exist: /var/www/pma2005
[Fri Feb 04 13:34:14 2011] [error] [client 209.172.35.141] File does not exist: /var/www/phpmanager
[Fri Feb 04 13:34:14 2011] [error] [client 209.172.35.141] File does not exist: /var/www/php-myadmin
[Fri Feb 04 13:34:15 2011] [error] [client 209.172.35.141] File does not exist: /var/www/phpmy-admin
[Fri Feb 04 13:34:15 2011] [error] [client 209.172.35.141] File does not exist: /var/www/webadmin
[Fri Feb 04 13:34:15 2011] [error] [client 209.172.35.141] File does not exist: /var/www/sqlweb
[Fri Feb 04 13:34:15 2011] [error] [client 209.172.35.141] File does not exist: /var/www/websql
[Fri Feb 04 13:34:15 2011] [error] [client 209.172.35.141] File does not exist: /var/www/webdb
[Fri Feb 04 13:34:15 2011] [error] [client 209.172.35.141] File does not exist: /var/www/mysqladmin
[Fri Feb 04 13:34:15 2011] [error] [client 209.172.35.141] File does not exist: /var/www/mysql-admin
[Fri Feb 04 13:34:16 2011] [error] [client 209.172.35.141] File does not exist: /var/www/pmy
[Fri Feb 04 13:34:16 2011] [error] [client 209.172.35.141] File does not exist: /var/www/published
[Fri Feb 04 13:34:16 2011] [error] [client 209.172.35.141] File does not exist: /var/www/wbsadmin
[Fri Feb 04 13:34:16 2011] [error] [client 209.172.35.141] File does not exist: /var/www/vhcs2
[Fri Feb 04 13:34:16 2011] [error] [client 209.172.35.141] File does not exist: /var/www/tools
[Fri Feb 04 13:34:16 2011] [error] [client 209.172.35.141] File does not exist: /var/www/phpadmin
[Fri Feb 04 13:34:16 2011] [error] [client 209.172.35.141] File does not exist: /var/www/phpMyAdmin-2.11.4
[Fri Feb 04 13:34:17 2011] [error] [client 209.172.35.141] File does not exist: /var/www/administrator
[Fri Feb 04 13:34:17 2011] [error] [client 209.172.35.141] File does not exist: /var/www/Admin
[Fri Feb 04 13:34:17 2011] [error] [client 209.172.35.141] File does not exist: /var/www/phpmyadmin_old
[Fri Feb 04 13:34:17 2011] [error] [client 209.172.35.141] File does not exist: /var/www/phpmyadmino-ld
[Fri Feb 04 13:34:17 2011] [error] [client 209.172.35.141] File does not exist: /var/www/test
[Fri Feb 04 13:34:17 2011] [error] [client 209.172.35.141] File does not exist: /var/www/my
[Sat Feb 05 07:16:11 2011] [error] [client 89.108.119.29] File does not exist: /var/www/
w00tw00t.at.blackhats.romanian.anti-sec:)
```

QUESTIONS AND REVIEW (20)

some thoughts from
the assignments

ASSIGNMENT

request:

<http://URL?a=123&b=321>

your response:

```
{  
    "result": 444,  
    "uwnetid": "dorseyb",  
    "time": 1296594367.0974219  
}
```

```
# must work. Verify with: check_assignment.py  
# must stay up for 24 hours
```

how'd it go?

blockers?

what was easier
than you expected?

- | | | |
|--------------------------|---|--|
| <input type="checkbox"/> |  UwipipBot |  Week 4 assignment: it worked! - He |
| <input type="checkbox"/> |  UwipipBot |  Week 4 assignment: success! - He |
| <input type="checkbox"/> |  UwipipBot |  Week 4 assignment: it worked! - He |
| <input type="checkbox"/> |  UwipipBot |  Week 4 assignment: it worked! - He |
| <input type="checkbox"/> |  UwipipBot |  Week 4 assignment: it worked! - He |
| <input type="checkbox"/> |  UwipipBot |  Week 4 assignment: success! - He |
| <input type="checkbox"/> |  UwipipBot |  Week 4 assignment: it worked! - He |
| <input type="checkbox"/> |  UwipipBot |  Week 4 assignment: success! - He |

http://208.85.148.131:8000/add/
http://208.85.148.174/myapp
http://208.85.148.196/cgi-bin/test2.py
http://ahosch-addition.appspot.com/
http://block115379-pwc.blueboxgrid.com:8000/cgi-bin/assignment4.py
http://block115384-uav.blueboxgrid.com/adder/adder
http://block115385-39q.blueboxgrid.com:8000/hw/
http://block115386-osv.blueboxgrid.com/cgi-bin/foo.py
http://block115387-yvm.blueboxgrid.com:8080/execute_py/jsonout
http://block115387-yvm.blueboxgrid.com:8081/execute_py/jsonout
http://block115389-wtj.blueboxgrid.com:8001/params.py
http://block115392-nzs.blueboxgrid.com:8080/
http://block115393-ake.blueboxgrid.com/add/
http://block115396-r76.blueboxgrid.com:8000
http://block115397-xwp.blueboxgrid.com/cgi-bin/adder-cgi-json.py
http://block115397-xwp.blueboxgrid.com/wsgiapp/wsgi-adder-json
http://block115400-msj.blueboxgrid.com:8008/
http://block115400-msj.blueboxgrid.com:8008/cgi-bin/cgiadd.py
http://block115402-4qe.blueboxgrid.com:8080
http://block115404-exb.blueboxgrid.com/cgi-bin/assignment4.py

video

permissions?

JSON vs python
(in printed output)

```
string = json.dumps(object)
```

```
object = json.loads(string)
```

what Content-type?

application/json

(except for javascript
callbacks: text/javascript)

in particular, see the comments:

<http://simonwillison.net/2009/Feb/6/json/>

CherryPy version

```
import cherrypy  
import json  
import time  
  
class Assignment4:  
    def index(self, a, b):  
        result = int(a) + int(b)  
        data = {'uninetid': 'dorseyb'}  
        data['time'] = time.time()  
        data['result'] = result  
        return json.dumps(data, indent=4)  
index.exposed = True  
  
cherrypy.quickstart(Assignment4())
```

*including
webserver!*

LECTURE A

(25)

remember the thirty minute
webserver & using
subprocess to call scripts?

easy to make dynamic
pages, just print headers
and body from a script

but...

“How do I get data
into the script?

CGI

environment variables

bash

```
$ export VARIABLE=some_value
$ echo this is the value: $VARIABLE
this is the value: some_value
```

python

```
$ export VARIABLE=some_value  
$ echo this is the value: $VARIABLE  
this is the value: some_value
```

```
$ python  
>>> import os  
>>> print os.environ['VARIABLE']  
some_value
```

changing environment variables

```
>>> print os.environ['VARIABLE']
some_value
>>> os.environ['VARIABLE']='new_value'
>>> print os.environ['VARIABLE']
new_value
>>> ^D
$ echo this is the value: $VARIABLE
this is the value: some_value
```

- parent processes don't see env changes
- subprocesses inherit env from parent
- you can set the subprocess env explicitly:

```
subprocess.Popen(args, bufsize=0, executable=None, stdin=None, stdout=None,  
stderr=None, preexec_fn=None, close_fds=False, shell=False, cwd=None, env=None,  
universal_newlines=False, startupinfo=None, creationflags=0)
```



CGI is mostly a set of
standard environment
variables

4.	The CGI Request	10
4.1.	Request Meta-Variables	10
4.1.1.	AUTH_TYPE	11
4.1.2.	CONTENT_LENGTH	12
4.1.3.	CONTENT_TYPE	12
4.1.4.	GATEWAY_INTERFACE	13
4.1.5.	PATH_INFO	13
4.1.6.	PATH_TRANSLATED	14
4.1.7.	QUERY_STRING	15
4.1.8.	REMOTE_ADDR	15
4.1.9.	REMOTE_HOST	16
4.1.10.	REMOTE_IDENT	16
4.1.11.	REMOTE_USER	16
4.1.12.	REQUEST_METHOD	17
4.1.13.	SCRIPT_NAME	17
4.1.14.	SERVER_NAME	17
4.1.15.	SERVER_PORT	18
4.1.16.	SERVER_PROTOCOL	18
4.1.17.	SERVER_SOFTWARE	19

oh, and fighting with
permissions

(and other fiddly bits)

in Python

(a hint)


```
import cgitb  
cgitb.enable()
```

<type 'exceptions.KeyError'>

Python 2.6.5: /usr/bin/python
Tue Feb 8 03:37:25 2011

A problem occurred in a Python script. Here is the sequence of function calls leading up to the error, in the order they occurred.

/home/dorseyb/uwpython_web/week05/cgi_example/cgi-bin/assignment4.py in ()

```
12
13 fs = cgi.FieldStorage()
14 a = fs['a']
15 b = fs['b']
16 a = int(a.value)
a undefined, fs = FieldStorage(None, None, [])
```

/usr/lib/python2.6/cgi.py in __getitem__(self=FieldStorage(None, None, []), key='a')

```
539         if item.name == key: found.append(item)
540     if not found:
541         raise KeyError, key
542     if len(found) == 1:
543         return found[0]
```

builtin KeyError = <type 'exceptions.KeyError'>, key = 'a'

<type 'exceptions.KeyError'>: 'a'

args = ('a',)

running CGI scripts

a few options:

CGIHTTPServer

Apache

CGIHTTPServer

```
# in your git checkout of uwpython_web
$ cd week05/cgi_example
$ pwd
<...>/uwpython_web/week05/cgi_example
$ cp some_cgi.py cgi-bin
$ python -m CGIHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

```
# then open http://localhost:8000/cgi-bin/some_cgi.py
```

setting up CGI scripts on
our VMs

```
$ cd /etc/apache2
```

```
# this file contains the active configuration  
# for the default web site (the only one we have)  
$ less sites-available/default
```



```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost

    DocumentRoot /var/www
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/>
        Options Indexes FollowSymLinks Multiviews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>
```

```
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory "/usr/lib/cgi-bin">
    AllowOverride None
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
    Order allow,deny
    Allow from all
</Directory>
```

```
ErrorLog /var/log/apache2/error.log
```

```
# Possible values include: debug, info, notice,  
# warn, error, crit, alert, emerg.  
LogLevel warn
```

```
CustomLog /var/log/apache2/access.log combined
```

```
Alias /doc/ "/usr/share/doc/"  
<Directory "/usr/share/doc/">  
    Options Indexes MultiViews FollowSymLinks  
    AllowOverride None  
    Order deny,allow  
    Deny from all  
    Allow from 127.0.0.0/255.0.0.0 ::1/128  
</Directory>  
  
</VirtualHost>
```

```
$ ls -al /usr/lib/cgi-bin/
total 28
drwxr-xr-x  2 root root  4096 Apr 13  2010 .
drwxr-xr-x 70 root root 24576 Feb  5 06:07 ..
```



block115379-pwc.blueboxgrid.com/cgi-bin/

Forbidden

You don't have permission to access /cgi-bin/ on this server.

Apache/2.2.14 (Ubuntu) Server at block115379-pwc.blueboxgrid.com Port 80

```
$ sudo cp ~/uwpython_web/week05/cgi_test.py /usr/lib/cgi-bin/  
$ sudo chown www-data:www-data /usr/lib/cgi-bin/cgi_test.py
```

```
$ ls -al /usr/lib/cgi-bin/  
total 32  
drwxr-xr-x  2 root root      4096 Feb  6 18:14 .  
drwxr-xr-x 70 root root     24576 Feb  5 06:07 ..  
-rwxr-xr-x  1 www-data www-data   40 Feb  6 18:14 cgi_test.py
```

```
$ cat /usr/lib/cgi-bin/cgi_test.py  
#!/usr/bin/python  
import cgi  
cgi.test()
```

The screenshot shows a web browser window with the URL `block115379-pwc.blueboxgrid.com/cgi-bin/cgi_test.py` in the address bar. The page content is displayed below:

Command Line Arguments:

```
['/usr/lib/cgi-bin/cgi_test.py']
```

Form Contents:

No form fields.

Shell Environment:

```
DOCUMENT_ROOT  
/var/www  
GATEWAY_INTERFACE  
CGI/1.1
```

```
$ sudo tail /var/log/apache2/access.log
173.14.243.86 - - [06/Feb/2011:18:12:16 +0000] "GET /cgi-bin/ HTTP/1.1"
403 519 "-" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_6; en-US)
AppleWebKit/534.10 (KHTML, like Gecko) Chrome/8.0.552.237 Safari/534.10"
173.14.243.86 - - [06/Feb/2011:18:12:16 +0000] "GET /favicon.ico HTTP/
1.1" 404 518 "-" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_6; en-
US) AppleWebKit/534.10 (KHTML, like Gecko) Chrome/8.0.552.237 Safari/
534.10"
173.14.243.86 - - [06/Feb/2011:18:16:13 +0000] "GET /cgi-bin/cgi_test.py
HTTP/1.1" 200 1547 "-" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X
10_6_6; en-US) AppleWebKit/534.10 (KHTML, like Gecko) Chrome/8.0.552.237
Safari/534.10"
173.14.243.86 - - [06/Feb/2011:18:16:14 +0000] "GET /favicon.ico HTTP/
1.1" 404 518 "-" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_6; en-
US) AppleWebKit/534.10 (KHTML, like Gecko) Chrome/8.0.552.237 Safari/
534.10"

$ sudo tail /var/log/apache2/error.log
[Sun Feb 06 18:12:13 2011] [error] [client 173.14.243.86] File does not
exist: /var/www/cgi-bin
[Sun Feb 06 18:12:16 2011] [error] [client 173.14.243.86] attempt to
invoke directory as script: /usr/lib/cgi-bin/
```

LAB A

(20)

remember Jeff's PHP
troubleshooting script?

This is the title

This is some prose

This is a paragraph

The server IP address is 67.214.214.142:80 and you are coming from 173.14.243.86:50162
hostname (If empty, that is a DNS problem).

Your browser user agent is Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_6; en-US)
AppleWebKit/534.10 (KHTML, like Gecko) Chrome/8.0.552.237 Safari/534.10.

The X-Forwarded-For header is

Virtual Server block115398-mzu.blueboxgrid.com The server name is block115398-
mzu.blueboxgrid.com (if an IP address, then a DNS problem)

Your browser specified the keep-alive option

The cookie jeffs_cookie is set to Test value Sun, 06 Feb 11 16:48:03 +0000!

The currently executing script is /index.php

The request arrived at Monday 07-Feb-11 14:32:16 UTC

we're going to port
it to Python

```
git clone git://github.com/briandorsey/uwpython_web.git  
(or, if you already have a copy: git pull)
```

LAB A

- setup **lab_a_cgi.py** on a VM
(copy to **/usr/lib/cgi-bin/**)
- update **lab_a_cgi.py** to fill in the template with real values from the CGI environment and cookies

BREAK (10)

API TALKS

(25)

talks

GUEST SPEAKER (20)

a quick survey of templates (for web sites)

warning:

I'm not an expert in
templates.

and you don't
need to be either

I used to have all sorts of
strong opinions about
templates...

they're like mini
programming languages

one trick,
if you just need something
quick, in a script

```
>>> d = {'key' : 'value'}  
>>> message = """  
... Hello,  
...     This is my key: %(key)s  
...     """  
...  
>>> print message % d
```

```
Hello,  
This is my key: value
```

and if you're worried about `KeyErrors`, check out `collections.defaultdict()`



please don't write
your own

<- list of templating
modules

two main styles

blocks of code(ish)

Jinja 2

```
{% extends "layout.html" %}  
{% block body %}  
  <ul>  
    {% for user in users %}  
      <li><a href="{{ user.url }}">{{ user.username }}  
    </a></li>  
    {% endfor %}  
  </ul>  
{% endblock %}
```

attributes embedded in XML

Genshi

```
<?python
    title = "A Genshi Template"
    fruits = ["apple", "orange", "kiwi"]
?>
<html xmlns:py="http://genshi.edgewall.org/">
    <head>
        <title py:content="title">This is replaced.</title>
    </head>
    <body>
        <p>These are some of my favorite fruits:</p>
        <ul>
            <li py:for="fruit in fruits">
                I like ${fruit}s
            </li>
        </ul>
    </body>
</html>
```

recommendations:

in Django apps,
just use Django templates

otherwise,
look over Jinja and Genshi
and pick the one which is
more visually appealing

questions?

BREAK
(10)

RANDOM

LECTURE B

(25)

WSGI

actually there are a bunch of things basically like WSGI:
mod_python, FastCGI,
SCGI, etc

they all keep Python (or other language) processes running, and have some way for the webserver to send requests and get responses.

why bother?

```
ab -n 3000 -c 10 -k -d http://block115379-  
pwc.blueboxgrid.com/cgi-bin/lab a cgi template.py
```

```
ab -n 3000 -c 10 -k -d http://block115379-  
pwc.blueboxgrid.com/wsgi_lab_b
```

WSGI seems to be what
most people are using for
python hosting

so that's what we're going to
talk about

if you search for WSGI,
you'll find all kinds of
controversy about how it
should be adapted for
Python 3

we're going to ignore it

if you want to use Python 3
on the web...

please wait a bit

<http://lucumr.pocoo.org/2011/1/22/python-the-web-and-little-things/>

“All you really need to know...
Unless you are developing a new Web app
framework (admittedly, $p > 0.5...$) you don't
care about WSGI.

Use a Web app framework.

They all support WSGI at this point, which
means you basically need to worry about
configuration and deployment of your app,
and nothing else.”

--Titus Brown

I'm still going to
talk about it a bit

WSGI servers

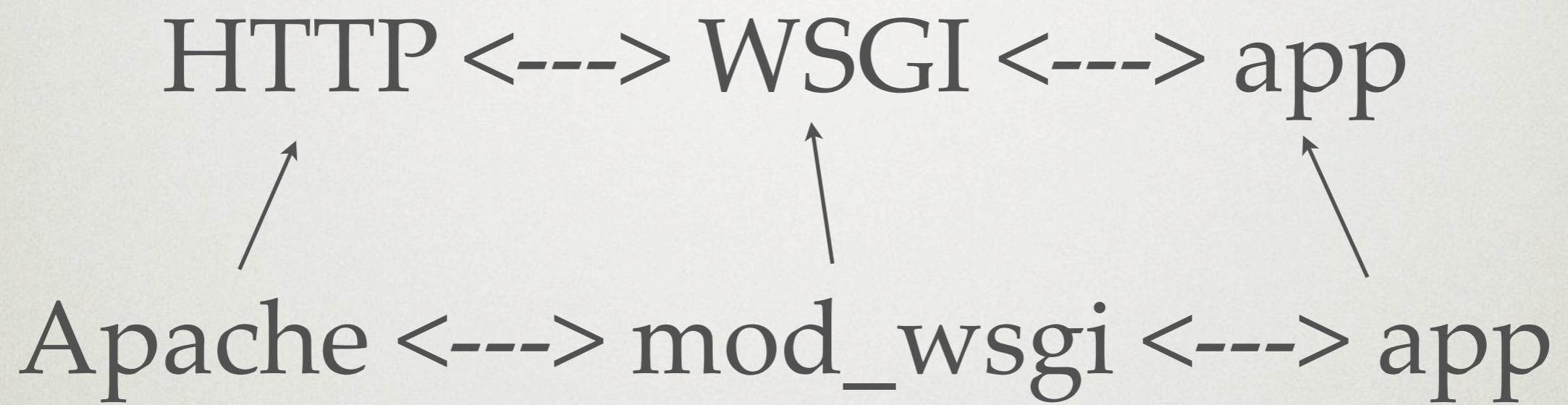
HTTP <----> WSGI

WSGI applications

WSGI <----> app code

HTTP <----> WSGI <----> app





middleware

WSGI <----> mw <----> WSGI

HTTP <---> WSGI <---> mw <---> WSGI <---> app

WSGI server (simplified)

```
from some_application import simple_app

def handle_request(app):
    iterable = app(environ, start_response)
    for data in iterable:
        # send data to client

def start_response(status, headers):
    # send status and headers to client

# listen for HTTP and pass to handle_request()
serve(simple_app)
```

Look familiar?

`REQUEST_METHOD`

The HTTP request method, such as "GET" or "POST". This cannot ever be an empty string, and so is always required.

`SCRIPT_NAME`

The initial portion of the request URL's "path" that corresponds to the application object, so that the application knows its virtual "location". This **may** be an empty string, if the application corresponds to the "root" of the server.

`PATH_INFO`

The remainder of the request URL's "path", designating the virtual "location" of the request's target within the application. This **may** be an empty string, if the request URL targets the application root and does not have a trailing slash.

`QUERY_STRING`

The portion of the request URL that follows the "?", if any. May be empty or absent.

`CONTENT_TYPE`

The contents of any Content-Type fields in the HTTP request. May be empty or absent.

`CONTENT_LENGTH`

The contents of any Content-Length fields in the HTTP request. May be empty or absent.

`SERVER_NAME`, `SERVER_PORT`

When combined with `SCRIPT_NAME` and `PATH_INFO`, these variables

WSGI application

```
def simple_app(environ, start_response):
    status = '200 OK'
    response_headers = [('Content-type','text/plain')]
    start_response(status, response_headers)
    return ['Hello world!\n']
```

WSGI server w/ middleware

```
from some_application import simple_app

def handle_request(app):    <...>
def start_response(status, headers):    <...>

class Upperware:
    def __init__(self, app):
        self.wrapped_app = app

    def __call__(self, environ, start_response):
        for data in self.wrapped_app(environ, start_response):
            return data.upper()

wrapped_simple_app = Upperware(simple_app)
serve(wrapped_simple_app)
```

middleware should be as
transparent as possible

test: without it, does your
app break?

some interesting
middleware

- AuthKit - <http://authkit.org>
- Sophisticated and extensible permissions system
- Built in support for HTTP basic, HTTP digest, form, cookie and OpenID authentication methods plus others
- Easily define users, passwords and roles
- Designed to be totally extensible so you can use the components to integrate with a database, LDAP connection or your own custom system

- Werkzeug - <http://werkzeug.pocoo.org/>
- HTTP header parsing and dumping
- Easy to use request and response objects
- **Interactive JavaScript based in-browser debugger**
- Supports Python from 2.4 up to 2.7. If you are interested in Python 3 support, please contact us.
- Unicode support
- Basic session and signed cookie support
- URI and IRI utilities with unicode awareness
- builtin library of fixes for buggy WSGI servers and browsers
- **integrated routing system for matching URLs to endpoints and vice versa**

lots more:

http://wsgi.org/wsgi/Middleware_and_Utils

setting up WSGI scripts on
our VMs

```
$ sudo apt-get install libapache2-mod-wsgi
```

```
$ cd /etc/apache2
```

```
$ ls mods-enabled/
```

```
<...>
```

```
wsgi.conf
```

```
wsgi.load
```

```
$ cd /usr/local/
$ sudo mkdir wsgi-scripts
$ sudo cp ~/uwpython_web/week05/wsgi_test.py wsgi-scripts/
$ ls -al wsgi-scripts/
total 12
drwxr-sr-x 2 root staff 4096 Feb  6 19:12 .
drwxrwsr-x 3 root staff 4096 Feb  6 19:11 ..
-rwxr-xr-x 1 root staff 1303 Feb  6 19:12 wsgi_test.py
```

```
$ sudo $EDITOR /etc/apache2/sites-available/default  
  
# add line:  
WSGIScriptAlias /wsgi_test /var/local/wsgi-scripts/wsgi_test.py
```



```
$ sudo /etc/init.d/apache2 restart
 * Restarting web server
apache2
... waiting
```



LAB B

(20)

```
git clone git://github.com/briandorsey/uwpython_web.git  
(or, if you already have a copy: git pull)
```

LAB B

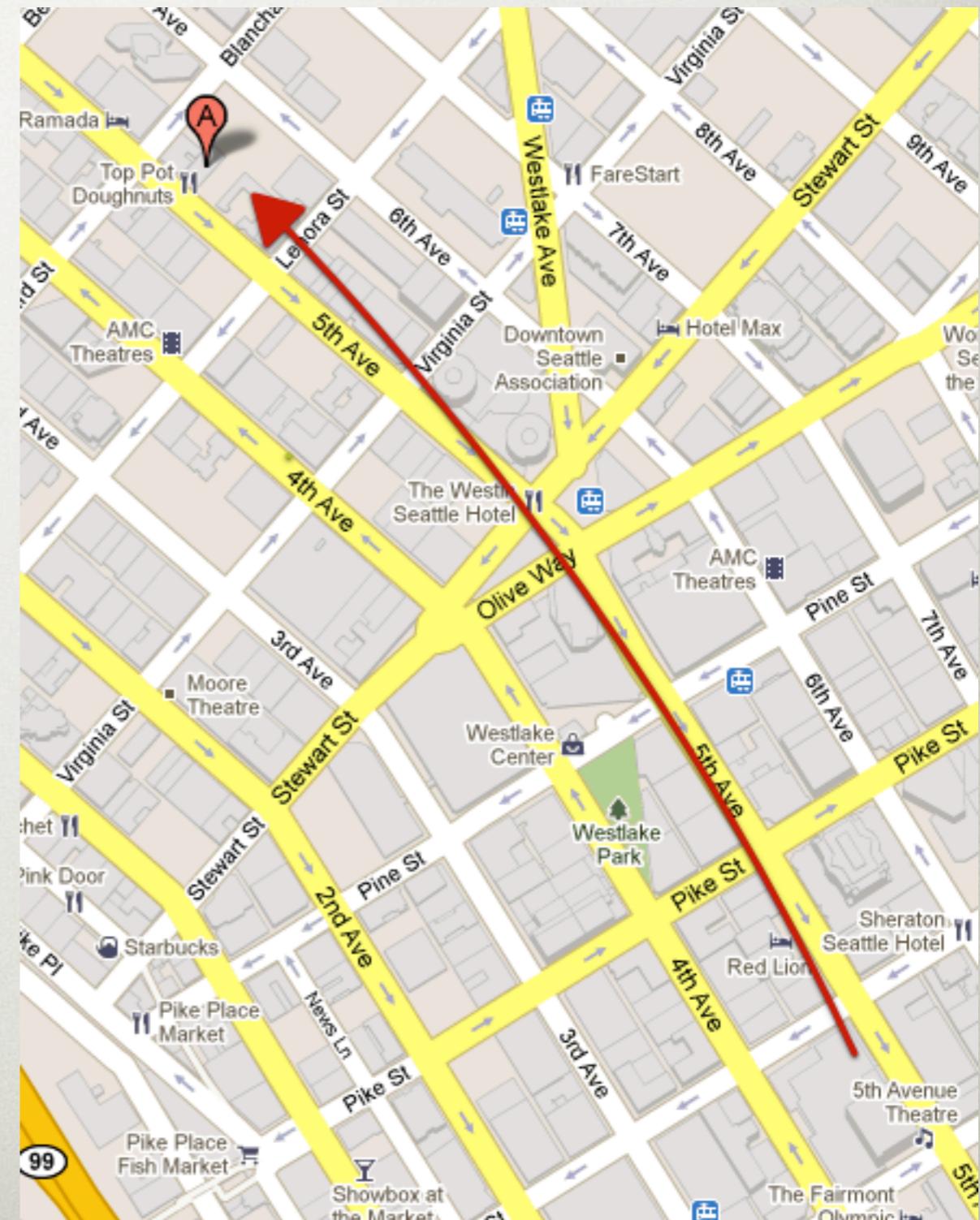
- setup **lab_a_wsgi.py** on a VM
- update **lab_a_wsgi.py** to fill in the template with real values from the WSGI environment and cookies

WRAPUP

INFO

Office hours:
Sunday 2-5pm

Top Pot Donuts
2124 5th Ave
Seattle, WA 98121
206-728-1966



turn in here: http://bit.ly/uwipip_week5

ASSIGNMENT

- write a multi page website, using **bookdb.py** (see **uwpython_web** git repo)
- index page is a list of books, link to a detail page per book
- each detail page displays info about one book
- use any Python tech you'd like
- it's probably easier if you do the week 6 readings, *then* this assignment
- fill out feedback form:
http://bit.ly/uwipip_midquarter