Modules

- When a Python program starts it only has access to a basic functions and classes.

  (“int”, “dict”, “len”, “sum”, “range”, ...)
- “Modules” contain additional functionality.
- Use “import” to tell Python to load a module.

```
>>> import math
>>> import nltk
```

import the math module

```
>>> import math
>>> math.pi
3.141592653589793l
>>> math.cos(0)
1.0
>>> math.cos(math.pi)
-1.0
>>> dir(math)
['__doc__', '__file__', '__name__', '__package__', 'acos', 'acosh',
'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos',
'cosh', 'degrees', 'e', 'exp', 'fabs', 'factorial', 'floor', 'fmod',
'frexp', 'fsum', 'hypot', 'isinf', 'isnan', 'ldexp', 'log', 'log10',
'log1p', 'modf', 'pi', 'pow', 'radians', 'sin', 'sinh', 'sqrt', 'tan',
'tanh', 'trunc']
>>> help(math)
>>> help(math.cos)
```

“import” and “from ... import ...”

```
>>> import math
math.cos
>>> from math import cos, pi
cos
>>> from math import *
```

Math commands

Python has useful commands for performing calculations.

Command name	Description
abs(***value***)	absolute value
ceil(***value***)	rounds up
cos(***value***)	cosine, in radians
floor(***value***)	rounds down
log(***value***)	logarithm, base *e*
log10(***value***)	logarithm, base 10
max(***value1***, ***value2***)	larger of two values
min(***value1***, ***value2***)	smaller of two values
round(***value***)	nearest whole number
sin(***value***)	sine, in radians
sqrt(***value***)	square root

Constant	Description
e	2.7182818...
pi	3.1415926...

To use many of these commands, you must write the following at the top of your Python program:

from math import *

Python Libraries for Data Science

NumPy:

- introduces objects for multidimensional arrays and matrices, as well as functions that allow to easily perform advanced mathematical and statistical operations on those objects.

- provides vectorization of mathematical operations on arrays and matrices which significantly improves the performance.

- many other python libraries are built on NumPy.

NumPy Example

```
>>> import numpy as np
>>> mat = np.ones((3,3))
>>> print mat
[[ 1.  1.  1.]
 [ 1.  1.  1.]
 [ 1.  1.  1.]]
>>> mat[1,1] = 5
>>> print mat
[[ 1.  1.  1.]
 [ 1.  5.  1.]
 [ 1.  1.  1.]]
>>> vec = np.array([1, 2, 3])
>>> np.dot(mat, vec)
array([  6.,  14.,   6.])
```

I can rename my module when I import it for convenience

It looks a lot like a list of lists!

Create arrays using `np.array`

Support for various linear algebra operations like dot products

Array Slicing

```
>>> a = np.array([[1, 2, 3], [4, 5, 6]], float)
>>> a[1,:] array([ 4., 5., 6.])
>>> a[:,2] array([ 3., 6.])
>>> a.shape (2, 3)
>>> len(a)
2
```

Convert from array to list

```
>>> a = np.array([1, 2, 3], float)
>>> a.tolist()
[1.0, 2.0, 3.0]
>>> list(a)
[1.0, 2.0, 3.0]
```

Array Mathematics

```
>>> a = np.array([1,2,3], float)
>>> b = np.array([5,2,6], float)
>>> a + b
array([6., 4., 9.])
>>> a – b
array([-4., 0., -3.])
>>> a * b
array([5., 4., 18.])
>>> b / a
array([5., 1., 2.])
>>> a % b
array([1., 0., 3.])
>>> b**a
array([5., 4., 216.])
```

- For more information visit:

  http://www.numpy.org/

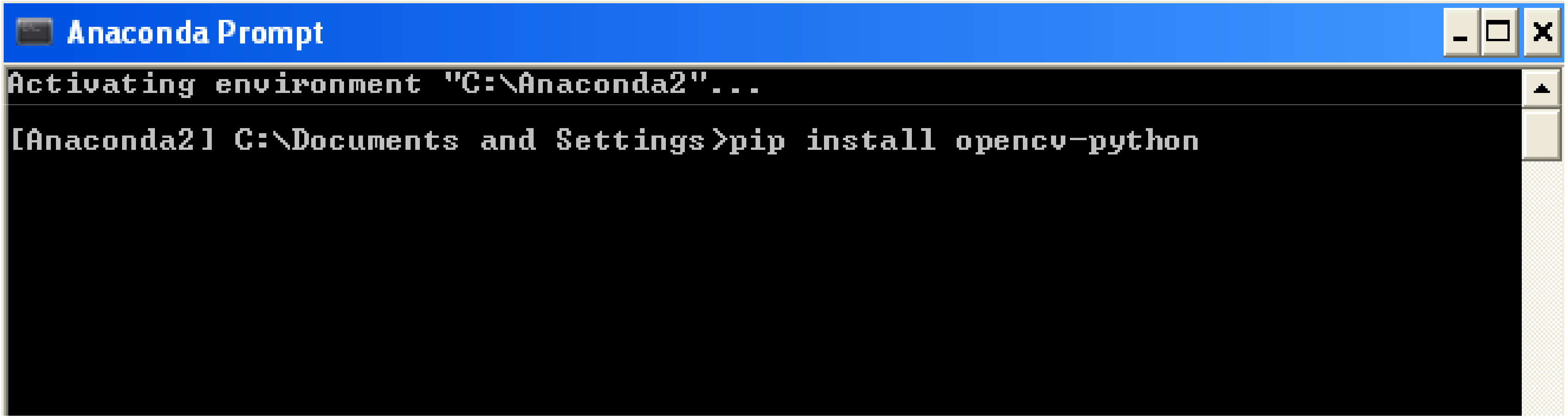
OpenCV

- OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library.
- Started at Intel in 1999 by Gary Bradsky and the first release came out in 2000.
- OpenCV supports a wide variety of programming languages like C++, Python, etc and is available on different platforms including Windows, Linux, OS X, Android, iOS etc. Also, interfaces based on CUDA and OpenCL are also under active development for high-speed GPU operations.

Installation

- For window users Start>All Programs>Anaconda>Anaconda Prompt

pip install opencv-python



```
Activating environment "C:\Anaconda2"...

[Anaconda2] C:\Documents and Settings>pip install opencv-python
```

Test your OpenCV installation on the python console:
> import cv2

Displaying an image

```
import cv2
import numpy as np
img =cv2.imread('watch.jpg',cv2.IMREAD_GRAYSCALE)
cv2.imshow('image',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Matplotlib

Matplotlib is a plotting library for Python which gives you wide variety of plotting methods. You can zoom images, save it etc using Matplotlib.

```
import numpy as np
import cv2 from matplotlib
import pyplot as plt
img = cv2.imread('messi5.jpg’)
plt.imshow(img)
```

Playing with Videos

```
import numpy as np
import cv2
cap = cv2.VideoCapture('vtest.avi')
while(cap.isOpened()):
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    cv2.imshow('frame',gray)
    if (cv2.waitKey(1) & 0xFF == ord('q')):
        break
cap.release()
cv2.destroyAllWindows()
```

Writing an Image

- cv2.imwrite('messigray.png',img)

- For more information visit:

https://opencv.org/about.html

PyLas

- Libraries to read LAS/LAZ in Python

- Installation:

  For window users Start>All Programs>Anaconda>Anaconda Prompt

```
pip install pylas
```

Reading a Lidar Image

```
from laspy.file import File
import numpy as np

source=" ../myfile.las"
las = File(source, mode="r")    #Reading of las file
mmmin = las.header.min         #Return[Longmin,Latmin,Zmin]
mmmax = las.header.max         #Return[Longmax,Latmax,Zmax]

for x, y, z, ite, c,nr,rn in np.nditer([las.x, las.y, las.z, las.Intensity, las.Classification, las.num_returns,las.return_num]):

        print(" Longitude: " ,x)
        print(" Latitude: " ,y)
        print(" Altitude: " ,z)
        print(" Intensity value: " ,ite)
        print(" Classification: " ,c)
        print(" Number of Returns: " ,nr)
        print(" Return Number: " ,rn)
```

Spectral Python (SPy)

- Spectral Python (SPy) is a pure Python module for processing hyperspectral image data.
- It has functions for reading, displaying, manipulating, and classifying hyperspectral imagery.

Installation

- For window users Start>All Programs>Anaconda>Anaconda Prompt

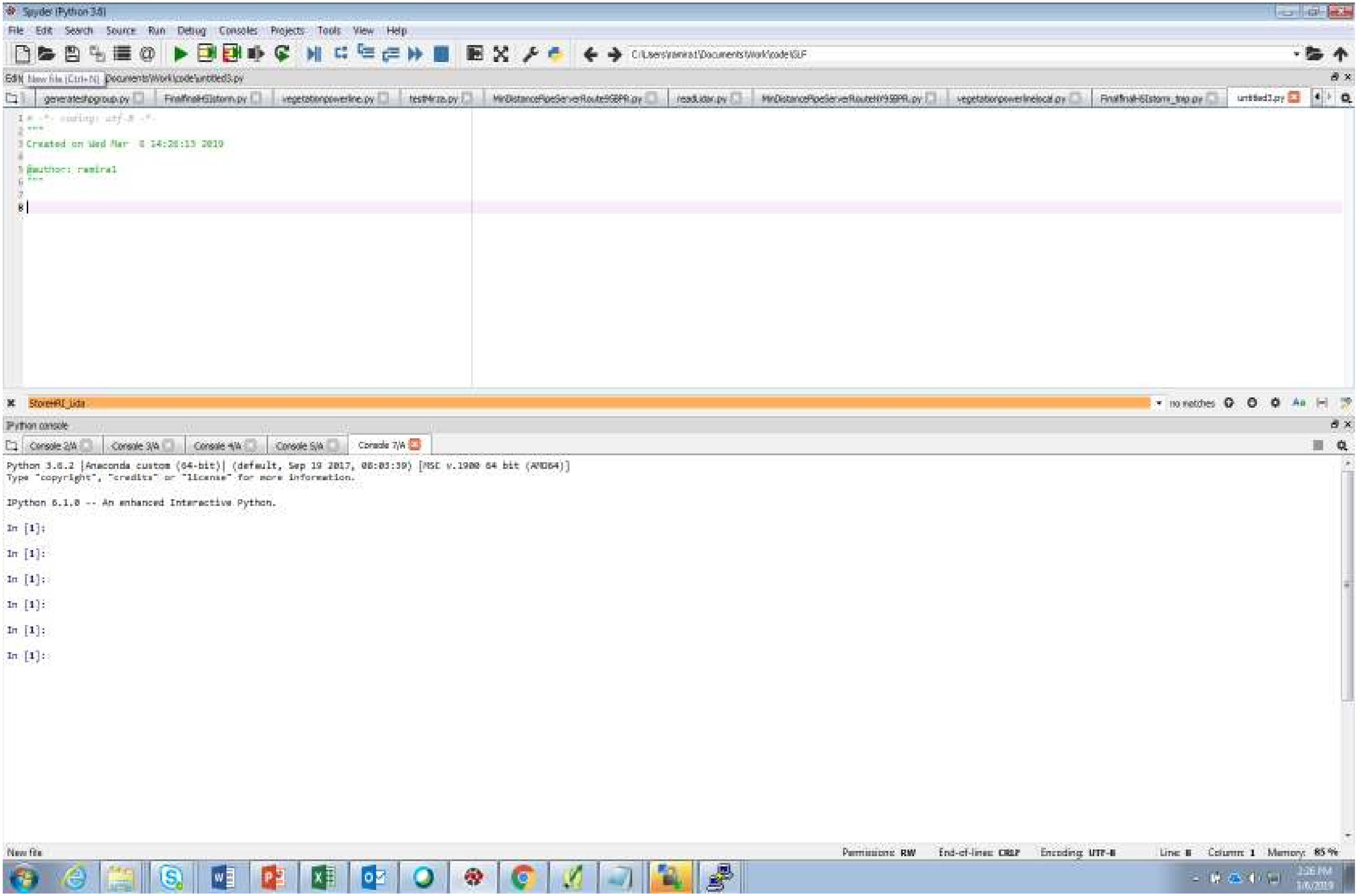
  pip install install spectral

Test your Spy installation on the python console:

```
> >import spectral
```

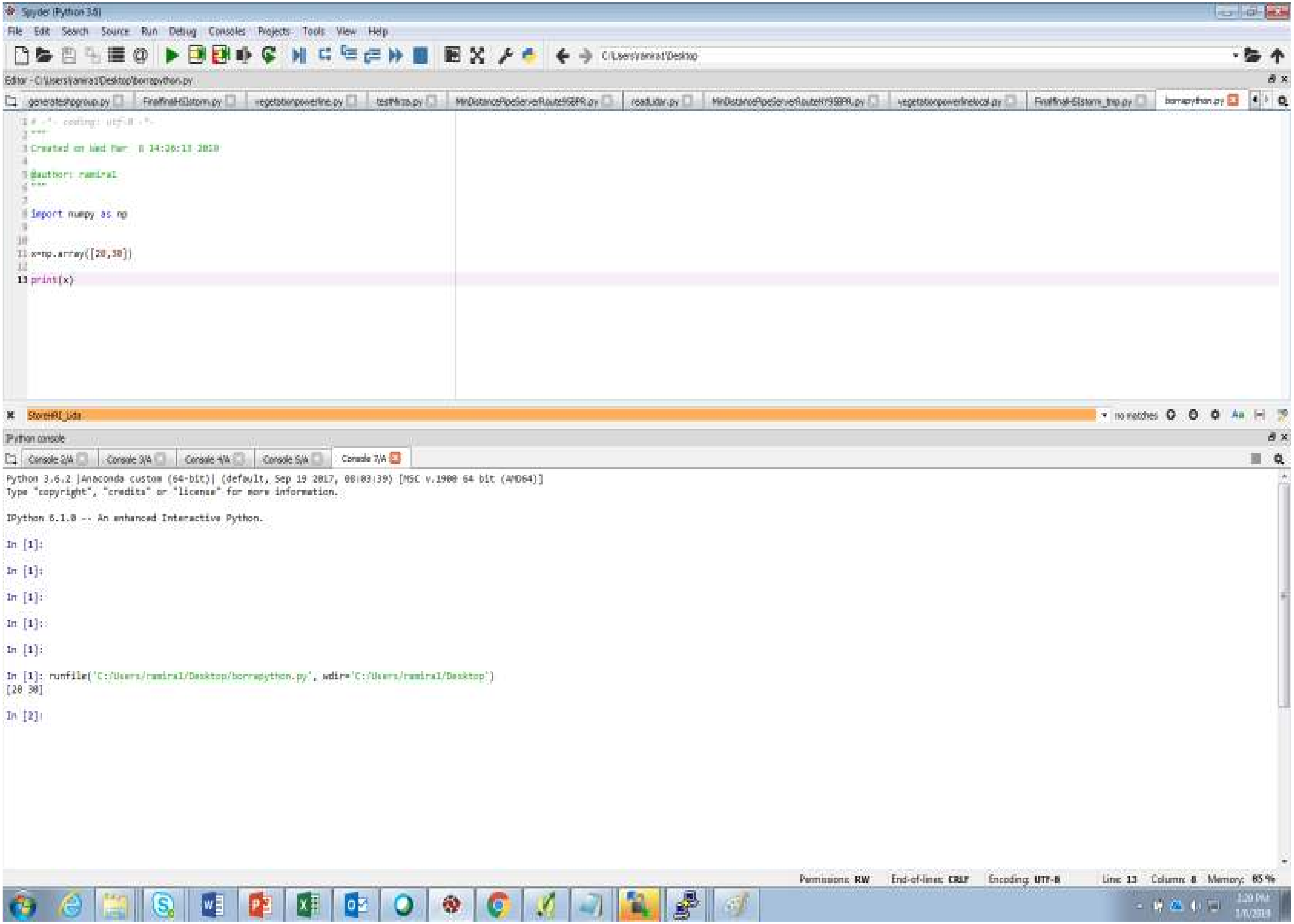
Reading a Hyperspectral Image

```
>>from spectral import *
>>img = open_image('92AV3C.lan')
>>img.__class__
spectral.io.bilfile.BilFile
>>print (img)
Data Source:  '/home/thomas/spectral_data/92AV3C.lan'
# Rows: 145
# Samples: 145
# Bands: 220
Interleave: BIL
Quantization: 16 bits
Data format: int16
```



```
# -*- coding: utf-8 -*-
"""
Created on Wed Mar  6 14:26:13 2019

@author: ramira1
"""

```

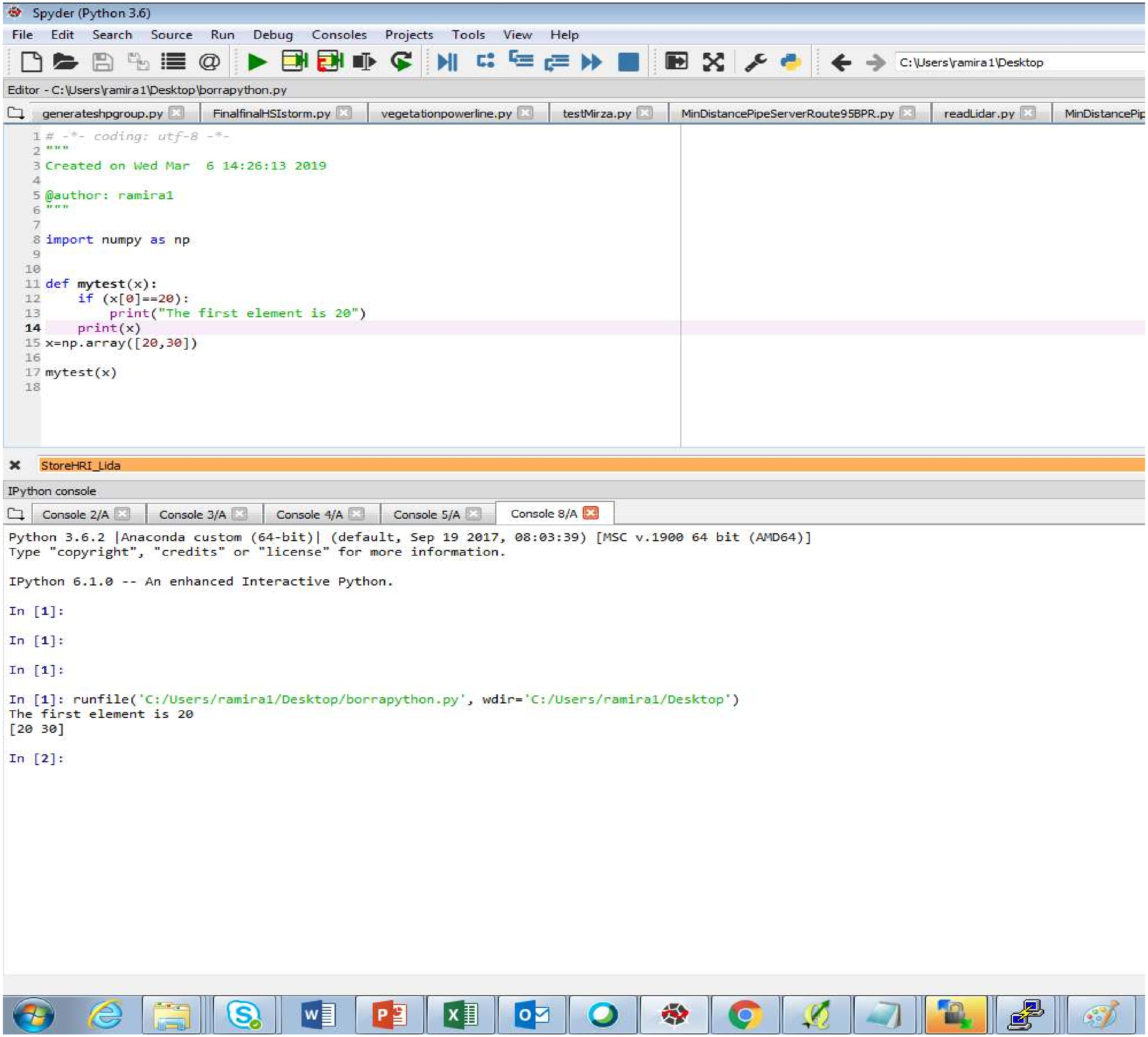
```
Python 3.6.2 |Anaconda custom (64-bit)| (default, Sep 19 2017, 08:03:39) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 6.1.0 -- An enhanced Interactive Python.

In [1]:

In [1]:

In [1]:

In [1]:

In [1]:

In [1]:
```



```python
# -*- coding: utf-8 -*-
"""
Created on Wed Mar  8 14:26:13 2019

@author: ramiral
"""

import numpy as np


x=np.array([20,30])

print(x)
```

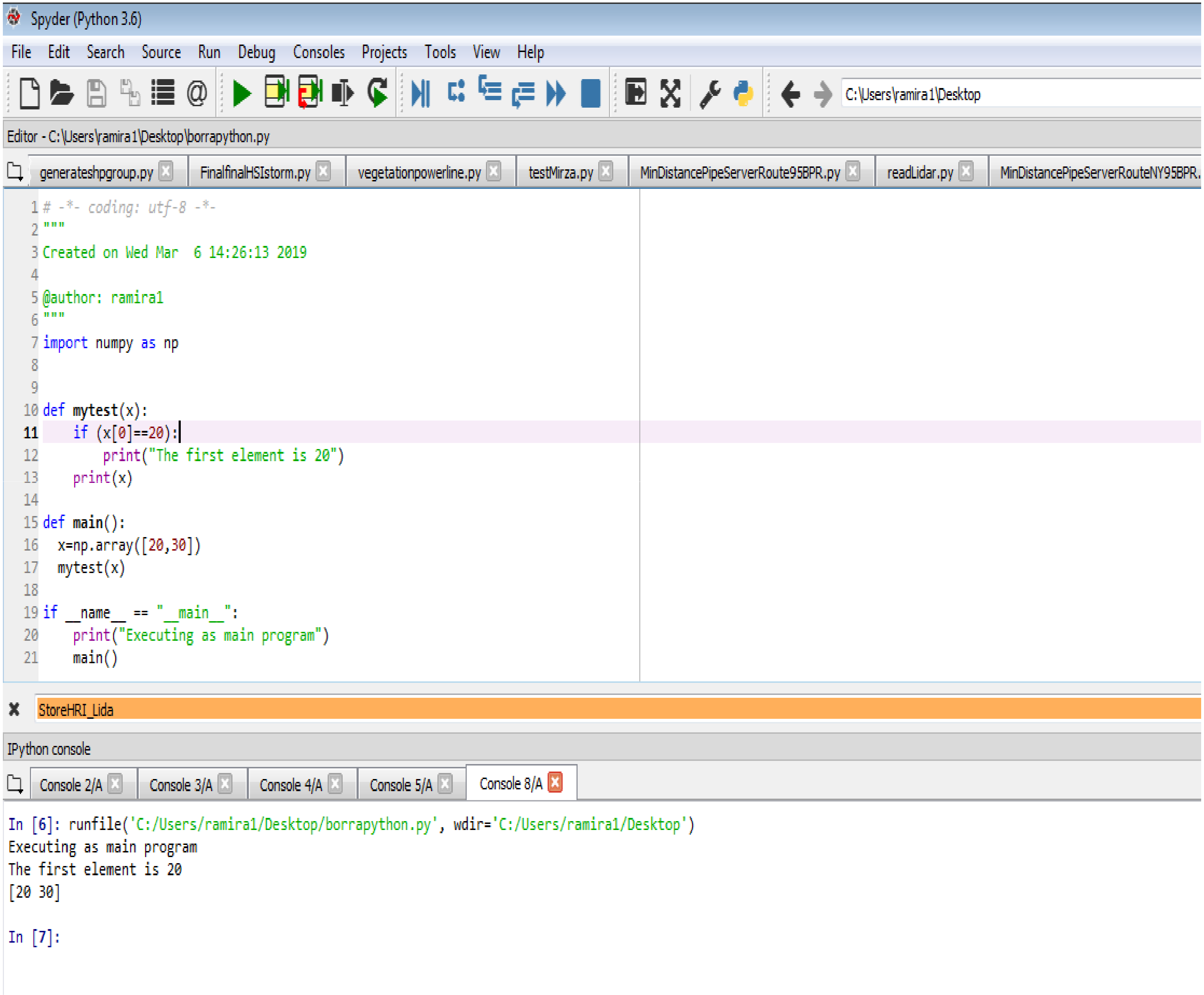
```
Python 3.6.2 |Anaconda custom (64-bit)| (default, Sep 19 2017, 08:03:39) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 6.1.0 -- An enhanced Interactive Python.

In [1]:

In [1]:

In [1]:

In [1]:

In [1]:

In [1]: runfile('C:/Users/ramiral/Desktop/borrapython.py', wdir='C:/Users/ramiral/Desktop')
[20 30]

In [2]:
```

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 13 Column: 8 Memory: 85 %



```python
# -*- coding: utf-8 -*-
"""
Created on Wed Mar  6 14:26:13 2019

@author: ramira1
"""

import numpy as np


def mytest(x):
    if (x[0]==20):
        print("The first element is 20")
    print(x)
x=np.array([20,30])

mytest(x)
```

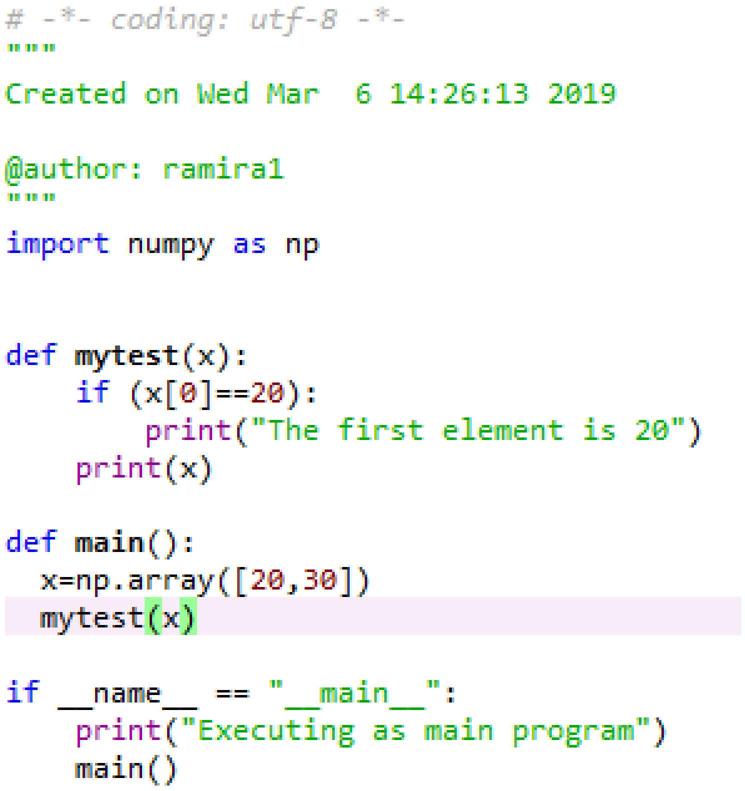
```
Python 3.6.2 |Anaconda custom (64-bit)| (default, Sep 19 2017, 08:03:39) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 6.1.0 -- An enhanced Interactive Python.

In [1]:

In [1]:

In [1]:

In [1]: runfile('C:/Users/ramira1/Desktop/borrapython.py', wdir='C:/Users/ramira1/Desktop')
The first element is 20
[20 30]

In [2]:
```



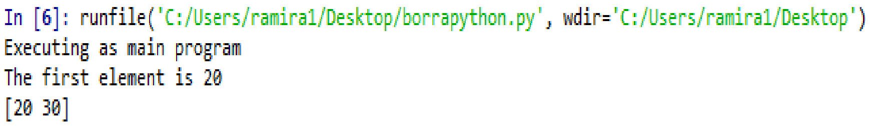
```python
# -*- coding: utf-8 -*-
"""
Created on Wed Mar  6 14:26:13 2019

@author: ramira1
"""
import numpy as np


def mytest(x):
    if (x[0]==20):
        print("The first element is 20")
    print(x)

def main():
  x=np.array([20,30])
  mytest(x)

if __name__ == "__main__":
    print("Executing as main program")
    main()
```

```
In [6]: runfile('C:/Users/ramira1/Desktop/borrapython.py', wdir='C:/Users/ramira1/Desktop')
Executing as main program
The first element is 20
[20 30]

In [7]:
```

Firstscript.py

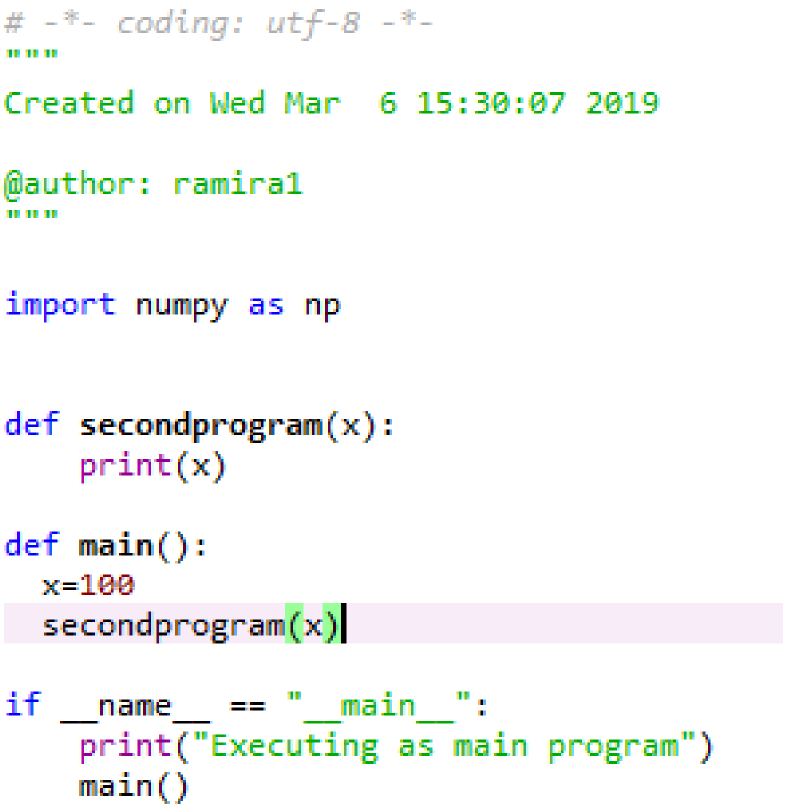


```python
# -*- coding: utf-8 -*-
"""
Created on Wed Mar  6 14:26:13 2019

@author: ramira1
"""
import numpy as np


def mytest(x):
    if (x[0]==20):
        print("The first element is 20")
    print(x)

def main():
  x=np.array([20,30])
  mytest(x)

if __name__ == "__main__":
    print("Executing as main program")
    main()
```

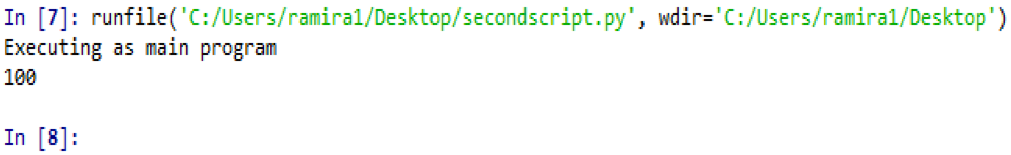


```
In [6]: runfile('C:/Users/ramira1/Desktop/borrapython.py', wdir='C:/Users/ramira1/Desktop')
Executing as main program
The first element is 20
[20 30]
```

Secondscript.py

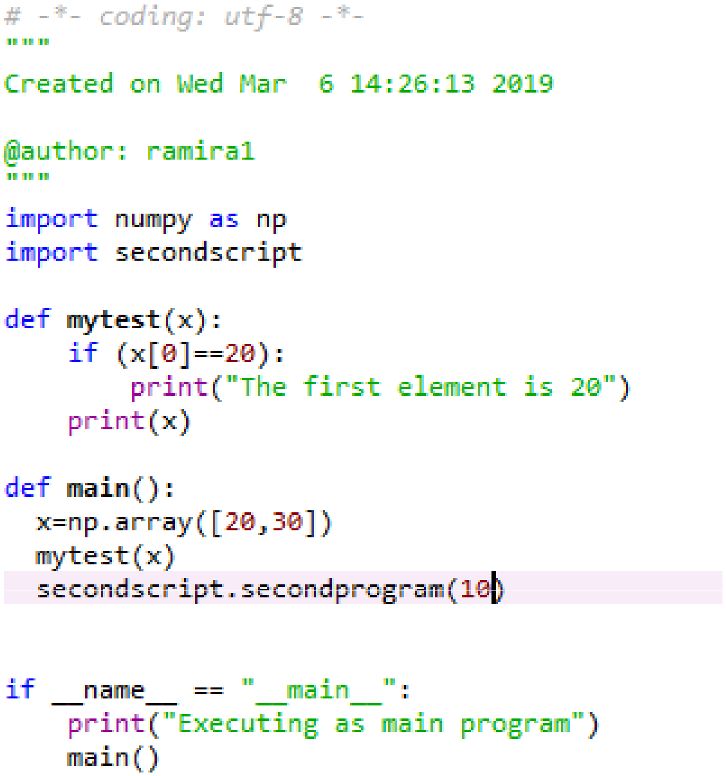


```python
# -*- coding: utf-8 -*-
"""
Created on Wed Mar  6 15:30:07 2019

@author: ramira1
"""

import numpy as np


def secondprogram(x):
    print(x)

def main():
  x=100
  secondprogram(x)

if __name__ == "__main__":
    print("Executing as main program")
    main()
```

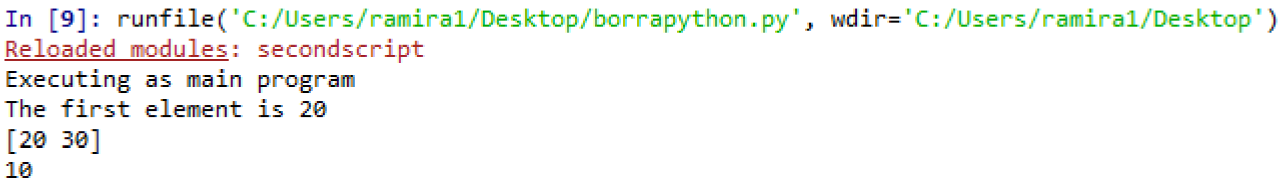


```
In [7]: runfile('C:/Users/ramira1/Desktop/secondscript.py', wdir='C:/Users/ramira1/Desktop')
Executing as main program
100

In [8]:
```

Firstscript.py

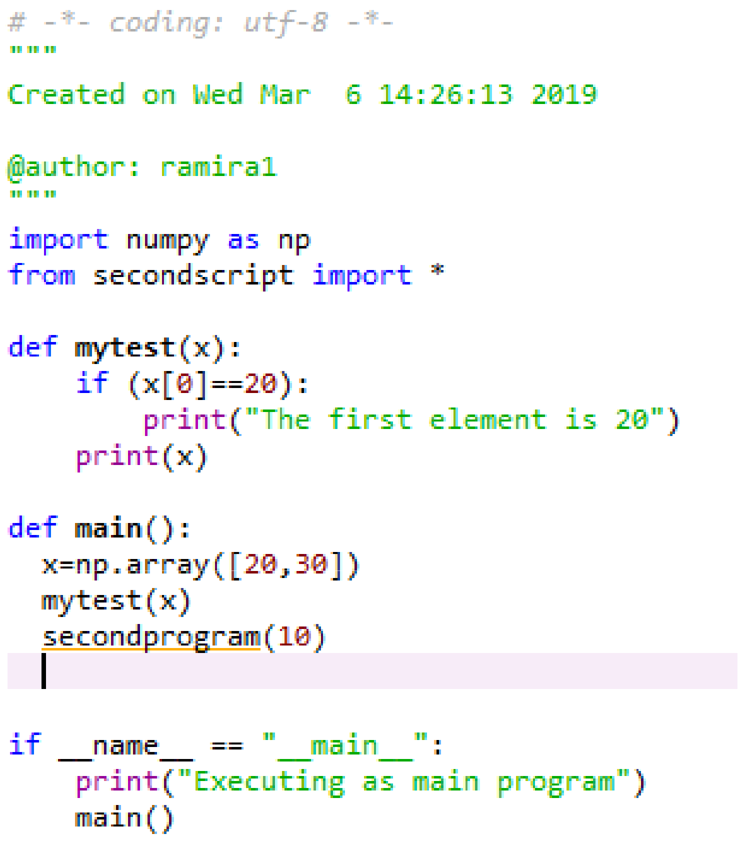


```python
# -*- coding: utf-8 -*-
"""
Created on Wed Mar  6 14:26:13 2019

@author: ramira1
"""
import numpy as np
import secondscript

def mytest(x):
    if (x[0]==20):
        print("The first element is 20")
    print(x)

def main():
  x=np.array([20,30])
  mytest(x)
  secondscript.secondprogram(10)


if __name__ == "__main__":
    print("Executing as main program")
    main()
```

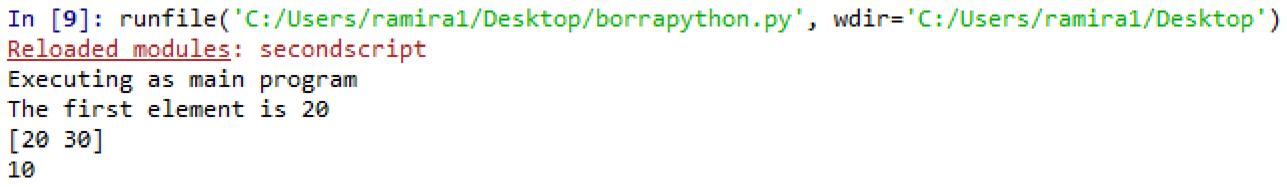


```
In [9]: runfile('C:/Users/ramira1/Desktop/borrapython.py', wdir='C:/Users/ramira1/Desktop')
Reloaded modules: secondscript
Executing as main program
The first element is 20
[20 30]
10
```

Firstscript.py



```python
# -*- coding: utf-8 -*-
"""
Created on Wed Mar  6 14:26:13 2019

@author: ramira1
"""
import numpy as np
from secondscript import *

def mytest(x):
    if (x[0]==20):
        print("The first element is 20")
    print(x)

def main():
  x=np.array([20,30])
  mytest(x)
  secondprogram(10)
  

if __name__ == "__main__":
    print("Executing as main program")
    main()
```



```
In [9]: runfile('C:/Users/ramira1/Desktop/borrapython.py', wdir='C:/Users/ramira1/Desktop')
Reloaded modules: secondscript
Executing as main program
The first element is 20
[20 30]
10
```