

BBM469 - Data Intensive Applications Laboratory

Exercise : Introduction to Python, Numpy, and Pandas
Date : 21.02.2022

The Aim

The aim of this exercise is to get you acquainted with basic Python programming and some key libraries used in data processing.

Useful Links

Python basics

- <https://web.cs.hacettepe.edu.tr/~bbm101/>
- <https://github.com/jerry-git/learn-python3>

Pandas, Numpy

- <https://cloudxlab.com/blog/numpy-pandas-introduction/>
- <https://data36.com/pandas-tutorial-1-basics-reading-data-files-dataframes-data-selection/>
- <http://pandas.pydata.org/pandas-docs/version/0.15.2/10min.html>

Exercises for Python Basics:

We provide you with some exercises to check your python understanding. You may use Python interpreter, Python tutor (<http://pythontutor.com/>), pycharm or Jupyter Notebook to run these examples. We recommend Jupyter Notebooks.

- Basic types: Numbers, Strings

```
In [1]:  
str1 = 'spam eggs'a, b  
str2 = "spam eggs"  
str3 = '"Isn\'t," they said.'  
  
a, b = 1, 2  
a, b  
  
b, a = a, b  
a, b
```

```

# Integer
a = 1
print(a)

# Float
b = 1.0
print(b)

# String
c = "Hello world"
print(c)

```

- Collections: Lists, Tuples, Dictionaries, Sets

```

In [1]:
# List (array)
e = [1, 2, 3]
print(e[2]) # 3

a = [ 3, 1, 2*2, 1, 10/2, 10-1 ]
a
b = [ 5, 3, 'hi' ]
b
c = [ 4, 'a', a ]
c
a = [3, 4, 5]
a

L = ['I did it all', 4, 'love']
for i in range(len(L)):
    print(L[i])

Techs = ['MIT', 'Caltech']
Ivys = ['Harvard', 'Yale', 'Brown']
Univs = [Techs, Ivys]
Univs1 = [['MIT', 'Caltech'], ['Harvard', 'Yale', 'Brown']]
print('Univs =', Univs)
print('Univs1 =', Univs1)
print(Univs == Univs1)

In [2]:

#list functions
'MIT' in Techs

Techs.index('MIT')
Techs.count('MIT')
Techs.append('MIT')
Techs.extend(Ivys)
Techs.insert(1, 'HUBBM')
Techs.remove('MIT')

```

```

Techs.remove('MIT')
Techs.remove('MIT')
un=Techs.pop(0)
un
Techs[0] = 'MIT'
Techs.sort()
Techs.reverse()

#list slicing
test_list = ['e0', 'e1', 'e2', 'e3', 'e4', 'e5', 'e6']
test_list[2:]
test_list[:5]
test_list[-1]
test_list[-4:]
test_list[:-3]
test_list[::-1]

```

In [3]:

```

# Tuple (constant array)
f = (1, 2, 3)
print(f[0])

```

In [4]:

```

# Set
g = {1, 1, 1, 2}
print(g)
odd = set([1, 3, 5])
prime = set([2, 3, 5])
empty = set([])

k = odd & prime
j = odd | prime
m = odd - prime
odd.add(9)

```

In [5]:

```

# Dictionary (hash table, hash map)
g = {1: 'One', 2: 'Two', 3: 'Three'}
print(g[1]) # 'One'

atomicnumber = {"H":1, "Fe":26, "Au":79}
# Print out all the keys:
for element_name in atomicnumber.keys():
    print(element_name)

# Another way to print out all the keys:
for element_name in atomicnumber:
    print(element_name)

```

```
# Print out the keys and the values
for (element_name, element_number) in atomicnumber.items():
    print("name:" ,element_name, "number:" ,element_number)

atomicnumber.pop("H")
atomicnumber["O"] = 16
```

- Iterations and Control Flow

In [1]:

```
for i in range(10):
    print(i)

print(i)

for i in range(0,10,2):
    print(i)

for i in range(10,0,-2):
    print(i)

for f in [30,40,50,60,70]:
    print(f, (f-32)/9.0*5)

for letter in 'hollywood':
    if letter == 'l':
        break
    print ('Current Letter :', letter)

for i in [0,1]:
    print("Outer", i)
    for j in [2,3]:
        print(" Inner", j)
        print(" Sum", i+j)
    print("Outer", i)
```

In [2]:

```
i = 0
while i < 10:
    print(i)
    i += 1
```

```
i = 0
while i < 10:
    print(i)
    for j in range(0,10,2):
```

```
        print(j)

items = ['apple', 'banana', 'stawberry', 'watermelon']
for item in items:
    print(item)
```

```
for i, item in enumerate(items):
    print(i, item)
```

In [3]:

```
val = -10
if val < 0:
    result = - val
else:
    result = val
print(result)

height = 75
if height > 100:
    print("space")
elif height > 50:
    print("mesosphere")
elif height > 20:
    print("stratosphere")
else:
    print("troposphere")
```

- Functions

In [1]:

```
def square(x):
    return x * x

def fahr_to_cent(fahr):
    return (fahr - 32) / 9.0 * 5

def cent_to_fahr(cent):
    result = cent / 5.0 * 9 + 32
    return result

def print_fahr_to_cent(fahr):
    result = fahr_to_cent(fahr)
    print(result)

print(print_fahr_to_cent(32))
square(3) + square(4)
boiling = fahr_to_cent(212)
```

```
cold = cent_to_fahr(-40)
```

```
In [2]:
```

```
import math

def double(x):
    return 2*x
print(double)

myfns = [math.sqrt, int, double, math.cos]
myfns[1](3.14)
myfns[2](3.14)
myfns[3](3.14)

def doubler():
    return double
doubler()(2.718)

def f(x):
    def g():
        x = "abc"
        print("x =", x)
    def h():
        z = x
        print("z =", z)
    x = x+1
    print("x =", x)
    h()
    g()
    print("x =", x)
    return g

x = 3
z = f(x)
print("x =", x)
print("z =", z)
z()
```

- File I/O: (Use pycharm or python interpreter for this examples)

```
In [1]:
```

```
# Count the number of words in a text file
in_file = "thesis.txt"
myfile = open(in_file)
num_words = 0
for line_of_text in myfile:
    word_list = line_of_text.split()
    num_words += len(word_list)
myfile.close()

print("Total words in file: ", num_words)
```

```

nameHandle = open('characters.txt', 'w')
for i in range(2):
    name = input('Enter name: ')
    nameHandle.write(name + '\n')
nameHandle.close()

nameHandle = open('characters.txt', 'r')
for line in nameHandle:
    print(line)
nameHandle.close()

```

- Numpy, Pandas:

```

In [1]:

import numpy
numpy.ones((2, 3))

a = numpy.array([[1,2,3], [4,5,6], [7,8,9], [10, 11, 12]])

a.shape
a
a.ndim
a
a.size
a - numpy.random.random(a.shape)

a.ravel()
a[1:-1]

a[:,1]
a[a % 2 == 0] = -1

In [2]:

import pandas as pd

s = pd.Series([1,3,5,np.nan,6,8])
s

dates = pd.date_range('20130101',periods=6)
dates

```

```

df =
pd.DataFrame(np.random.randn(6,4),index=dates,columns=list('
ABCD'))
df

df2 = pd.DataFrame({'A' : 1.,
                    'B' : pd.Timestamp('20130102'),
                    'C' :
pd.Series(1,index=list(range(4)),dtype='float32'),
                    'D' : np.array([3] * 4,dtype='int32'),
                    'E' :
pd.Categorical(["test","train","test","train"]),
                    'F' : 'foo' })

df2
df2.dtypes

df.head()
df.tail(3)
df.columns
df.values

df.describe()
df.sort_index(axis=1, ascending=False)

df['A']
df[0:3]

```

Grading

You will take no grade for this take home assignment. By completing this assignment, you will remember or get acquainted with some Python structures.

- Basics of Python
- Datatypes
- Functions
- Control-Flow
- Loops
- File I/O
- Lists
- Tuples
- Dictionaries
- Sets
- Numpy
- Pandas