

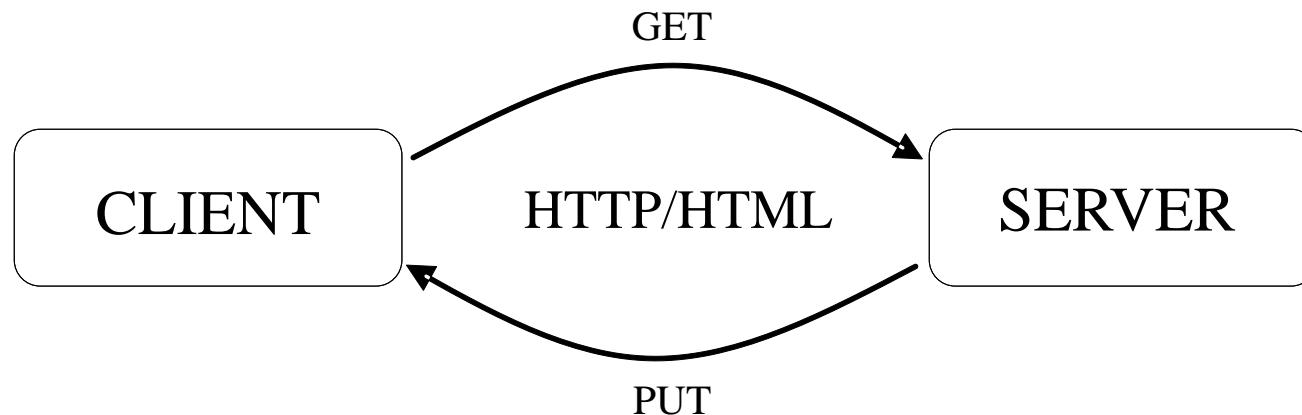
Extend the Server's Functionality

Python CGI Programming

Emmanuel Stefanakis
estef@unb.ca

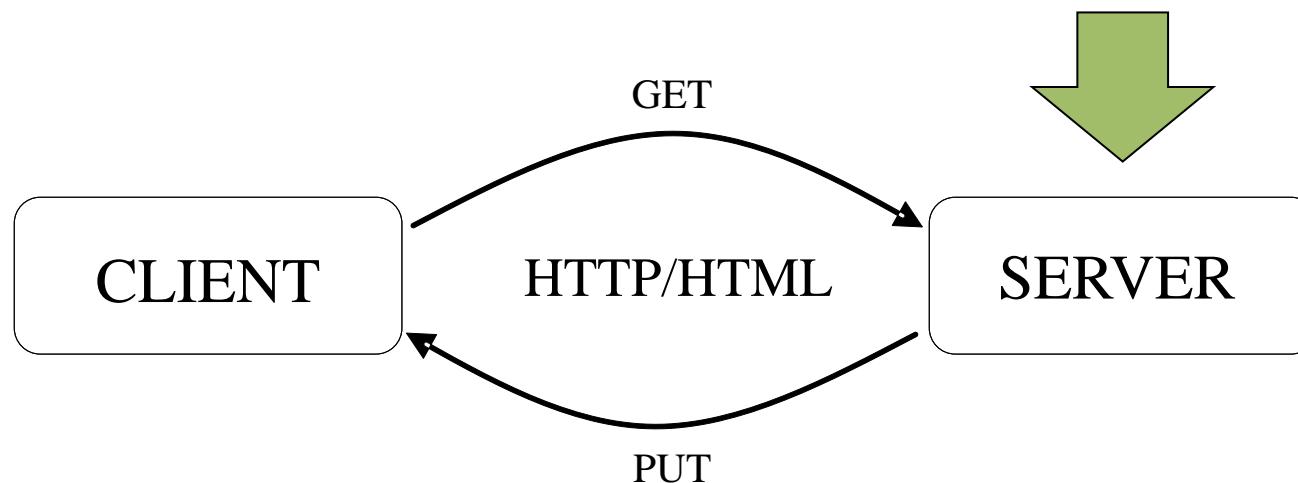
Interactive Maps...

- Enriched functionality by extending...
 - the **client side** functionality
 - the **server side** functionality



Interactive Maps...

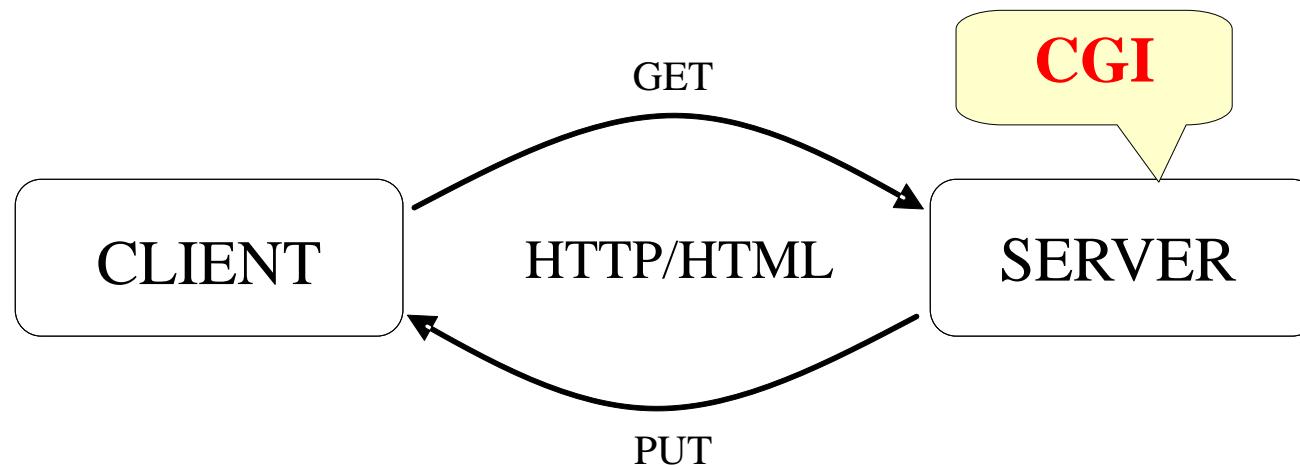
- Enriched functionality by extending...
 - the **client side** functionality
 - the **server side** functionality



Extend the Server-Side

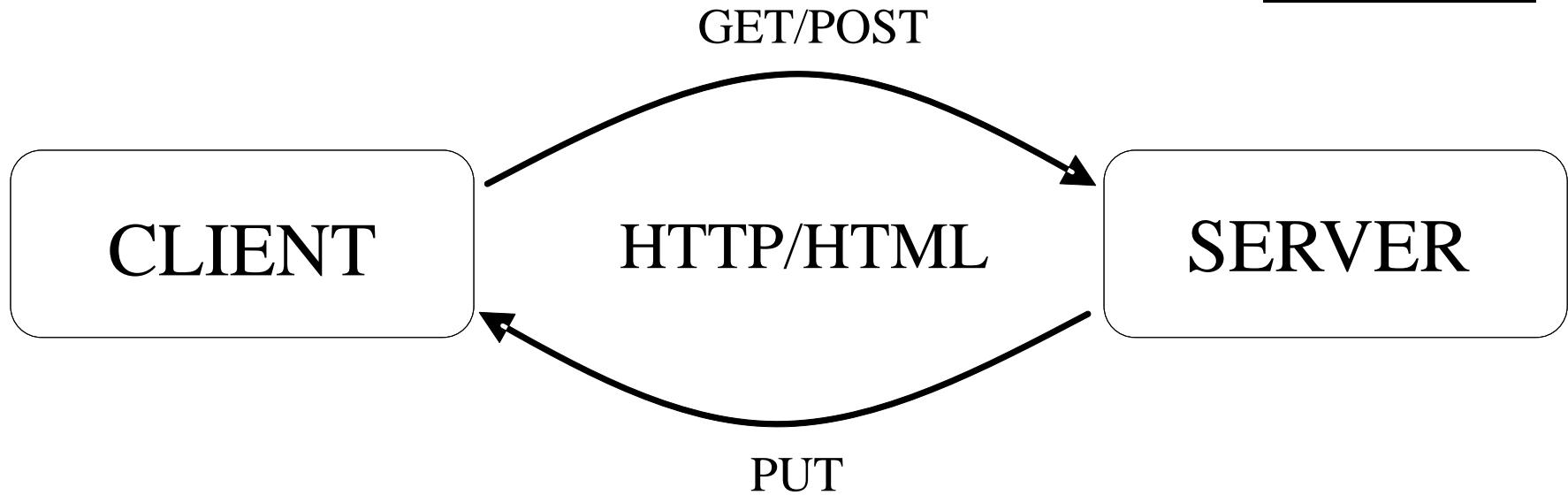
- Common Gateway Interface (CGI)

e.g., `http://server?doThis¶meter1¶meter2`



The Web...

- Communication of computers
 - Client-Server Architecture....



CGI

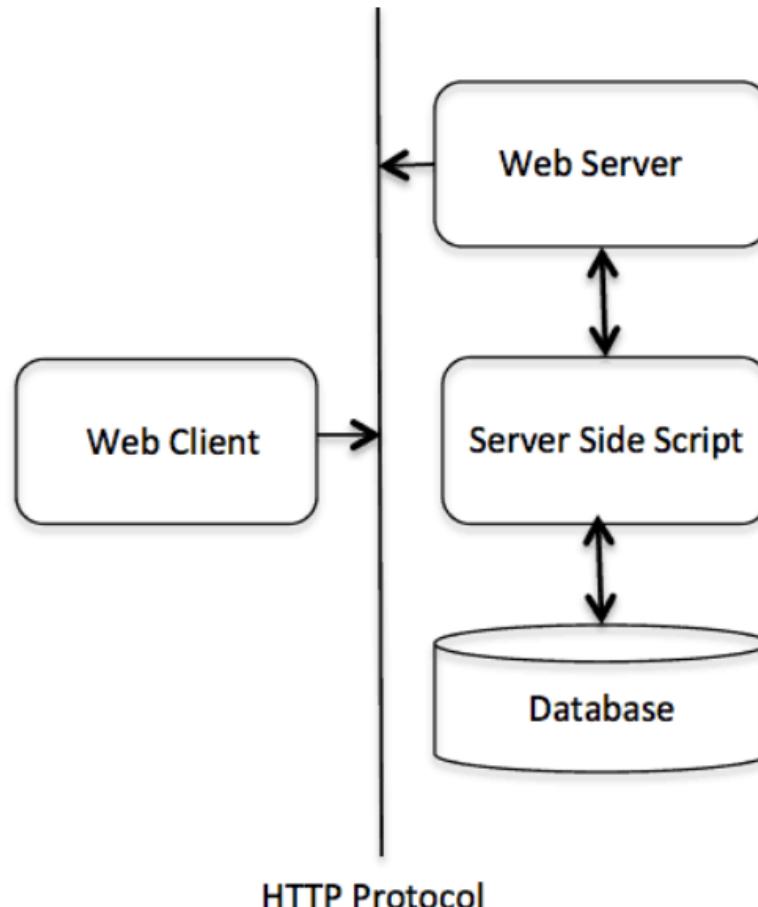
The Common Gateway Interface, or CGI, is a set of standards that define how information is exchanged between the web server and a custom script. The CGI specs are currently maintained by the NCSA.

What is CGI?

- The Common Gateway Interface, or CGI, is a standard for external gateway programs to interface with information servers such as HTTP servers.
- The current version is CGI/1.1 and CGI/1.2 is under progress.

CGI

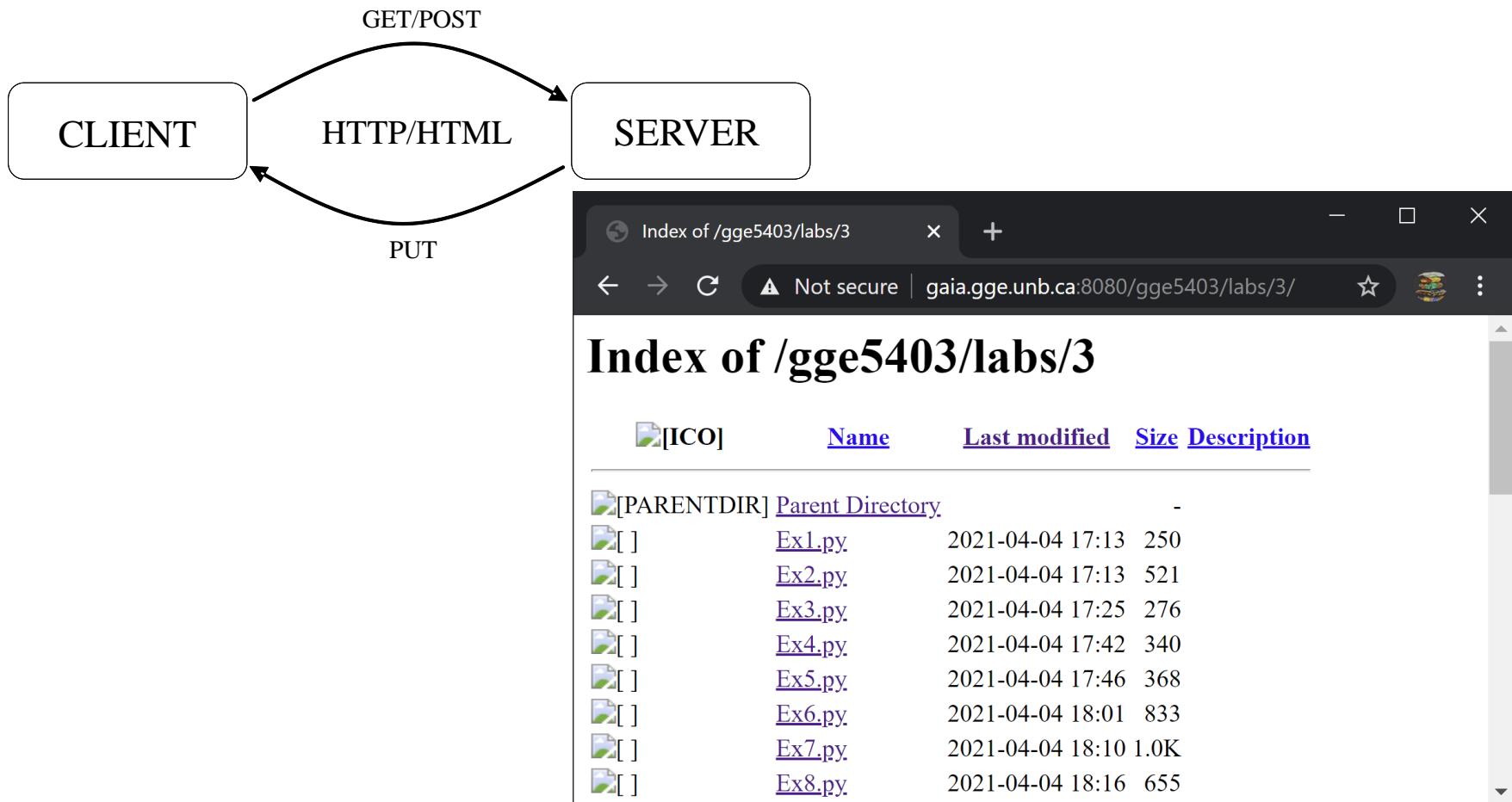
CGI Architecture Diagram



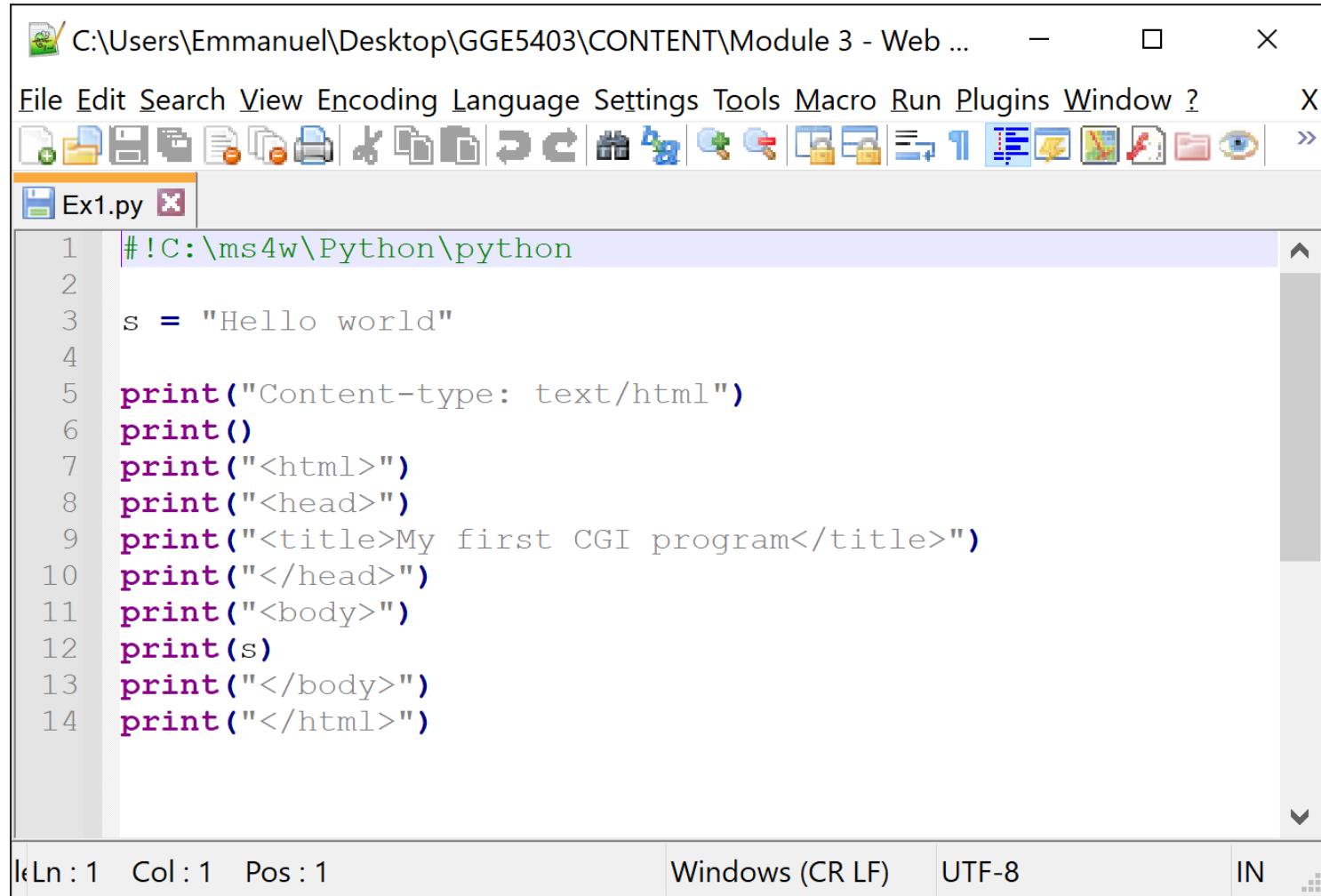
Gaia Server



Examples...



<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Ex1.py>



The screenshot shows a Python code editor window titled "Ex1.py". The code is a CGI program that prints an HTML response. It includes imports, a variable assignment, and several print statements to output the HTML structure.

```
# !C:\ms4w\Python\python
s = "Hello world"
print("Content-type: text/html")
print()
print("<html>")
print("<head>")
print("<title>My first CGI program</title>")
print("</head>")
print("<body>")
print(s)
print("</body>")
print("</html>")
```

At the bottom of the editor, status bars show "Ln : 1 Col : 1 Pos : 1", "Windows (CR LF)", "UTF-8", and "IN".

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Ex1.py>

The screenshot shows a Windows desktop environment. In the foreground, there is a code editor window titled "Ex1.py". The code in the editor is:

```
1 # !C:\ms4w\Python\python
2
3 s = "Hello world"
4
5 print("Content-type: text/html")
6 print()
7 print("<html>")
8 print("<head>")
9 print("<title>My first CGI program</title>")
10 print("</head>")
11 print("<body>")
12 print(s)
13 print("</body>")
14 print("</html>")
```

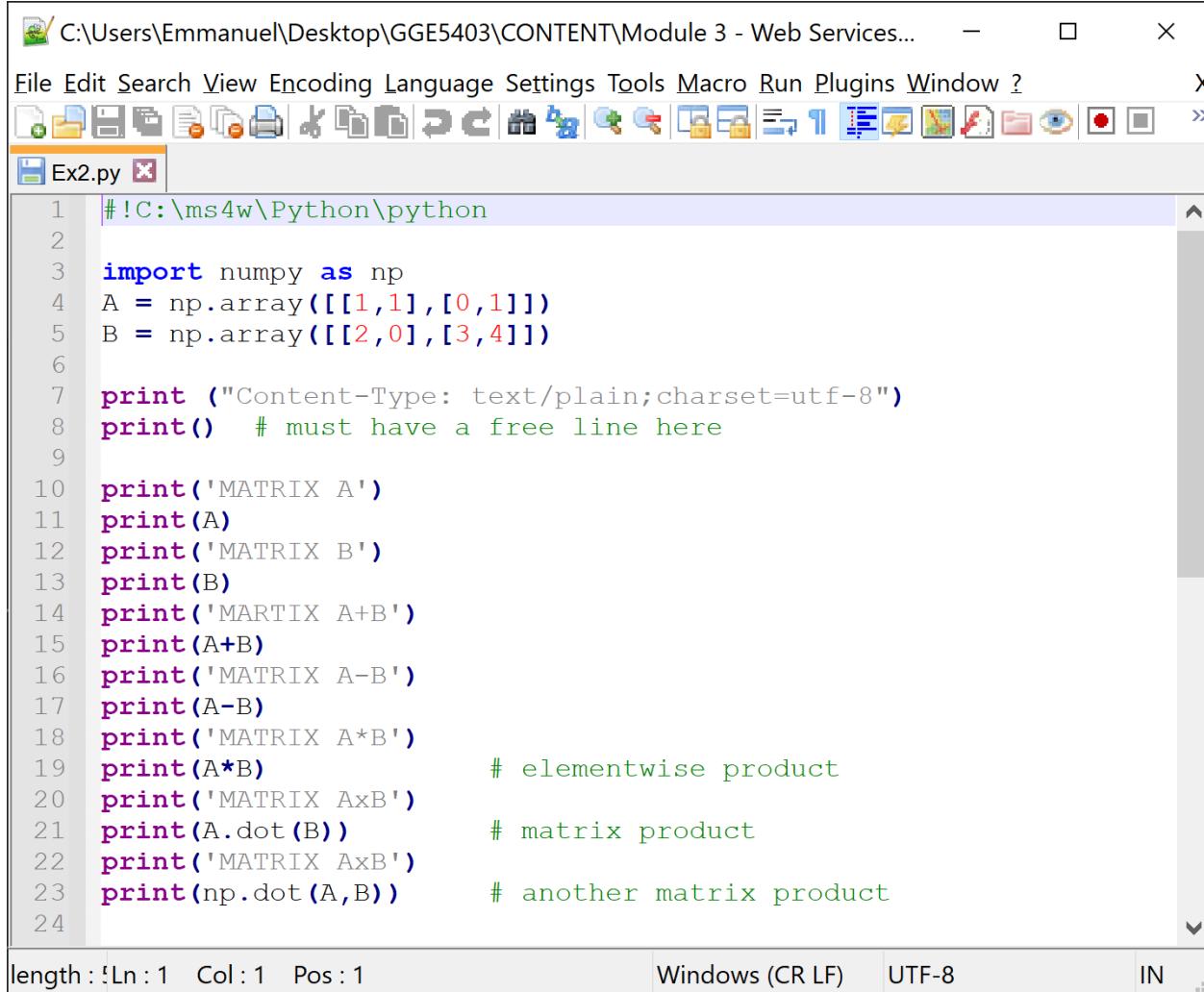
Below the code editor, the status bar displays "Ln : 1 Col : 1 Pos : 1" and "Windows". To the right of the code editor, a browser window titled "view-source:gaia.gge.unb.ca:8080/gge5403/labs/3/Ex1.py" shows the raw HTML source code of the CGI program. The browser window below it, titled "My first CGI program", displays the output "Hello world". Both browser windows have a warning icon indicating they are "Not secure".

NumPy library

- NumPy is an open source project aiming to enable numerical computing with Python.
- Available on gaia server



<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Ex2.py>



```
C:\Users\Emmanuel\Desktop\GGE5403\CONTENT\Module 3 - Web Services... - X
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Ex2.py X
1 # !C:/ms4w/Python/python
2
3 import numpy as np
4 A = np.array([[1,1],[0,1]])
5 B = np.array([[2,0],[3,4]])
6
7 print ("Content-Type: text/plain;charset=utf-8")
8 print() # must have a free line here
9
10 print('MATRIX A')
11 print(A)
12 print('MATRIX B')
13 print(B)
14 print('MARTIX A+B')
15 print(A+B)
16 print('MATRIX A-B')
17 print(A-B)
18 print('MATRIX A*B')
19 print(A*B) # elementwise product
20 print('MATRIX AxB')
21 print(A.dot(B)) # matrix product
22 print('MATRIX AxB')
23 print(np.dot(A,B)) # another matrix product
24

length : !Ln : 1 Col : 1 Pos : 1 Windows (CR LF) UTF-8 IN
```

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Ex2.py>

The screenshot shows a Python IDE interface on the left and a web browser window on the right.

IDE (Left):

- Title bar: C:\Users\Emmanuel\Desktop\GGE5403\CONTENT\Module 3 - Web Services...
- Toolbar icons: File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, Help.
- Code editor tab: Ex2.py
- Code content:

```
1 # !C:/ms4w/Python/python
2
3 import numpy as np
4 A = np.array([[1,1],[0,1]])
5 B = np.array([[2,0],[3,4]])
6
7 print ("Content-Type: text/plain;charset=utf-8")
8 print() # must have a free line here
9
10 print('MATRIX A')
11 print(A)
12 print('MATRIX B')
13 print(B)
14 print('MARTIX A+B')
15 print(A+B)
16 print('MATRIX A-B')
17 print(A-B)
18 print('MATRIX A*B')
19 print(A*B) # elementwise product
20 print('MATRIX AxB')
21 print(A.dot(B)) # matrix product
22 print('MATRIX AxB')
23 print(np.dot(A,B)) # another matrix product
24
```

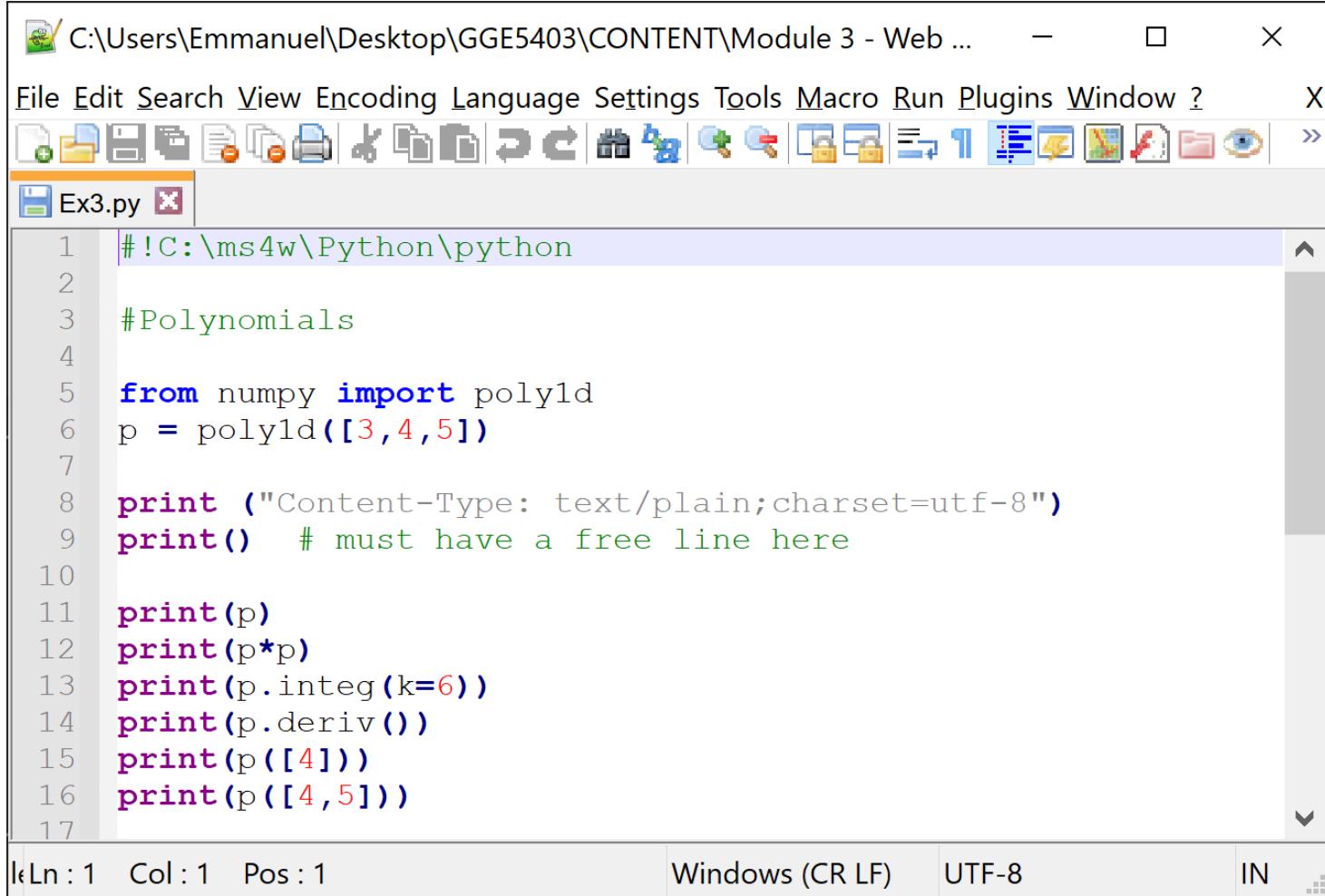
- Status bar: length : 1 Ln : 1 Col : 1 Pos : 1

Browser (Right):

- Title bar: gaia.gge.unb.ca:8080/gge5403/l...
- Address bar: Not secure | gaia.gge.unb.ca:8080/gge5403/labs/3/Ex2.py
- Output content:

```
MATRIX A
[[1 1]
 [0 1]]
MATRIX B
[[2 0]
 [3 4]]
MARTIX A+B
[[3 1]
 [3 5]]
MATRIX A-B
[[-1 1]
 [-3 -3]]
MATRIX A*B
[[2 0]
 [0 4]]
MATRIX Ax B
[[5 4]
 [3 4]]
MATRIX Ax B
[[5 4]
 [3 4]]
```

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Ex3.py>



The screenshot shows a Python code editor window with the following details:

- Title Bar:** C:\Users\Emmanuel\Desktop\GGE5403\CONTENT\Module 3 - Web ...
- Menu Bar:** File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, ?
- Toolbar:** Standard file operations like Open, Save, Print, Copy, Paste, Find, etc.
- Code Editor:** A tab labeled "Ex3.py" is active. The code content is as follows:

```
1 # !C:/ms4w/Python/python
2
3 #Polynomials
4
5 from numpy import poly1d
6 p = poly1d([3,4,5])
7
8 print ("Content-Type: text/plain;charset=utf-8")
9 print() # must have a free line here
10
11 print(p)
12 print(p*p)
13 print(p.integ(k=6))
14 print(p.deriv())
15 print(p([4]))
16 print(p([4,5]))
```
- Status Bar:** Ln : 1 Col : 1 Pos : 1, Windows (CR LF), UTF-8, IN

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Ex3.py>

The screenshot shows a Python IDE interface with a code editor and a browser window.

Code Editor:

```
1 #!C:\ms4w\Python\python
2
3 #Polynomials
4
5 from numpy import poly1d
6 p = poly1d([3,4,5])
7
8 print ("Content-Type: text/plain; charset=utf-8")
9 print() # must have a free line here
10
11 print(p)
12 print(p*p)
13 print(p.integ(k=6))
14 print(p.deriv())
15 print(p([4]))
16 print(p([4,5]))
```

Browser Window:

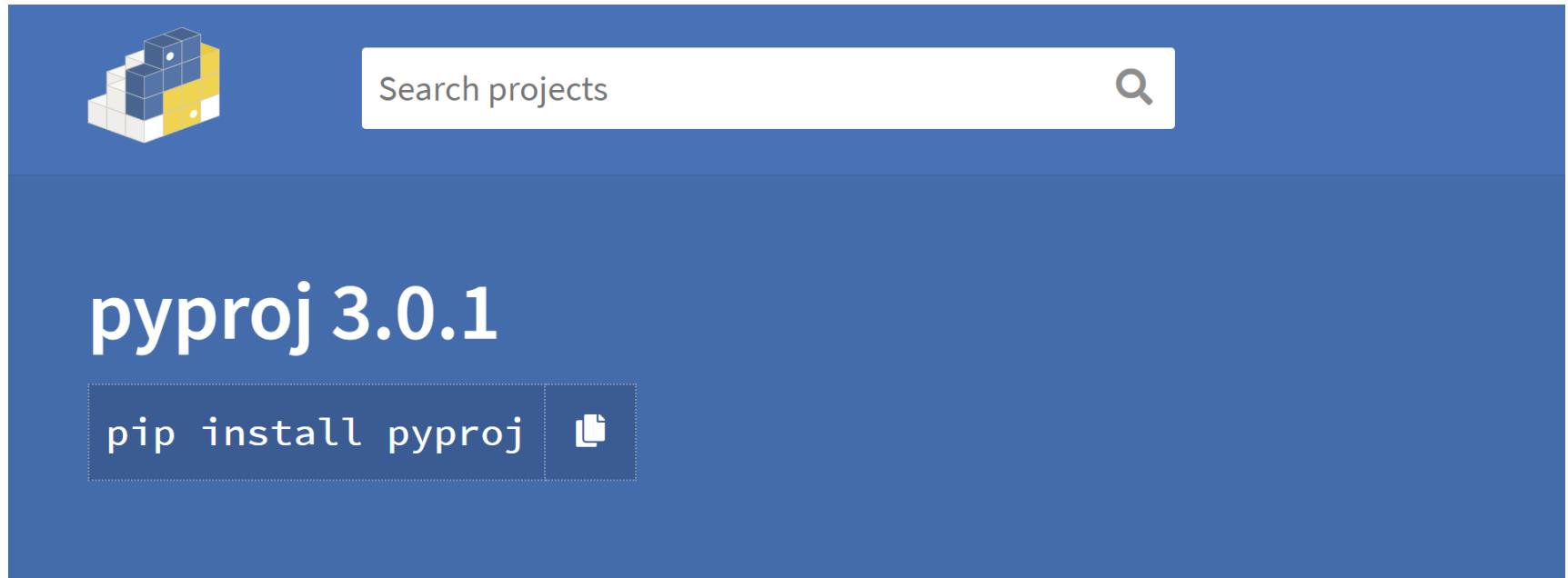
gaia.gge.unb.ca:8080/gge5403/labs/3/Ex3.py

Not secure | gaia.gge.unb.ca:8080/gge5403/labs/3/Ex3.py

```
2
3 x + 4 x + 5
4      3      2
9 x + 24 x + 46 x + 40 x + 25
3      2
1 x + 2 x + 5 x + 6

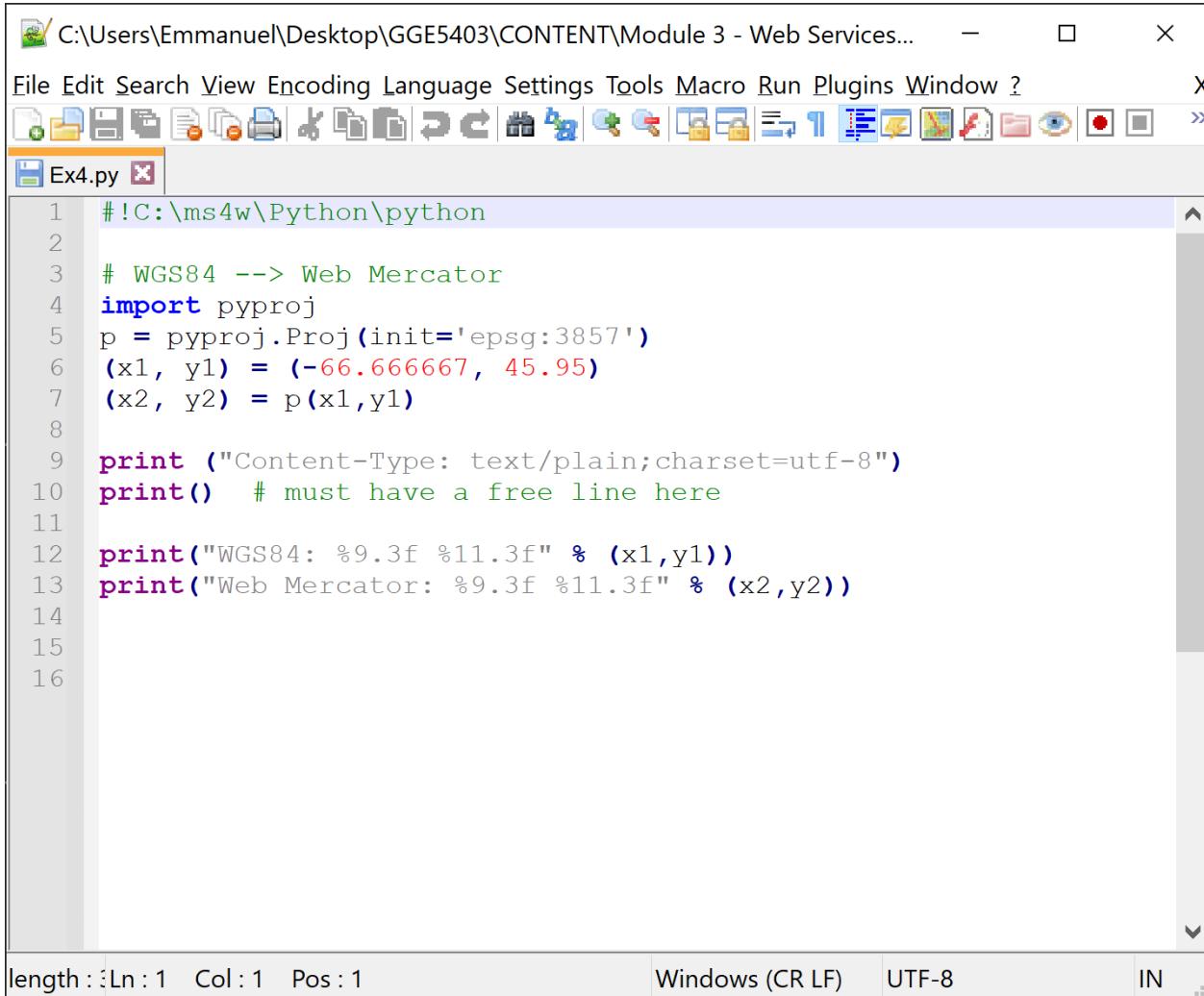
6 x + 4
[69]
[ 69 100]
```

pyproj library



Python interface to PROJ (cartographic projections and coordinate transformations library)

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Ex4.py>



```
C:\Users\Emmanuel\Desktop\GGE5403\CONTENT\Module 3 - Web Services... - X
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Ex4.py x
1 # !C:/ms4w/Python/python
2
3 # WGS84 --> Web Mercator
4 import pyproj
5 p = pyproj.Proj(init='epsg:3857')
6 (x1, y1) = (-66.666667, 45.95)
7 (x2, y2) = p(x1,y1)
8
9 print ("Content-Type: text/plain;charset=utf-8")
10 print() # must have a free line here
11
12 print("WGS84: %9.3f %11.3f" % (x1,y1))
13 print("Web Mercator: %9.3f %11.3f" % (x2,y2))
14
15
16
length : Ln : 1 Col : 1 Pos : 1 Windows (CR LF) UTF-8 IN
```

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Ex4.py>

The screenshot shows a Python code editor and a web browser side-by-side.

Code Editor (Left):

```
1 # !C:\ms4w\Python\python
2
3 # WGS84 --> Web Mercator
4 import pyproj
5 p = pyproj.Proj(init='epsg:3857')
6 (x1, y1) = (-66.666667, 45.95)
7 (x2, y2) = p(x1,y1)
8
9 print ("Content-Type: text/plain;charset=utf-8")
10 print() # must have a free line here
11
12 print("WGS84: %9.3f %11.3f" % (x1,y1))
13 print("Web Mercator: %9.3f %11.3f" % (x2,y2))
14
15
16
```

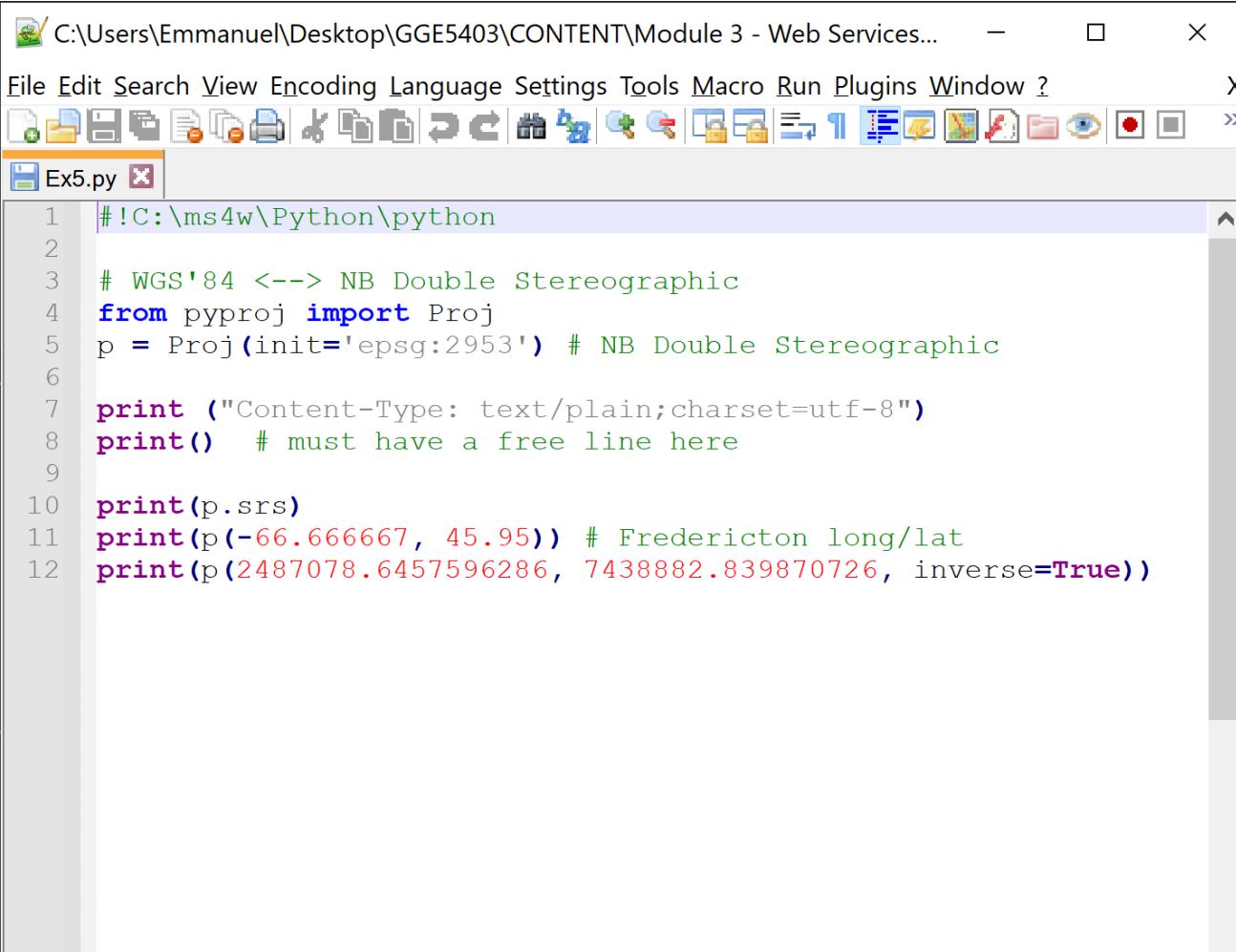
Browser Window (Right):

Address bar: gaia.gge.unb.ca:8080/gge5403/labs/3/Ex4.py

Content:

```
WGS84: -66.667 45.950
Web Mercator: -7421299.423 5772340.302
```

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Ex5.py>



The screenshot shows a Python code editor window titled "C:\Users\Emmanuel\Desktop\GGE5403\CONTENT\Module 3 - Web Services...". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and Help. The toolbar below the menu contains various icons for file operations like Open, Save, Print, and Find. The current tab is "Ex5.py". The code in the editor is:

```
1 #!C:\ms4w\Python\python
2
3 # WGS'84 <--> NB Double Stereographic
4 from pyproj import Proj
5 p = Proj(init='epsg:2953') # NB Double Stereographic
6
7 print ("Content-Type: text/plain;charset=utf-8")
8 print() # must have a free line here
9
10 print(p.srs)
11 print(p(-66.666667, 45.95)) # Fredericton long/lat
12 print(p(2487078.6457596286, 7438882.839870726, inverse=True))
```

At the bottom of the editor, status bars show "length : 3 Ln : 1 Col : 1 Pos : 1", "Windows (CR LF)", "UTF-8", and "IN".

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Ex5.py>

The screenshot shows a Python code editor and a web browser side-by-side.

Code Editor (Left):

```
1 #!C:\ms4w\Python\python
2
3 # WGS'84 <--> NB Double Stereographic
4 from pyproj import Proj
5 p = Proj(init='epsg:2953') # NB Double Stereographic
6
7 print ("Content-Type: text/plain;charset=utf-8")
8 print() # must have a free line here
9
10 print(p.srs)
11 print(p(-66.666667, 45.95)) # Fredericton long/lat
12 print(p(2487078.6457596286, 7438882.839970726, inverse=True))
```

Browser Window (Right):

gaia.gge.unb.ca:8080/gge5403/la

Not secure | gaia.gge.unb.ca:8080/gge5403/labs/3/Ex5.py

```
+proj=sterea +lat_0=46.5 +lon_0=-66.5 +k=0.999912 +x_0=2500000 +y_0=7500000 +ellps=GRS80
+towgs84=0,0,0,0,0,0,0 +units=m +no_defs
(2487078.6457598484, 7438882.8399755)
(-66.666667, 45.94999999905733)
```

length : 1 Ln : 1 Col : 1 Pos : 1

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Ex6.py>

The image shows two windows from a Python code editor. The left window displays the Python script `Ex6.py`. The right window shows the contents of the CSV file `points.csv`.

Ex6.py Content:

```
#!/usr/bin/python
#Read input points from file, do conversion, save in another file
import sys
import pyproj
import csv

#definition of a function
def proj_conversion(input_filename, output_filename):
    infile = open(input_filename, 'r')
    csv_f = csv.reader(infile)

    outfile = open(output_filename, 'w')
    p = pyproj.Proj(init='epsg:3857')

    print ("Content-Type: text/plain;charset=utf-8")
    print() # must have a free line here

    for row in csv_f:
        x1 = float(row[0])
        y1 = float(row[1])
        (x2, y2) = p(x1,y1)
        print("%.2f,%2f --> %.2f,%2f" % (x1,y1,x2,y2))
        outfile.write("%.2f,%2f\n" % (x2,y2))

    infile.close()
    outfile.close()

def main():
    f1 = "./points.csv"
    f2 = "./p.csv"
    proj_conversion(f1,f2)

if __name__ == '__main__':
    main()
```

points.csv Content:

Row	Column 1	Column 2
1	30.00	45.00
2	180.00	85.05
3	0.00	90.00
4	-180.00	-45.00
5	45.00	45.00

Length : 833 Ln : 1 Col : 1 Pos : 1 Windows (CR LF) UTF-8 IN

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Ex6.py>

The image shows three windows illustrating the execution of a Python script named `Ex6.py`.

Code Editor 1 (Left): Displays the contents of `Ex6.py`. The script reads input points from a CSV file, performs a coordinate transformation, and writes the results to another CSV file.

```
#!/usr/bin/python
#Read input points from file, do conversion, save in another file
import sys
import pyproj
import csv

#definition of a function
def proj_conversion(input_filename, output_filename):
    infile = open(input_filename, 'r')
    csv_f = csv.reader(infile)

    outfile = open(output_filename, 'w')

    p = pyproj.Proj(init='epsg:3857')

    print ("Content-Type: text/plain;charset=utf-8")
    print() # must have a free line here

    for row in csv_f:
        x1 = float(row[0])
        y1 = float(row[1])
        (x2, y2) = p(x1,y1)
        print("%.2f,%2f --> %.2f,%2f" % (x1,y1,x2,y2))

        outfile.write("%.2f,%2f\n" % (x2,y2))

    infile.close()
    outfile.close()

def main():
    f1 = "./points.csv"
    f2 = "./p.csv"
    proj_conversion(f1,f2)

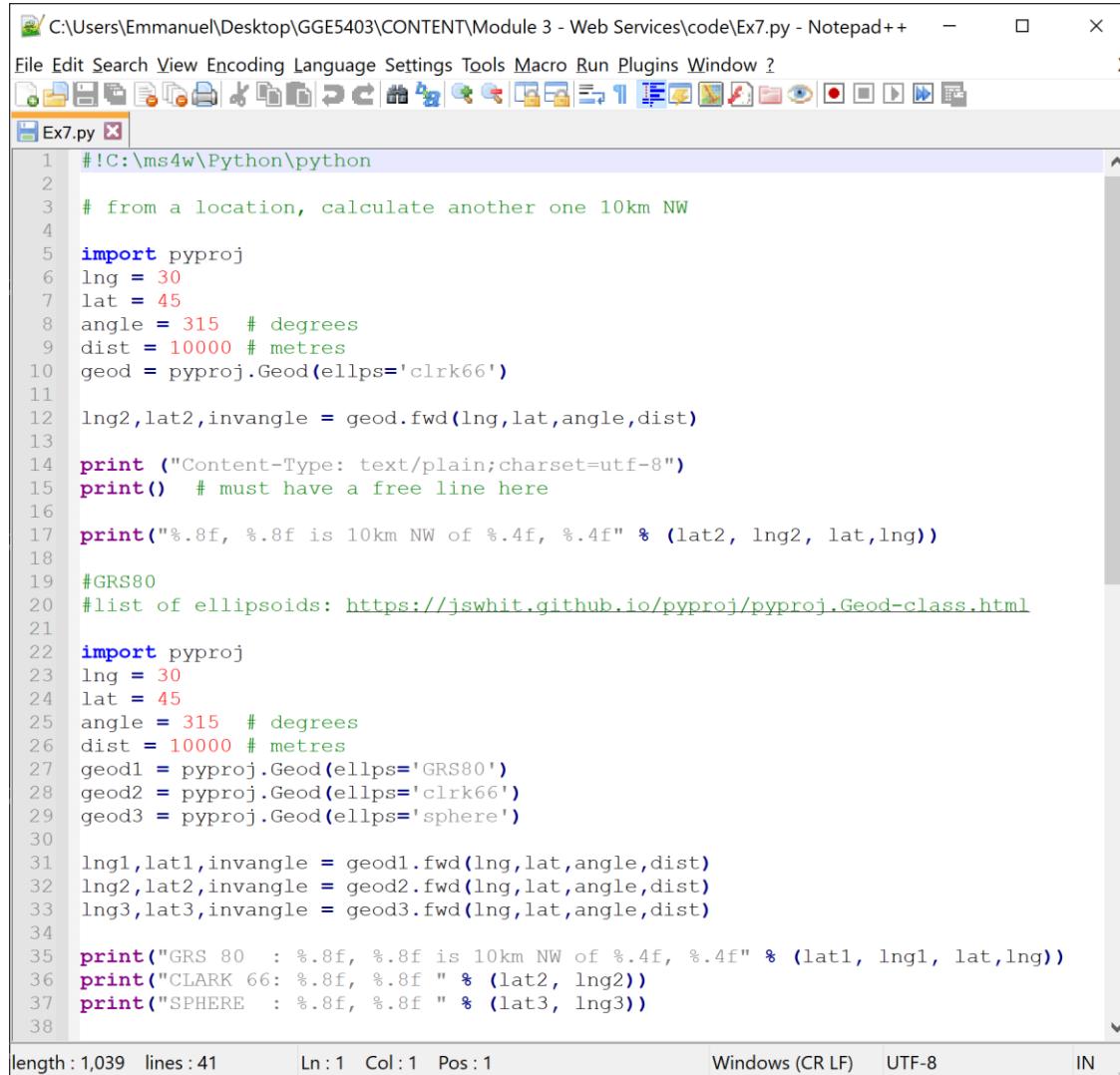
if __name__ == '__main__':
    main()
```

Code Editor 2 (Top Right): Shows the contents of `points.csv`, which contains five points: (30.00, 45.00), (180.00, 85.05), (0.00, 90.00), (-180.00, -45.00), and (45.00, 45.00).

Point ID	X	Y
1	30.00	45.00
2	180.00	85.05
3	0.00	90.00
4	-180.00	-45.00
5	45.00	45.00

Browser Window (Bottom): Shows the output of the script's standard output. It lists the original coordinates followed by their transformed coordinates: (30.00, 45.00) --> (3339584.72, 5621521.49), (180.00, 85.05) --> (20037508.34, 20036051.92), (0.00, 90.00) --> (100000000000000019884624838656.00, 100000000000000019884624838656.00), (-180.00, -45.00) --> (-20037508.34, -5621521.49), and (45.00, 45.00) --> (5009377.09, 5621521.49).

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Ex7.py>



The screenshot shows a Notepad++ window with the file 'Ex7.py' open. The code uses the pyproj library to calculate a point 10km NW from a given location. It compares results using GRS80, Clark 66, and Sphere ellipsoids.

```
#!C:/ms4w/Python/python
# from a location, calculate another one 10km NW
import pyproj
lng = 30
lat = 45
angle = 315 # degrees
dist = 10000 # metres
geod = pyproj.Geod(ellps='clrk66')

lng2,lat2,invangle = geod.fwd(lng,lat,angle,dist)
print ("Content-Type: text/plain;charset=utf-8")
print() # must have a free line here
print("%.8f, %.8f is 10km NW of %.4f, %.4f" % (lat2, lng2, lat,lng))

#GRS80
#list of ellipsoids: https://jswhit.github.io/pyproj/pyproj.Geod-class.html

import pyproj
lng = 30
lat = 45
angle = 315 # degrees
dist = 10000 # metres
geod1 = pyproj.Geod(ellps='GRS80')
geod2 = pyproj.Geod(ellps='clrk66')
geod3 = pyproj.Geod(ellps='sphere')

lng1,lat1,invangle = geod1.fwd(lng,lat,angle,dist)
lng2,lat2,invangle = geod2.fwd(lng,lat,angle,dist)
lng3,lat3,invangle = geod3.fwd(lng,lat,angle,dist)

print("GRS 80 : %.8f, %.8f is 10km NW of %.4f, %.4f" % (lat1, lng1, lat,lng))
print("CLARK 66: %.8f, %.8f " % (lat2, lng2))
print("SPHERE : %.8f, %.8f " % (lat3, lng3))
```

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Ex7.py>

The screenshot shows a Notepad++ window with the file 'Ex7.py' open. The code uses the pyproj library to calculate a new location 10km NW from a starting point at 30°E, 45°N. It demonstrates calculations for GRS80, Clark 66, and a sphere. The browser window below shows the output of the script.

```
1  #!C:\ms4w\Python\python
2
3  # from a location, calculate another one 10km NW
4
5  import pyproj
6  lng = 30
7  lat = 45
8  angle = 315 # degrees
9  dist = 10000 # metres
10 geod = pyproj.Geod(ellps='clrk66')
11
12 lng2,lat2,invangle = geod.fwd(lng,lat,angle,dist)
13
14 print ("Content-Type: text/plain;charset=utf-8")
15 print() # must have a free line here
16
17 print("%.8f, %.8f is 10km NW of %.4f, %.4f" % (lat2, lng2, lat,lng))
18
19 #GRS80
20 #list of ellipsoids: https://jswhit.github.io/pyproj/pyproj.Geod-class.html
21
22 import pyproj
23 lng = 30
24 lat = 45
25 angle = 315 # degrees
26 dist = 10000 # metres
27 geod1 = pyproj.Geod(ellps='GRS80')
28 geod2 = pyproj.Geod(ellps='clrk66')
29 geod3 = pyproj.Geod(ellps='sphere')
30
31 lng1,lat1,invangle = geod1.fwd(lng,lat,angle,dist)
32 lng2,lat2,invangle = geod2.fwd(lng,lat,angle,dist)
33 lng3,lat3,invangle = geod3.fwd(lng,lat,angle,dist)
34
35 print("GRS 80 : %.8f, %.8f is 10km NW of %.4f, %.4f"
36 print("CLARK 66: %.8f, %.8f " % (lat2, lng2))
37 print("SPHERE : %.8f, %.8f " % (lat3, lng3))
```

The browser window displays the output of the Python script. It shows the calculated coordinates for each ellipsoid and a summary statement.

gaia.gge.unb.ca:8080/gge5403/labs/3/Ex7.py

45.06359266, 29.91022222 is 10km NW of 45.0000, 30.0000
GRS 80 : 45.06359216, 29.91021957 is 10km NW of 45.0000, 30.0000
CLARK 66: 45.06359266, 29.91022222
SPHERE : 45.06355633, 29.90996787

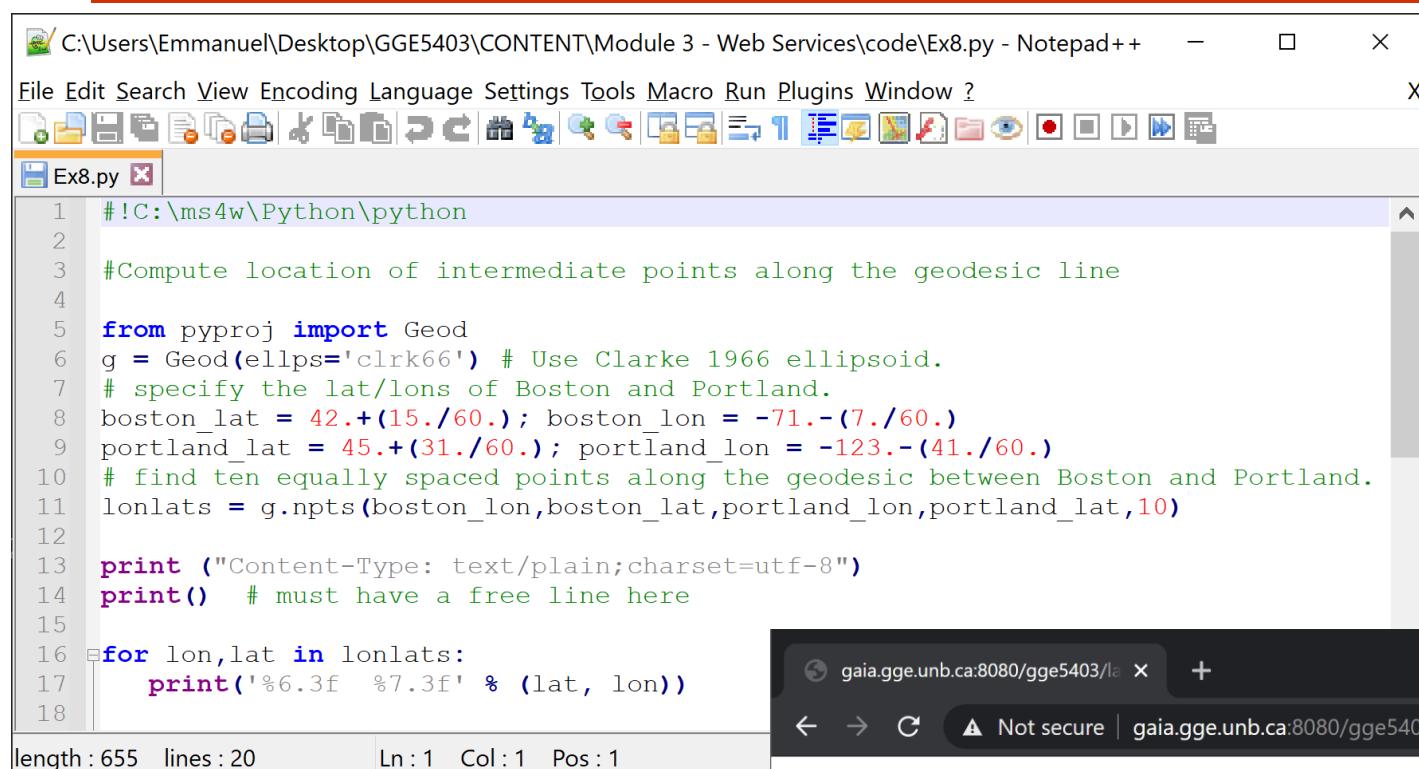
<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Ex8.py>

The screenshot shows a Notepad++ window with the following details:

- Title Bar:** C:\Users\Emmanuel\Desktop\GGE5403\CONTENT\Module 3 - Web Services\code\Ex8.py - Notepad++
- Menu Bar:** File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
- Toolbar:** Standard file operations like Open, Save, Print, Find, Copy, Paste, etc.
- Document Area:** The code is named "Ex8.py".

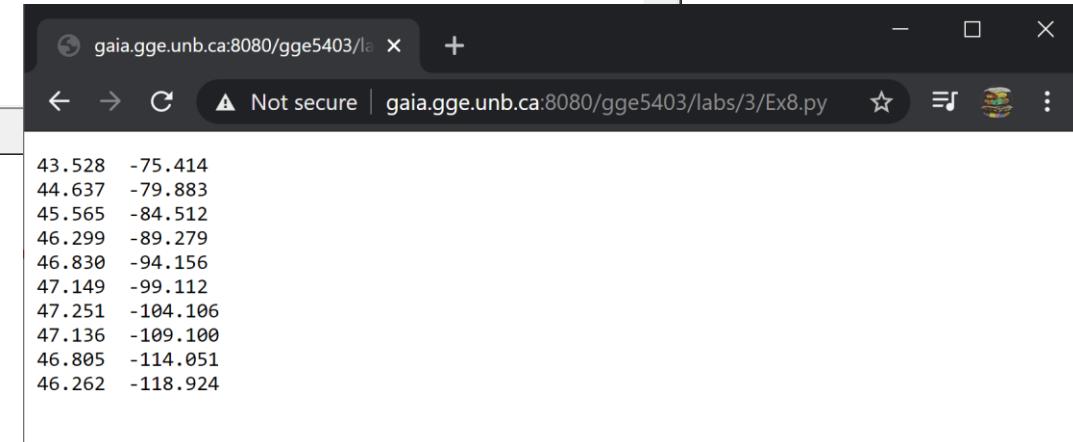
```
1  #!C:\ms4w\Python\python
2
3 #Compute location of intermediate points along the geodesic line
4
5 from pyproj import Geod
6 g = Geod(ellps='clrk66') # Use Clarke 1966 ellipsoid.
7 # specify the lat/lons of Boston and Portland.
8 boston_lat = 42.+(15./60.); boston_lon = -71.-(7./60.)
9 portland_lat = 45.+(31./60.); portland_lon = -123.-(41./60.)
10 # find ten equally spaced points along the geodesic between Boston and Portland.
11 lonlats = g.npts(boston_lon,boston_lat, portland_lon, portland_lat, 10)
12
13 print ("Content-Type: text/plain; charset=utf-8")
14 print() # must have a free line here
15
16 for lon,lat in lonlats:
17     print('%.6.3f %.7.3f' % (lat, lon))
18
```
- Status Bar:** length : 655 lines : 20 Ln : 1 Col : 1 Pos : 1 Windows (CR LF) UTF-8 IN

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Ex8.py>



The screenshot shows a Notepad++ window with the file "Ex8.py" open. The code uses the pyproj library to calculate ten intermediate points between Boston and Portland. The browser window below shows the output of the script, listing the latitude and longitude coordinates for each point.

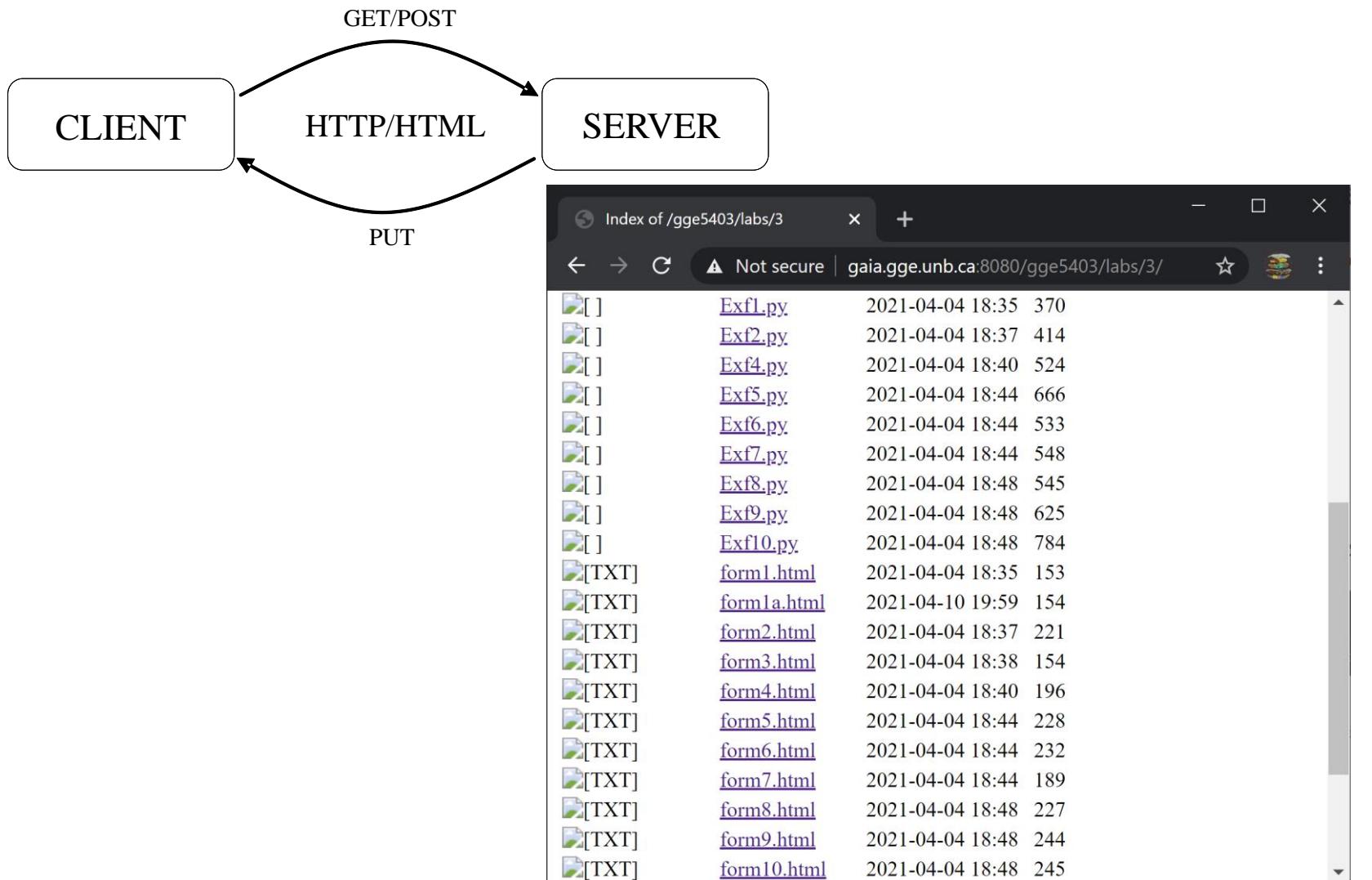
```
1  #!C:/ms4w/Python/python
2
3  #Compute location of intermediate points along the geodesic line
4
5  from pyproj import Geod
6  g = Geod(ellps='clrk66') # Use Clarke 1966 ellipsoid.
7  # specify the lat/lons of Boston and Portland.
8  boston_lat = 42.+(15./60.); boston_lon = -71.-(7./60.)
9  portland_lat = 45.+(31./60.); portland_lon = -123.-(41./60.)
10 # find ten equally spaced points along the geodesic between Boston and Portland.
11 lonlats = g.npts(boston_lon,boston_lat,Portland_lon,Portland_lat,10)
12
13 print ("Content-Type: text/plain;charset=utf-8")
14 print() # must have a free line here
15
16 for lon,lat in lonlats:
17     print('%.6.3f %.7.3f' % (lat, lon))
18
length : 655  lines : 20  Ln : 1  Col : 1  Pos : 1
```



The browser window displays the output of the Python script, which lists the coordinates of ten intermediate points along the geodesic line between Boston and Portland. The coordinates are listed in pairs of latitude and longitude.

Latitude	Longitude
43.528	-75.414
44.637	-79.883
45.565	-84.512
46.299	-89.279
46.830	-94.156
47.149	-99.112
47.251	-104.106
47.136	-109.100
46.805	-114.051
46.262	-118.924

More Examples...



<http://gaia.gge.unb.ca:8080/gge5403/labs/3/form1.html>

method=get

```
form1.html
1 <form name="dimension" action="Exf1.py" method="get">
2 Dimension: <input type="number" name="matrix_dim">
3 <input type="submit" value="Submit">
4 </form>
```

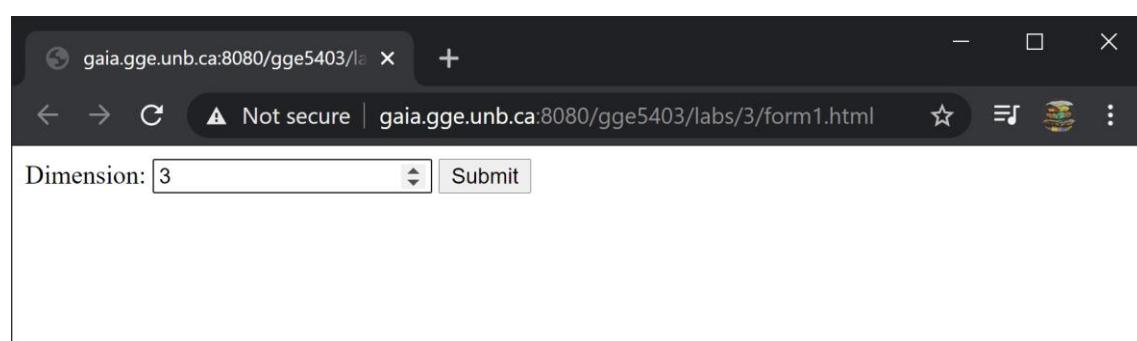
```
Exf1.py
1 #!C:/ms4w/Python/python
2
3 import numpy as np
4
5 # pass parameter from form
6 import cgi
7 form = cgi.FieldStorage()
8 dim = int(form.getvalue('matrix_dim'))
9
10 d = np.random.random((dim, dim)) # Create an array filled with random values
11
12 print ("Content-Type: text/plain; charset=utf-8")
13 print() # must have a free line here
14
15 print(dim)
16 print(type(dim))
17 print(d)
18
```

length : 153 lines : 4 Ln : 1 Col : 1

length : 370 lines : 18 Ln : 1 Col : 1 Pos : 1

Windows (CR LF) UTF-8 IN

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/form1.html>



method=get

http://gaia.gge.unb.ca:8080/gge5403/labs/3/Exf1.py?matrix_dim=3

A screenshot of a web browser window. The address bar shows the URL "gaia.gge.unb.ca:8080/gge5403/labs/3/Exf1.py?matrix_dim=3". Below the address bar, there is a message "Not secure". The main content area displays the output of a Python script:

```
3
<class 'int'>
[[0.57554108 0.09586823 0.38722215]
 [0.54467604 0.78815987 0.98446596]
 [0.57105049 0.70621981 0.17393914]]
```

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/form1a.html>

method=post

C:\Users\Emmanuel\Desktop\GGE5403\CONTENT\Module 3 - Web Services\code1\form1a.html - ... X

File Edit Search Encoding Language Settings Tools Macro Run Plugins Window ?

Exf1.py form1a.html

```
1 <form name="dimension" action="Exf1.py" method="post">
2 Dimension: <input type="number" name="matrix_dim">
3 <input type="submit" value="Submit">
4 </form>
```

length : 154 lines : 4 Ln : 4 Col : 9

C:\Users\Emmanuel\Desktop\GGE5403\CONTENT\Module 3 - Web Services\code1\Exf1.py - Note... X

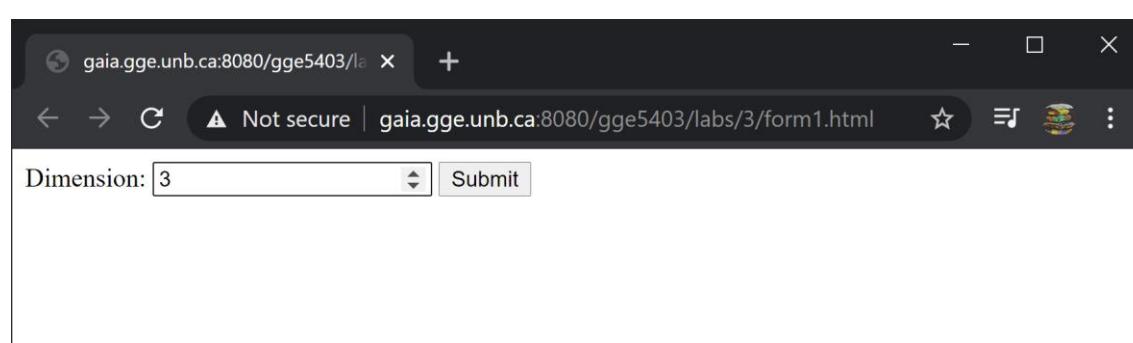
File Edit Search Encoding Language Settings Tools Macro Run Plugins Window ?

form1.html Exf1.py

```
1 #!C:/ms4w/Python/python
2
3 import numpy as np
4
5 # pass parameter from form
6 import cgi
7 form = cgi.FieldStorage()
8 dim = int(form.getvalue('matrix_dim'))
9
10 d = np.random.random((dim, dim)) # Create an array filled with random values
11
12 print ("Content-Type: text/plain; charset=utf-8")
13 print() # must have a free line here
14
15 print(dim)
16 print(type(dim))
17 print(d)
18
```

length : 370 lines : 18 Ln : 1 Col : 1 Pos : 1 Windows (CR LF) UTF-8 IN

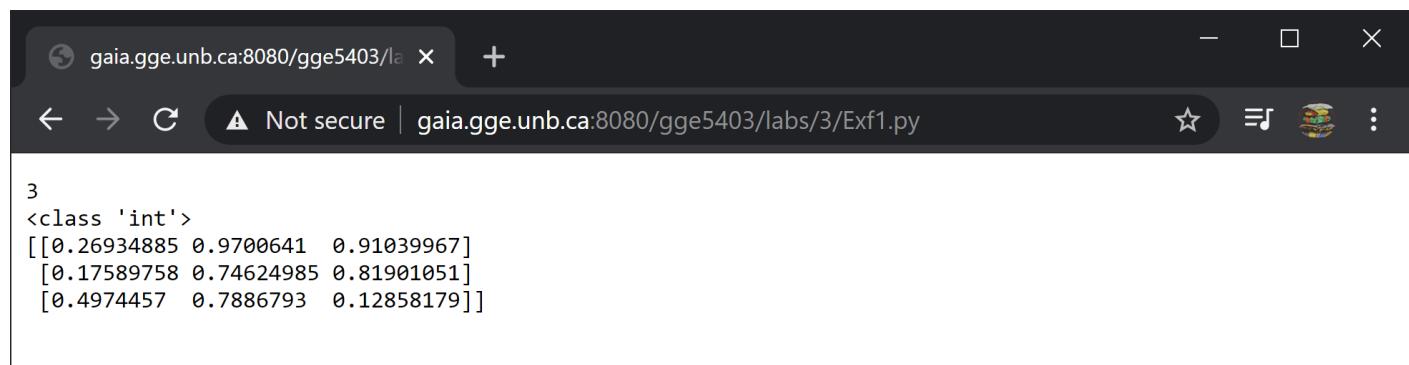
<http://gaia.gge.unb.ca:8080/gge5403/labs/3/form1a.html>



A screenshot of a web browser window. The address bar shows the URL "gaia.gge.unb.ca:8080/gge5403/labs/3/form1a.html". Below the address bar, there is a message "Not secure". The main content area contains a form with a dropdown menu labeled "Dimension:" containing the value "3". To the right of the browser window, the text "method=post" is displayed in blue.

method=post

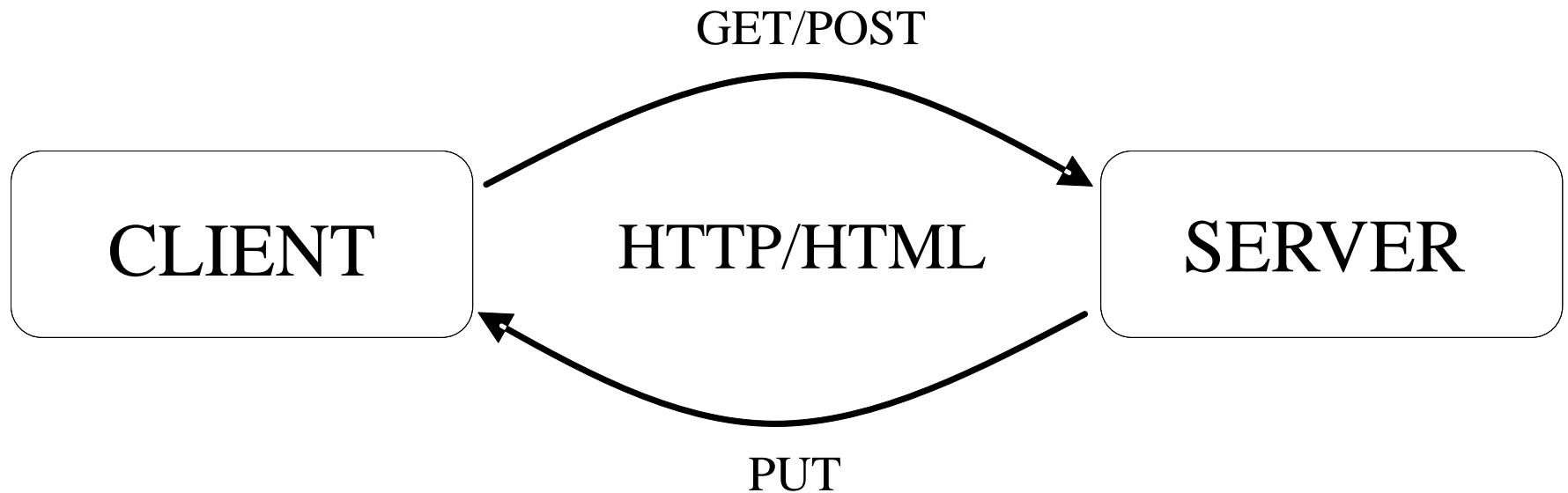
http://gaia.gge.unb.ca:8080/gge5403/labs/3/Exf1.py?matrix_dim=3



A screenshot of a web browser window. The address bar shows the URL "gaia.gge.unb.ca:8080/gge5403/labs/3/Exf1.py". Below the address bar, there is a message "Not secure". The main content area displays the output of a Python script, which includes the integer value "3" and a 3x3 matrix represented as a list of lists:

```
3
<class 'int'>
[[0.26934885  0.9700641  0.91039967]
 [0.17589758  0.74624985  0.81901051]
 [0.4974457   0.7886793   0.12858179]]
```

GET (vs) POST



Passing Information using GET method

The GET method sends the encoded user information appended to the page request. The page and the encoded information are separated by the ? character as follows –

```
http://www.test.com/cgi-bin/hello.py?key1=value1&key2=value2
```

The GET method is the default method to pass information from browser to web server and it produces a long string that appears in your browser's Location:box. Never use GET method if you have password or other sensitive information to pass to the server. The GET method has size limitation: only 1024 characters can be sent in a request string. The GET method sends information using QUERY_STRING header and will be accessible in your CGI Program through QUERY_STRING environment variable.

You can pass information by simply concatenating key and value pairs along with any URL or you can use HTML <FORM> tags to pass information using GET method.

Passing Information Using POST Method

A generally more reliable method of passing information to a CGI program is the POST method. This packages the information in exactly the same way as GET methods, but instead of sending it as a text string after a ? in the URL it sends it as a separate message. This message comes into the CGI script in the form of the standard input.

Below is same `hello_get.py` script which handles GET as well as POST method.

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/form2.html>

The image shows two Notepad windows side-by-side. The left window displays an HTML form named 'dimension' with fields for 'matrix_dim_x' and 'matrix_dim_y'. The right window displays a Python script named 'Exf2.py' that generates a random matrix based on user input.

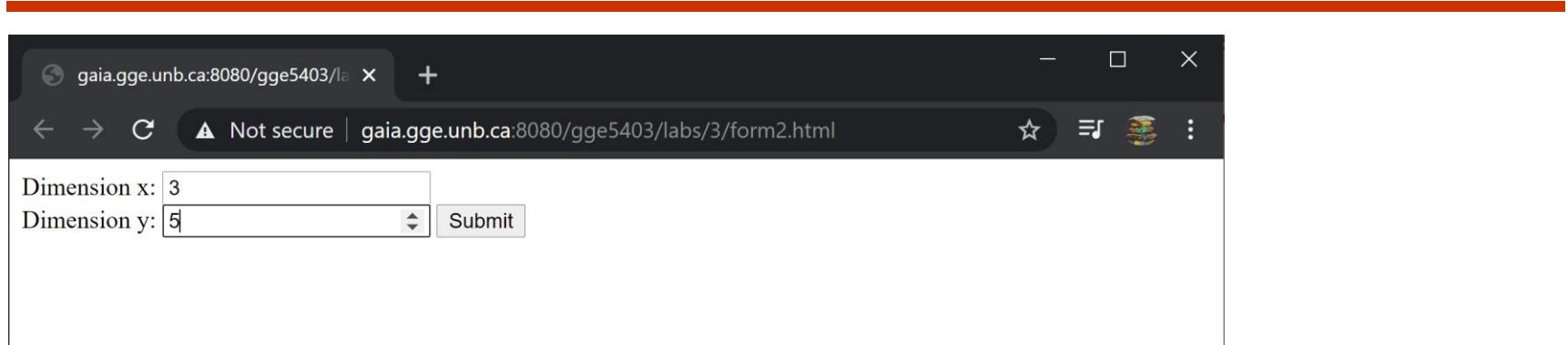
Left Window (Notepad):

```
1 <form name="dimension" action="Exf2.py" method="get">
2 Dimension x: <input type="number" name="matrix_dim_x">
3 <br />
4 Dimension y: <input type="number" name="matrix_dim_y">
5 <input type="submit" value="Submit">
6 </form>
```

Right Window (Notepad):

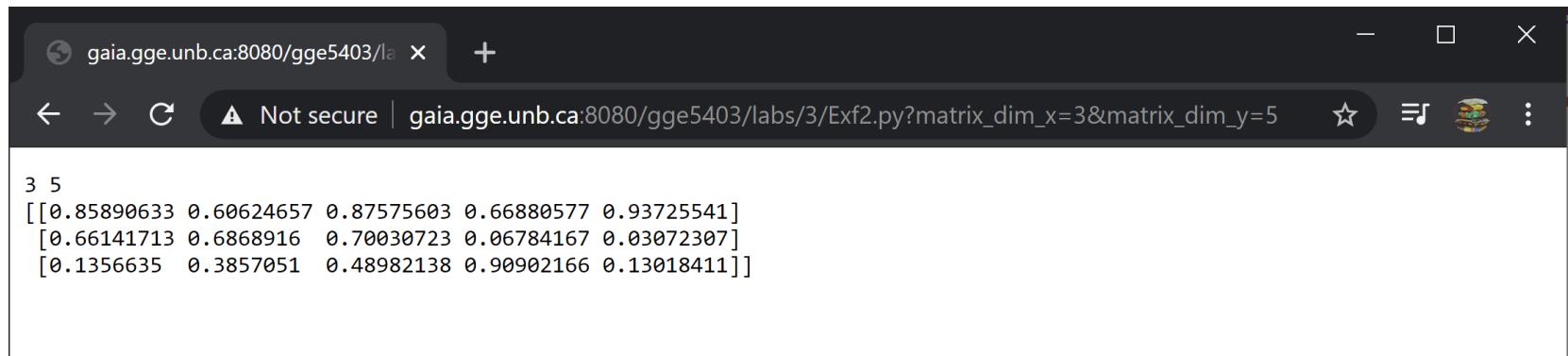
```
1 #!C:\ms4w\Python\python
2
3 import numpy as np
4
5 # pass parameter from form
6 import cgi
7 form = cgi.FieldStorage()
8 dim_x = int(form.getvalue('matrix_dim_x'))
9 dim_y = int(form.getvalue('matrix_dim_y'))
10
11 d = np.random.random((dim_x, dim_y)) # Create an array filled with random values
12
13 print ("Content-Type: text/plain; charset=utf-8")
14 print() # must have a free line here
15
16 print(dim_x, dim_y)
17 print(d)
```

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/form2.html>



A screenshot of a web browser window. The address bar shows the URL `gaia.gge.unb.ca:8080/gge5403/labs/3/form2.html`. The page content is a form with two input fields: "Dimension x:" containing the value "3" and "Dimension y:" containing the value "5". A "Submit" button is located to the right of the "y" field.

http://gaia.gge.unb.ca:8080/gge5403/labs/3/Exf2.py?matrix_dim_x=3&matrix_dim_y=5



A screenshot of a web browser window. The address bar shows the URL `gaia.gge.unb.ca:8080/gge5403/labs/3/Exf2.py?matrix_dim_x=3&matrix_dim_y=5`. The page displays the output of the Python script, which consists of a 3x5 matrix of floating-point numbers:

```
3 5
[[0.85890633 0.60624657 0.87575603 0.66880577 0.93725541]
 [0.66141713 0.6868916 0.70030723 0.06784167 0.03072307]
 [0.1356635 0.3857051 0.48982138 0.90902166 0.13018411]]
```

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/form4.html>

The image shows two Notepad windows side-by-side. The left window displays an HTML form file named 'form4.html' with the following content:

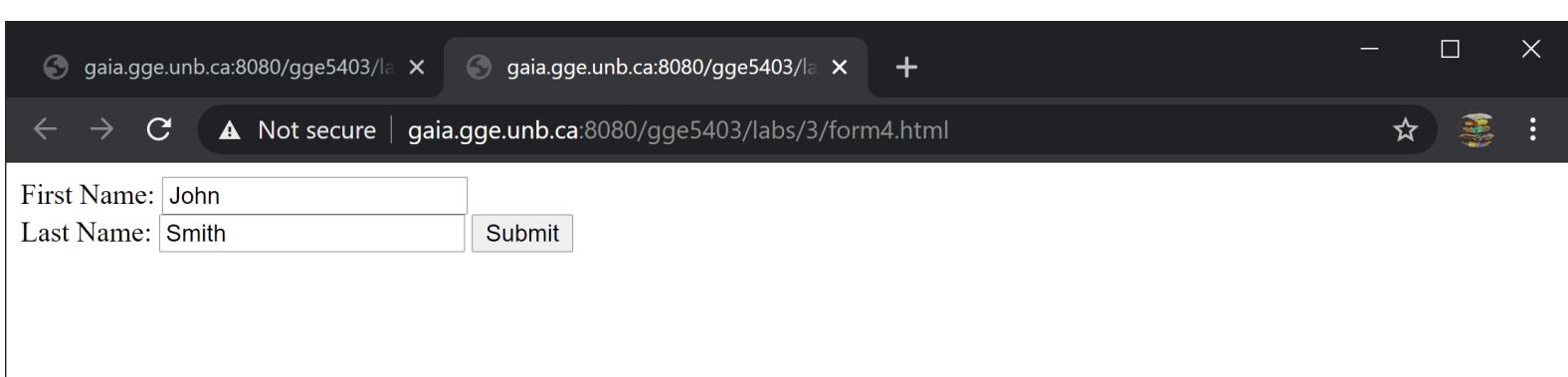
```
1 <form action="Exf4.py" method="post">
2 First Name: <input type="text" name="first_name"><br />
3 Last Name: <input type="text" name="last_name" />
4
5 <input type="submit" value="Submit" />
6 </form>
```

The right window displays a Python CGI script file named 'Exf4.py' with the following content:

```
1 #!C:\ms4w\Python\python
2
3 # Import modules for CGI handling
4 import cgi, cgitb
5
6 # Create instance of FieldStorage
7 form = cgi.FieldStorage()
8
9 # Get data from fields
10 first_name = form.getvalue('first_name')
11 last_name = form.getvalue('last_name')
12
13 print ("Content-type:text/html\r\n\r\n")
14 print ("<html>")
15 print ("<head>")
16 print ("<title>Hello - Second CGI Program</title>")
17 print ("</head>")
18 print ("<body>")
19 print ("<h2>Hello %s %s</h2>" % (first_name, last_name))
20 print ("</body>")
21 print ("</html>")
```

Both windows have standard Notepad toolbars and status bars at the bottom.

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/form4.html>

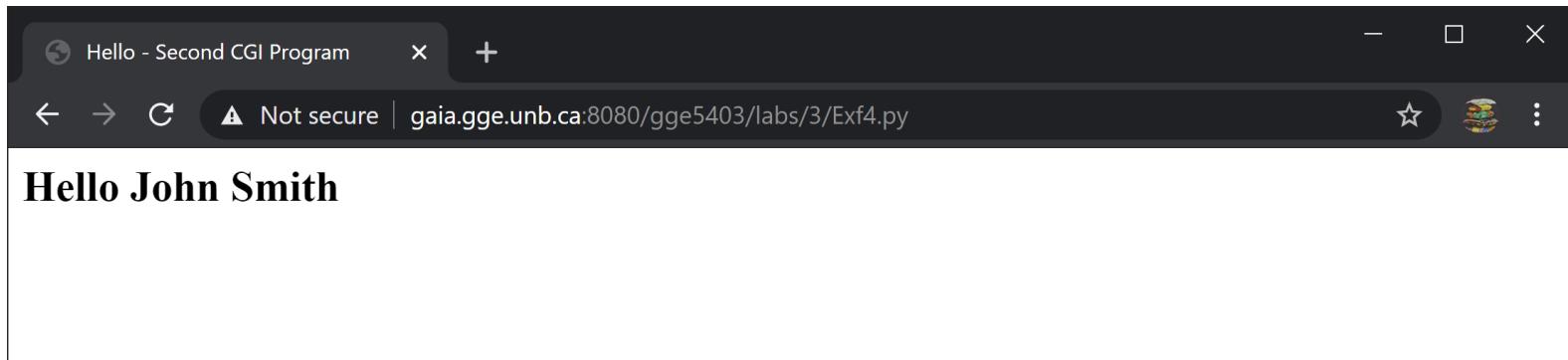


A screenshot of a web browser window. The address bar shows two tabs: "gaia.gge.unb.ca:8080/gge5403/la" and "gaia.gge.unb.ca:8080/gge5403/la". The main content area displays a form with the following fields:

First Name:

Last Name:

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Exf4.py>



<http://gaia.gge.unb.ca:8080/gge5403/labs/3/form5.html>

The image shows two side-by-side code editors. The left editor displays an HTML form file named 'form5.html' with the following content:

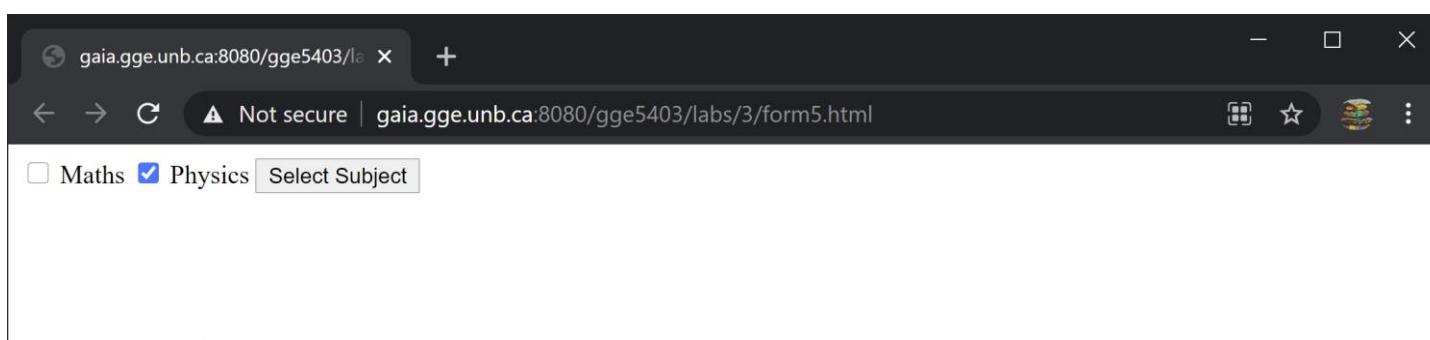
```
1 <form action="Exf5.py" method="POST" target="_blank">
2   <input type="checkbox" name="maths" value="on" /> Maths
3   <input type="checkbox" name="physics" value="on" /> Physics
4   <input type="submit" value="Select Subject" />
5 </form>
```

The right editor displays a Python CGI script named 'Exf5.py' with the following code:

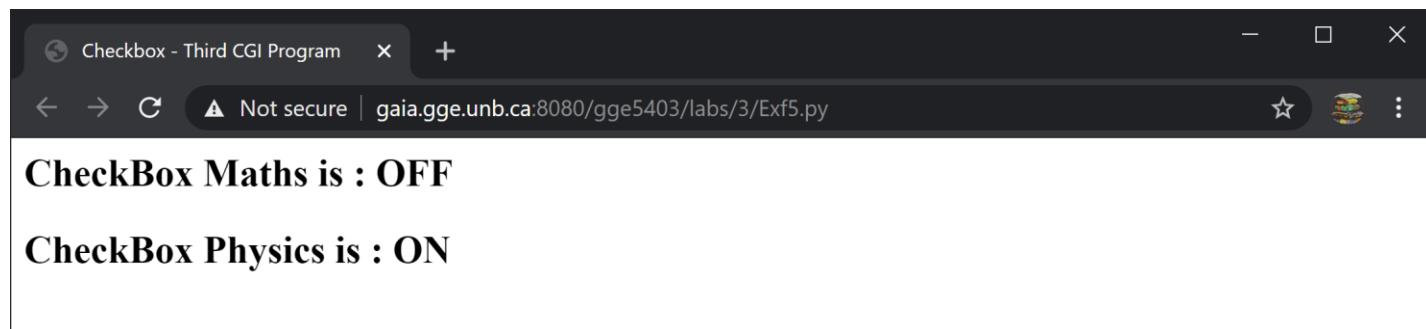
```
1 # !C:\ms4w\Python\python
2
3 # Import modules for CGI handling
4 import cgi, cgitb
5
6 # Create instance of FieldStorage
7 form = cgi.FieldStorage()
8
9 # Get data from fields
10 if form.getvalue('maths'):
11     math_flag = "ON"
12 else:
13     math_flag = "OFF"
14
15 if form.getvalue('physics'):
16     physics_flag = "ON"
17 else:
18     physics_flag = "OFF"
19
20 print ("Content-type:text/html\r\n\r\n")
21 print ("<html>")
22 print ("<head>")
23 print ("<title>Checkbox - Third CGI Program</title>")
24 print ("</head>")
25 print ("<body>")
26 print ("<h2> CheckBox Maths is : %s</h2>" % math_flag)
27 print ("<h2> CheckBox Physics is : %s</h2>" % physics_flag)
28 print ("</body>")
29 print ("</html>")
```

Both editors have a toolbar with various icons and a status bar at the bottom showing file paths, line counts, and character counts.

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/form5.html>



<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Exf5.py>



<http://gaia.gge.unb.ca:8080/gge5403/labs/3/form6.html>

The image shows two side-by-side Notepad windows. The left window displays an HTML form with radio buttons for selecting a subject. The right window displays a Python CGI script that handles the form submission.

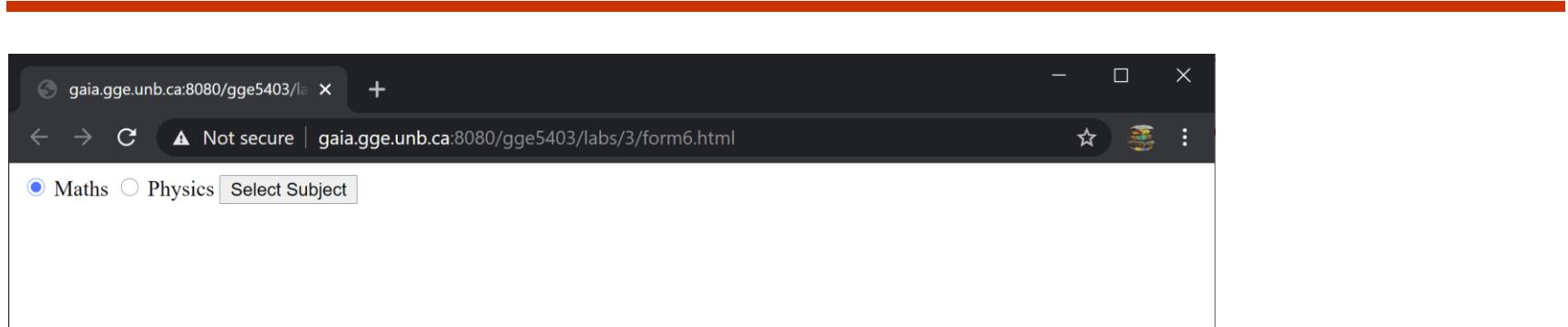
Left Window (form6.html):

```
1 <form action="Exf6.py" method="post" target="_blank">
2   <input type="radio" name="subject" value="maths" /> Maths
3   <input type="radio" name="subject" value="physics" /> Physics
4   <input type="submit" value="Select Subject" />
5 </form>
```

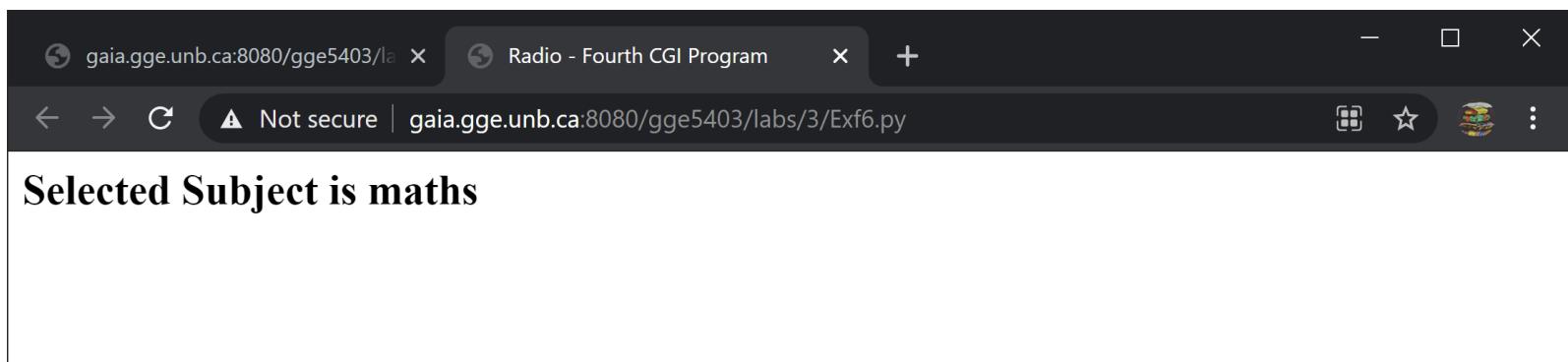
Right Window (Exf6.py):

```
1 #!C:\ms4w\Python\python
2 # Import modules for CGI handling
3 import cgi, cgitb
4
5 # Create instance of FieldStorage
6 form = cgi.FieldStorage()
7
8 # Get data from fields
9 if form.getvalue('subject'):
10   subject = form.getvalue('subject')
11 else:
12   subject = "Not set"
13
14 print ("Content-type:text/html\r\n\r\n")
15 print ("<html>")
16 print ("<head>")
17 print ("<title>Radio - Fourth CGI Program</title>")
18 print ("</head>")
19 print ("<body>")
20 print ("<h2> Selected Subject is %s</h2>" % subject)
21 print ("</body>")
22 print ("</html>")
```

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/form6.html>



<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Exf6.py>



<http://gaia.gge.unb.ca:8080/gge5403/labs/3/form7.html>

The image shows two Notepad windows side-by-side. The left window, titled 'form7.html', contains the following HTML code:

```
1 <form action="Exf7.py" method="post" target="_blank">
2 <textarea name="textcontent" cols="40" rows="4">
3 Type your text here...
4 </textarea>
5 <input type="submit" value="Submit" />
6 </form>
```

The right window, titled 'Exf7.py', contains the following Python code:

```
1 #!C:\ms4w\Python\python
2 # Import modules for CGI handling
3 import cgi, cgitb
4
5 # Create instance of FieldStorage
6 form = cgi.FieldStorage()
7
8 # Get data from fields
9 if form.getvalue('textcontent'):
10     text_content = form.getvalue('textcontent')
11 else:
12     text_content = "Not entered"
13
14 print ("Content-type:text/html\r\n\r\n")
15 print ("<html>")
16 print ("<head>")
17 print ("<title>Text Area - Fifth CGI Program</title>")
18 print ("</head>")
19 print ("<body>")
20 print ("<h2> Entered Text Content is %s</h2>" % text_content)
21 print ("</body>")
```

Both windows have standard Windows-style toolbars at the top and status bars at the bottom indicating file length, line count, column count, and position.

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/form7.html>

The image displays two side-by-side screenshots of a web browser window. Both screenshots show a simple form consisting of a text input field and a 'Submit' button. In the first screenshot, the text input field contains the placeholder text 'Type your text here...'. In the second screenshot, the text input field contains the text 'UNB is located in New Brunswick'. The browser's title bar indicates the URL is 'gaia.gge.unb.ca:8080/gge5403/labs/3/form7.html' and the connection is 'Not secure'.

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Exf7.py>

The image shows a single screenshot of a web browser window. The title bar of the browser says 'Text Area - Fifth CGI Program'. The main content area of the browser displays the text 'Entered Text Content is UNB is located in New Brunswick' in a large, bold, black font. The browser's title bar also shows the URL 'gaia.gge.unb.ca:8080/gge5403/labs/3/Exf7.py' and the status 'Not secure'.

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/form8.html>

The image shows two side-by-side Notepad windows. The left window contains an HTML form code snippet, and the right window contains a Python CGI script.

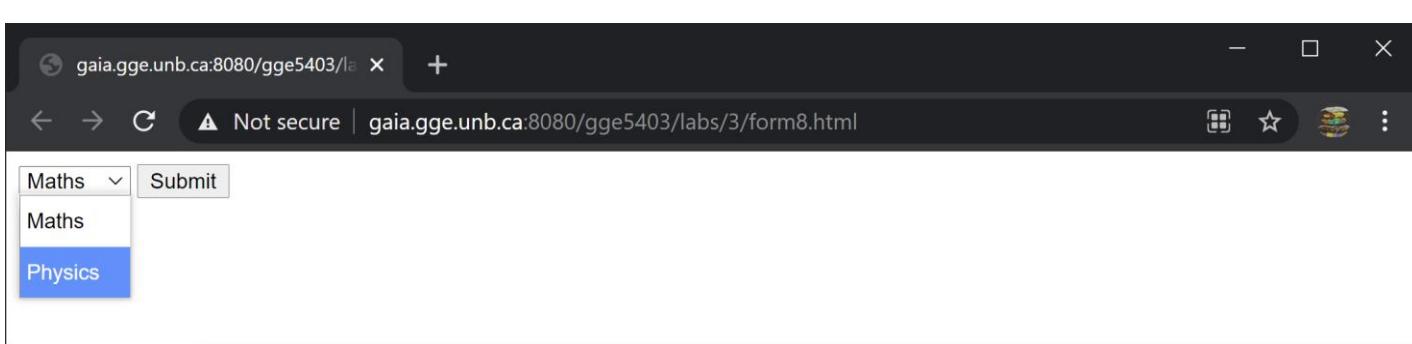
Left Window (HTML Form):

```
1 <form action="Exf8.py" method="post" target="_blank">
2   <select name="dropdown">
3     <option value="Maths" selected>Maths</option>
4     <option value="Physics">Physics</option>
5   </select>
6   <input type="submit" value="Submit"/>
7 </form>
```

Right Window (Python CGI Script):

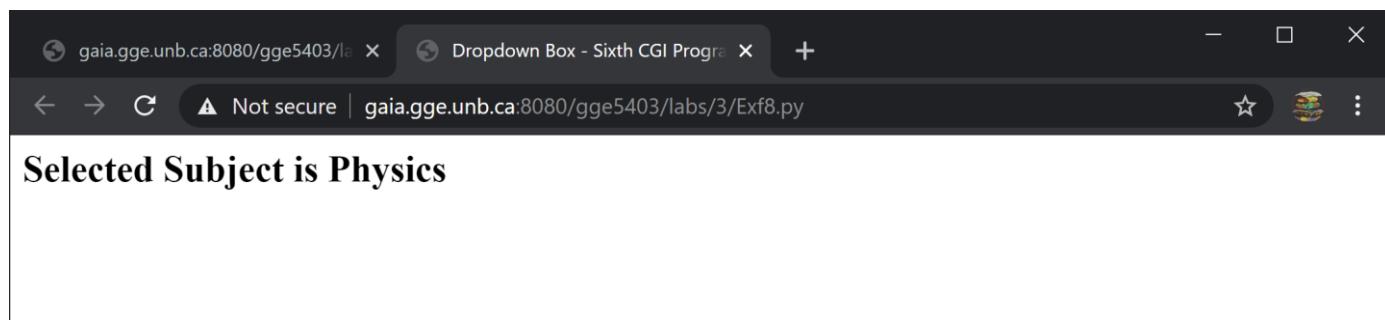
```
1 # !C:\ms4w\Python\python
2
3 # Import modules for CGI handling
4 import cgi, cgitb
5 # Create instance of FieldStorage
6 form = cgi.FieldStorage()
7
8 # Get data from fields
9 if form.getvalue('dropdown'):
10   subject = form.getvalue('dropdown')
11 else:
12   subject = "Not entered"
13
14 print ("Content-type:text/html\r\n\r\n")
15 print ("<html>")
16 print ("<head>")
17 print ("<title>Dropdown Box - Sixth CGI Program</title>")
18 print ("</head>")
19 print ("<body>")
20 print ("<h2> Selected Subject is %s</h2>" % subject)
21 print ("</body>")
22 print ("</html>")
```

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/form8.html>



A screenshot of a web browser window. The address bar shows the URL <http://gaia.gge.unb.ca:8080/gge5403/labs/3/form8.html>. The page content displays a dropdown menu with two options: "Maths" and "Physics". The "Physics" option is highlighted with a blue background. A "Submit" button is located to the right of the dropdown menu.

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Exf8.py>



A screenshot of a web browser window. The address bar shows the URL <http://gaia.gge.unb.ca:8080/gge5403/labs/3/Exf8.py>. The page content displays the text "Selected Subject is Physics" in bold black font.

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/form9.html>

The image shows two side-by-side code editors. The left editor contains an HTML file named 'form9.html' with the following content:

```
1 <html>
2 <body>
3   <form enctype="multipart/form-data"
4     action="Exf9.py" method="post">
5     <p>File: <input type="file" name="filename" /></p>
6     <p><input type="submit" value="Upload" /></p>
7   </form>
8 </body>
9 </html>
```

The right editor contains a Python file named 'Exf9.py' with the following content:

```
1 #!C:\ms4w\Python\python
2 import cgi, os
3 import cgitb; cgitb.enable()
4
5 form = cgi.FieldStorage()
6
7 # Get filename here.
8 fileitem = form['filename']
9
10 # Test if the file was uploaded
11 if fileitem.filename:
12     # strip leading path from file name to avoid
13     # directory traversal attacks
14     fn = os.path.basename(fileitem.filename)
15     open('./tmp/' + fn, 'wb').write(fileitem.file.read())
16
17     message = 'The file "' + fn + '" was uploaded successfully'
18
19 else:
20     message = 'No file was uploaded'
21
22 print ("""
23 Content-Type: text/html\n
24 <html>
25 <body>
26   <p>%s</p>
27 </body>
28 </html>
29 """ % (message,))
```

Both editors have a toolbar at the top with various icons, and status bars at the bottom indicating file length, line count, column count, and encoding.

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/form8.html>

A screenshot of a web browser window. The address bar shows the URL <http://gaia.gge.unb.ca:8080/gge5403/labs/3/form9.html>. Below the address bar, the page title is "Not secure | gaia.gge.unb.ca:8080/gge5403/labs/3/form9.html". The main content area contains a file input field with the placeholder "Choose File" and the file name "points.csv", followed by a "Upload" button.

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Exf9.py>

The file "points.csv" was uploaded successfully

A screenshot of a web browser window titled "Index of /gge5403/labs/3/tmp". The address bar shows the URL <http://gaia.gge.unb.ca:8080/gge5403/labs/3/tmp/>. The page displays a table with the following data:

	Name	Last modified	Size	Description
	Parent Directory		-	
	points.csv	2021-04-10 21:17	66	

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/form10.html>

The image shows two side-by-side code editors. The left editor contains an HTML file named 'form10.html' with the following content:

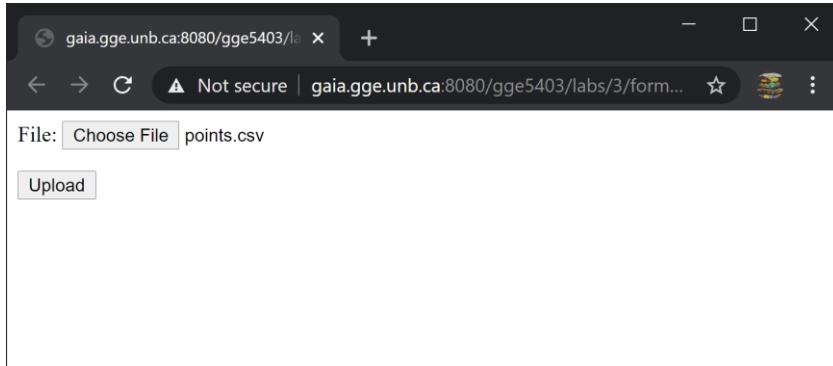
```
1 <html>
2   <body>
3     <form enctype="multipart/form-data"
4       action="Exf10.py" method="post">
5       <p><input type="file" name="filename" /></p>
6       <p><input type="submit" value="Upload" /></p>
7     </form>
8   </body>
9 </html>
```

The right editor contains a Python file named 'Exf10.py' with the following content:

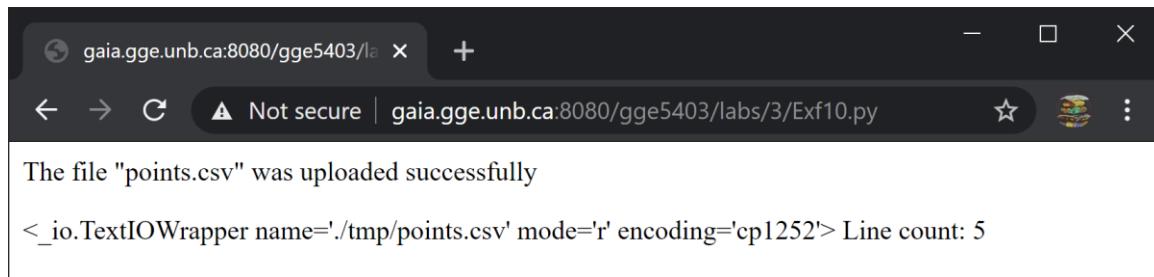
```
1 #!C:\ms4w\Python\python
2 import cgi, os
3 import cgitb; cgitb.enable()
4
5 form = cgi.FieldStorage()
6
7 # Get filename here.
8 fileitem = form['filename']
9
10 # Test if the file was uploaded
11 if fileitem.filename:
12     # strip leading path from file name to avoid
13     # directory traversal attacks
14     fn = os.path.basename(fileitem.filename)
15     open('./tmp/' + fn, 'wb').write(fileitem.file.read())
16
17     message = 'The file "' + fn + '" was uploaded successfully'
18 else:
19     message = 'No file was uploaded'
20
21 print ('''\
22 Content-Type: text/html\n
23 <html>
24   <body>
25     <p>%s</p>
26   </body>
27 </html>
28 ''' % (message,))
29
30 #Read file - count #rows
31 fin = open('./tmp/' + fn, 'r')
32 print(fin)
33 count=0
34 for line in fin:
35     count = count+1
36 print('Line count:', count)
37 fin.close()
```

Both editors have a toolbar at the top with various icons. The bottom status bar of each editor shows the file length, line count, column, position, encoding, and character set.

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/form10.html>



<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Exf10.py>



Index of /gge5403/labs/3/tmp				
	ICO	Name	Last modified	Description
	[PARENTDIR]	Parent Directory		
	[TXT]	points.csv	2021-04-10 21:22	66

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/form11.html>

C:\Users\Emmanuel\Desktop\GGE5403\CONTENT\Module 3 - Web Services\code1\form11.html - Notepad++

```
<html>
<body>
<form enctype="multipart/form-data" action="Exfil.py" method="post">
<p>CSV File of Cities [Name, lat, long]: <input type="file" name="filename" /></p>
<p><input type="submit" value="Upload" /></p>
</form>
</body>
</html>
```

Length : 277 Lines : 9 Ln : 9 Col : 8 Pos : 278 Windows (CR LF) UTF-8 IN

C:\Users\Emmanuel\Desktop\GGE5403\CONTENT\... - Notepad++

```
Fredericton, 45.96, -66.64
Thunder Bay, 48.38, -89.25
Calgary, 51.05, -114.07
```

Ln : 3 Col : 23 Pos : 80 Windows (CR LF) UTF-8 IN

C:\Users\Emmanuel\Desktop\GGE5403\CONTENT\Module 3 - Web Services\code... - Notepad++

```
form = cgi.FieldStorage()
# Get filename here.
fileitem = form['filename']

# Test if the file was uploaded
if fileitem.filename:
    # strip leading path from file name to avoid
    # directory traversal attacks
    fn = os.path.basename(fileitem.filename)
    open('.tmp/' + fn, 'wb').write(fileitem.file.read())
    message = 'The file "' + fn + '" was uploaded successfully'
else:
    message = 'No file was uploaded'

print """
Content-Type: text/html\n
<html>
<body>
<p>%s</p>
</body>
</html>
""" % (message,)
```

length : 1,274 Ln : 42 Col : 53 Pos : 958 Windows (CR LF) UTF-8 IN

C:\Users\Emmanuel\Desktop\GGE5403\CONTENT\Module 3 - Web Services\code1\Exf11.py - Notepad++

```
"""
% (message,)

# Open and Read the file - count #rows and compute the avg lat and lng of the cities.
fin = open('.tmp/' + fn, 'r')
count=0
sumlat=0.
sumlng=0.
print('Input data')
print ("<br>") # New line
for line in fin:
    print(line)
    print ("<br>")
    count = count+1
    # Splits at ',' word[0] will be the city name; word[1] the lat; and word[2] the lng
    word = line.split(',')
    sumlat += float(word[1])
    sumlng += float(word[2])
print ("<br>")
print('...Number of cities: ', count)
print ("<br>")
print("AVG lat = %.2f" % (sumlat/count))
print ("<br>")
print("AVG lng = %.2f" % (sumlng/count))
fin.close()
```

Pylength : 1,274 Lines : 52 Ln : 42 Col : 53 Pos : 958 Windows (CR LF) UTF-8 IN

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/form11.html>

The left screenshot shows a browser window with the URL <http://gaia.gge.unb.ca:8080/gge5403/labs/3/form11.html>. The page title is "Not secure | gaia.gge.unb.ca 8080/gge5403/labs/3/form11.html". A form field labeled "CSV File of Cities [Name, lat, long]:" contains the text "Choose File cities.csv". Below it is a "Upload" button.

The right screenshot shows a code editor window titled "cities.csv" with the following content:

```
1 Fredericton, 45.96, -66.64
2 Thunder Bay, 48.38, -89.25
3 Calgary, 51.05, -114.07
4
```

Below the code editor, status information includes "Ln : 3 Col : 23 Pos : 80 Windows (CR LF) UTF-8 IN".

<http://gaia.gge.unb.ca:8080/gge5403/labs/3/Exf11.py>

The browser window title is "Not secure | gaia.gge.unb.ca:8080/gge5403/labs/3/Exf11.py". The page displays the message "The file "cities.csv" was uploaded successfully".

The "Input data" section shows the following lines of text:

```
Fredericton, 45.96, -66.64
Thunder Bay, 48.38, -89.25
Calgary, 51.05, -114.07
```

The "Output" section shows the following results:

```
...Number of cities: 3
AVG lat = 48.46
AVG lng = -89.99
```

References

- Python CGI Programming
 - https://www.tutorialspoint.com/python/python_cgi_programming.htm
- Configuring Apache Web Server to Run Python on Windows
 - http://editrocket.com/articles/python_apache_windows.html