02820 – Python Programming

Text and Web mining and Python

CGI, urllib, urlparse, urljoin

Bartłomiej Wilkowski

DTU Informatics
B321/117

bw@imm.dtu.dk

CGI module

- CGI (Common Gateway Interface)
- standard protocol for interfacing external application software with an information server, commonly a web server
- In Python: import cgi

Python CGI example

EXAMPLE WEB PAGE WITH A FORM

```
<HTML>
<HEAD>
<TITLE> Hello World</TITLE>
</HEAD>
<BODY>
<H1>Greetings</H1>
<form action="http://neuroinf.imm.dtu.dk/cgi-bin/02820example_CGI.py" method="get">

<input type="radio" name="drink" value="tea" checked > Tea <br>
<input type="radio" name="drink" value="coffee" > Coffee <br>
<input type="radio" name="drink" value="hot chocolate" > Hot Chocolate <p>

<input type="submit" value="Place order">
</form>
</BODY>
</HTML>
```

Python CGI example

```
import cgi
print "Content-Type: text/html\n"
form = cgi.FieldStorage()

drink = form.getvalue("drink")

print """
<html>
<head> <title>What would you like to drink</title> </head>
<body>
<h4>Your drink: </h4><p>
"""
if drink == "tea":
    print "You requested tea."
elif drink == "coffee":
    print "You requested coffee."
elif drink == "hot chocolate":
    print "You requested hot chocolate."
else:
    print "You need to select a drink!"

print """
<p>Thank you for your visit. Please come again. <p>
</body></html>
"""
```

Python CGI example

Your drink:

You requested tea.

Thank you for your visit. Please come again.

urllib module

- High-level interface for fetching data across the World Wide Web.
- In Python: import urllib

```
import urllib

params = urllib.urlencode({'drink': 'tea'})

# GET
f = urllib.urlopen("http://neuroinf.imm.dtu.dk/cgi-bin/02820example_CGI.py?%s" % params)

# POST
# f = urllib.urlopen("http://neuroinf.imm.dtu.dk/cgi-bin/02820example_CGI.py", params)

print f.read()
```

urllib & proxies

```
proxies = {'http': 'http://proxy.site.com:8080/'}

#defined proxies
opener = urllib.FancyURLopener(proxies)

# no proxies at all
opener = urllib.FancyURLopener({})

#proxies from environment
opener = urllib.FancyURLopener(None)

f = opener.open("http://www.python.org")

f.read()
```

urllib – other methods

```
urllib.urlretrieve("
http://www.kapelanawesela.pl/muzyka/Chiwawa.mp3",
"/home/bw/Desktop/retrieved.mp3")
```

```
web = urllib.quote('http://www.imm.dtu.dk/~bw')
```

```
[web = 'http://www.imm.dtu.dk/%7ebw']
```

```
urllib.unquote(string)
```

urlparse module

- Standard interface to break Uniform Resource Locator (URL) strings up in components (addressing scheme, network location, path etc.)
- In Python: import urlparse

```
urlparse.urlparse('http://neuroinf.imm.dtu.dk/cgi-bin/02820example_CGI.py?drink=tea')
```

Out:

```
ParseResult(scheme='http', netloc='neuroinf.imm.dtu.dk',
path='/cgi-bin/02820example_CGI.py', params='', query='drink=tea', fragment='')
```

urljoin

- Construct a full (“absolute”) URL by combining a “base URL” (base) with another URL (url)
- Informally, this uses components of the base URL, in particular the addressing scheme, the network location and (part of) the path, to provide missing components in the relative URL

```
from urlparse import urljoin

urljoin('http://www.imm.dtu.dk/%7Ebw/index.html', 'mysite.html')
OUT:
'http://www.imm.dtu.dk/%7Ebw/mysite.html'
```