Information:

Name: Nguyễn Thanh Lộc

ID: 19521763

Link Github: https://github.com/talo33/Data-mining-Lab1.git

1. Data types:

a. Number:

```
Entrée [221]: 1 + 1
   Out[221]: 2

Entrée [222]: 1 * 3
   Out[222]: 3

Entrée [223]: 1 / 2
   Out[223]: 0.5

Entrée [224]: 2 ** 4
   Out[224]: 16

Entrée [225]: 4 % 2
   Out[225]: 0

Entrée [226]: 5 % 2
   Out[226]: 1

Entrée [227]: (2+3) *(5+5)
   Out[227]: 50
```

b. Variable assignment

```
Entrée [10]: name_of_var=2

Entrée [11]: x=2
             y=3

Entrée [12]: z=x+y

Entrée [13]: z
    Out[13]: 5
```

c. Strings

```
Entrée [228]: 'single quotes'
   Out[228]: 'single quotes'

Entrée [229]: 'double quotes'
   Out[229]: 'double quotes'

Entrée [230]: "wrap lot's of other quotes"
   Out[230]: "wrap lot's of other quotes"
```

d. Printing

```
Entrée [231]: x = 'Hello'

Entrée [232]: x
   Out[232]: 'Hello'

Entrée [233]: print(x)
              Hello

Entrée [234]: num = 12
              name = 'Sam'

Entrée [235]: print('My name is:{one}, and my number is: {two}'.format(one= 12, two = 'Sam') )
              My name is:12, and my number is: Sam

Entrée [236]: print('My number is: {}, and my name is: {}'.format(num,name))
              My number is: 12, and my name is: Sam
```

e. Lists

```
Entrée [237]: [1,2,3]
   Out[237]: [1, 2, 3]

Entrée [238]: ['hi',1,[1,2]]
   Out[238]: ['hi', 1, [1, 2]]

Entrée [239]: my_list = ['a','b','c']

Entrée [240]: my_list.append('d')

Entrée [241]: my_list
   Out[241]: ['a', 'b', 'c', 'd']

Entrée [242]: my_list[0]
   Out[242]: 'a'

Entrée [243]: my_list[1]
   Out[243]: 'b'

Entrée [244]: my_list[1:]
   Out[244]: ['b', 'c', 'd']

Entrée [245]: my_list[:1]
   Out[245]: ['a']
```

```
Entrée [245]: my_list[:1]
   Out[245]: ['a']

Entrée [246]: my_list[0]= 'NEW'

Entrée [247]: my_list
   Out[247]: ['NEW', 'b', 'c', 'd']

Entrée [248]: nest = [1,2,3,[4,5,['target']]]

Entrée [249]: nest[3]
   Out[249]: [4, 5, ['target']]

Entrée [250]: nest[3][2]
   Out[250]: ['target']

Entrée [251]: nest[3][2][0]
   Out[251]: 'target'
```

f. Dictionaries

```
Entrée [42]: d={'key1':'item1','key2':'item2'}

Entrée [43]: d
   Out[43]: {'key1': 'item1', 'key2': 'item2'}

Entrée [44]: d['key1']
   Out[44]: 'item1'
```

g. Booleans

```
Entrée [255]: True
   Out[255]: True

Entrée [256]: False
   Out[256]: False
```

h. Tuples

```
Entrée [257]: t = (1,2,3)

Entrée [258]: t[0]
   Out[258]: 1

Entrée [259]: t1=list(t)
              t1[0]= 'NEW'
              t=list(t1)
```

i. Sets

```
Entrée [260]: {1,2,3}
   Out[260]: {1, 2, 3}

Entrée [261]: {1,2,3,1,2,1,2,3,3,3,3,2,2,2,1,1,2}
   Out[261]: {1, 2, 3}
```

j. Comparison Operators

```
Entrée [262]: 1>2
```
```
Out[262]: False
```
```
Entrée [263]: 1<2
```
```
Out[263]: True
```
```
Entrée [264]: 1 >=1
```
```
Out[264]: True
```
```
Entrée [265]: 1 <=4
```
```
Out[265]: True
```
```
Entrée [266]: 1 == 1
```
```
Out[266]: True
```
```
Entrée [267]: 'hi' == 'bye'
```
```
Out[267]: False
```

k. Logic Operators

```
Entrée [268]: (1>2) and (2<3)
```
```
Out[268]: False
```
```
Entrée [269]: (1>2) or (2<3)
```
```
Out[269]: True
```
```
Entrée [270]: (1==2) or (2==3) or (4==4)
```
```
Out[270]: True
```

l. If, elseif, else Statement

```
Entrée [271]: if 1 < 2:
                print('Yeb!')
```
```
Yeb!
```
```
Entrée [272]: if 1 < 2:
                print('yeb!')
```
```
yeb!
```
```
Entrée [273]: if 1 < 2:
                print('first')
              else:
                  print('last')
```
```
first
```
```
Entrée [274]: if 1 == 2:
                print('first')
              elif 3 == 3:
                  print('midle')
              else:
                print('Last')
```
```
midle
```

m. For Loops

Entrée [275]:
```
seq = [1,2,3,4,5]
```

Entrée [276]:
```
for item in seq:
 print(item)
```

```
1
2
3
4
5
```

Entrée [277]:
```
for item in seq:
  print('yeb')
```

```
yeb
yeb
yeb
yeb
yeb
```

Entrée [278]:
```
for jelly in seq:
  print(jelly+jelly)
```

```
2
4
6
8
10
```

n. While Loops

Entrée [279]:
```
i = 1
while i < 5:
  print('i is: {}'.format(i))
  i = i +1
```

```
i is: 1
i is: 2
i is: 3
i is: 4
```

o. Range

Entrée [280]:
```
range(5)
```

Out[280]:
```
range(0, 5)
```

Entrée [281]:
```
for i in range(5):
  print(i)
```

```
0
1
2
3
4
```

Entrée [282]:
```
list(range(5))
```

Out[282]:
```
[0, 1, 2, 3, 4]
```

p. List Comprehension

Entrée [283]:
```
x = [1,2,3,4]
```

Entrée [284]:
```
out = []
for item in x:
  out.append(item**2)
  print(out)
```

```
[1]
[1, 4]
[1, 4, 9]
[1, 4, 9, 16]
```

Entrée [285]:
```
[item**2 for item in x]
```

Out[285]:
```
[1, 4, 9, 16]
```

q. Functions

Entrée [286]:
```
def my_func(param1='default'):
  """
  Docstring goes here.
  """
  print(param1)
```

Entrée [287]:
```
my_func
```

Out[287]:
```
<function __main__.my_func(param1='default')>
```

Entrée [288]:
```
my_func()
```

```
default
```

Entrée [289]:
```
my_func('new param')
```

```
new param
```

Entrée [290]:
```
my_func(param1='new param')
```

```
new param
```

Entrée [291]:
```
def square(x):
  return x**2
```

Entrée [292]:
```
out = square(2)
```

Entrée [293]:
```
print(out)
```

```
4
```

r. Lambda expression

Entrée [294]:
```
def times2(var):
  return var*2
```

Entrée [295]:
```
times2(2)
```

Out[295]:
```
4
```

Entrée [296]:
```
lambda var: var*2
```

Out[296]:
```
<function __main__.<lambda>(var)>
```

s. Map and filter

```
Entrée [297]: seq = [1,2,3,4,5]

Entrée [298]: map(times2,seq)
   Out[298]: <map at 0x7fac67420ee0>

Entrée [299]: list(map(times2,seq))
   Out[299]: [2, 4, 6, 8, 10]

Entrée [300]: list(map(lambda var: var*2,seq))
   Out[300]: [2, 4, 6, 8, 10]

Entrée [301]: filter(lambda item: item%2 == 0,seq)
   Out[301]: <filter at 0x7fac674479d0>

Entrée [302]: list(filter(lambda item: item%2 == 0, seq))
   Out[302]: [2, 4]
```

t. Methods

```
Entrée [303]: st = 'hello my name is Sam'

Entrée [304]: st.lower()
   Out[304]: 'hello my name is sam'

Entrée [305]: st.upper()
   Out[305]: 'HELLO MY NAME IS SAM'

Entrée [306]: st.split()
   Out[306]: ['hello', 'my', 'name', 'is', 'Sam']

Entrée [307]: tweet = 'Go Sport: #Sports'

Entrée [308]: tweet.split('#')
   Out[308]: ['Go Sport: ', 'Sports']

Entrée [309]: tweet.split('#')[1]
   Out[309]: 'Sports'

Entrée [310]: d
   Out[310]: {'key': 'item1', 'key2': 'item2'}

Entrée [311]: d.keys()
   Out[311]: dict_keys(['key', 'key2'])


Entrée [312]: d.items()
   Out[312]: dict_items([('key', 'item1'), ('key2', 'item2')])

Entrée [313]: lst =[1,2,3]

Entrée [314]: lst.pop()
   Out[314]: 3

Entrée [315]: lst
   Out[315]: [1, 2]

Entrée [316]: 'x' in [1,2,3]
   Out[316]: False

Entrée [317]: 'x' in ['x','y','z']
   Out[317]: True
```

IV. Python basics – Exercises

Entrée [318]:
```
7 **4
```
Out[318]: 2401

Entrée [326]:
```
s = "Hi there Sam!"
```

Entrée [327]:
```
t = s.split()
```

Entrée [330]:
```
t1=list(t)
t1[2]= 'dad!'
t=list(t1)
```

Entrée [331]:
```
t
```
Out[331]: ['Hi', 'there', 'dad!']

Entrée [332]:
```
planet = "Earth"
diameter = 12742
```

Entrée [334]:
```
print('The diameter of {}  is {} kilometers'.format(planet,diameter))
```
```
The diameter of Earth  is 12742 kilometers
```

Entrée [350]:
```
lst = [1,2,[3,4],[5,[100,200,['hello']],24,11],1,7]
```

Entrée [353]:
```
lst[3][1][2]
```
Out[353]: ['hello']

Entrée [356]:
```
d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}
```

Entrée [367]:
```
d['k1'][3]['tricky'][3]['target'][3]
```
Out[367]: 'hello'

Entrée [ ]:
```
# Tupe is immutable
```

Entrée [365]:
```
def domainGet(email):
    return email.split('@')[-1]
```

Entrée [366]:
```
domainGet('user@domain.com')
```
Out[366]: 'domain.com'

Entrée [368]:
```
def findDog(st):
    return 'dog' in st.lower().split()
```

Entrée [375]:
```
findDog('Is there a dog here?')
```
Out[375]: True

Entrée [372]:
```
def countDog(st):
    count = 0
    for word in st.lower().split():
        if word == 'dog':
            count += 1
    return count
```

Entrée [376]:
```
countDog('This dog runs faster than the other dog dude!')
```
Out[376]: 2

Entrée [377]:
```
seq = ['soup','dog','salad','cat','great']
```

Entrée [378]:
```
list(filter(lambda word: word[0]=='s',seq))
```
Out[378]: ['soup', 'salad']

Entrée [9]:
```
def caught_speeding(speed, is_birthday):

    if is_birthday:
        speeding = speed - 5
    else:
        speeding = speed

    if speeding > 80:
        return 'Big Ticket'
    else:
        return 'Small Ticket'
```

Entrée [10]:
```
caught_speeding(81,True)
```
Out[10]: 'Small Ticket'

Entrée [ ]:

Entrée [11]:
```
caught_speeding(81,False)
```
Out[11]: 'Big Ticket'

Entrée [10]:
```
caught_speeding(81,True)
```
Out[10]: 'Small Ticket'

Entrée [ ]:

Entrée [11]:
```
caught_speeding(81,False)
```
Out[11]: 'Big Ticket'

Entrée [12]:
```
caught_speeding(40,0)
```
Out[12]: 'Small Ticket'