```
# LAB01 - Introduction
```

```
## Nguyen Quoc Tuan - 19522476
```

```
## Link github: https://github.com/tuNQws/data_mining.git
```

```
### III.Basic Python
```

```
1 +1
```

```
     2
```

```
1 * 3
```

```
     3
```

```
1 / 2
```

```
     0.5
```

```
2 ** 4
```

```
     16
```

```
4 % 2
```

```
     0
```

```
( 2 + 3 ) * ( 5 +5 )
```

```
     50
```

```
name_of_var = 2
```

```
x = 2
y = 3
```

```
z = x + y
```

```
z
```

```
     5
```

```
'single quotes'
```

```
     'single quotes'
```

```
"double quotes"
```

```
     'double quotes'
```

```
" wrap lot's of other quotes"
```

```
     ' wrap lot's of other quotes'
```

```
x = 'hello'
```

```
x
```

```
     'hello'
```

```
print(x)
```

```
     hello
```

```
num = 12
name = 'Sam'
```

```
print('My number is: {one}, and my name is: {two}'.format(one= num, two= name))
```

```
     My number is: 12, and my name is: Sam
```

```
print('My number is: {}, and my name is: {}'.format(num,name))
```

```
     My number is: 12, and my name is: Sam
```

```
[1,2,3]
```

```
     [1, 2, 3]
```

```
['hi',1,[1,2]]
```

```
     ['hi', 1, [1, 2]]
```

```
my_list = ['a','b','c']
```

```
my_list.append('d')
```

```
my_list
```

```
     ['a', 'b', 'c', 'd']
```

```
my_list[0]
```

```
     'a'
```

```
my_list[1]
```

```
     'b'
```

```
my_list[1:]
```

```
     ['b', 'c', 'd']
```

```
my_list[:1]
```

```
     ['a']
```

```
my_list[0] = 'NEW'
```

```
my_list
```

```
     ['NEW', 'b', 'c', 'd']
```

```
nest = [1,2,3,[4,5,['target']]]
```

```
nest[3]
```

```
     [4, 5, ['target']]
```

```
nest[3][2]
```

```
     ['target']
```

```
d = {'key1':'item1','key2':'item2'}
```

```
d
```

```
     {'key1': 'item1', 'key2': 'item2'}
```

```
d['key1']
```

```
     'item1'
```

```
True
```

```
     True
```

```
False
```

```
     False
```

```
t = (1,2,3)
```

```
t[0]
```

```
1
```

```
t[0] = 'NEW'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-75-93bfe9be1549> in <module>
----> 1 t[0] = 'NEW'

TypeError: 'tuple' object does not support item assignment
```

SEARCH STACK OVERFLOW

```
lst=list(t)
lst[0]='NEW'
t=tuple(lst)
t
```

```
('NEW', 2, 3)
```

```
{1,2,3}
```

```
{1, 2, 3}
```

```
{1,2,3,1,2,1,2,3,3,3,3,2,2,2,1,1,2}
```

```
{1, 2, 3}
```

```
1 >2
```

```
False
```

```
1 < 2
```

```
True
```

```
1 >= 1
```

```
True
```

```
1 <= 4
```

```
True
```

```
1 == 1
```

```
True
```

```
'hi' == 'bye'
```

```
False
```

```
(1 >2) and (2 <3)
```

```
False
```

```
(1>2) or (2<3)
```

```
True
```

```
(1==2) or (2==3) or (4==4)
```

```
True
```

```
if 1<2:
    print('Yep!')
```

```
Yep!
```

```
if 1<2:
    print('yep!')
```

```
yep!
```

```
if 1<2:
    print('first')
else:
    print('last')
```

```
first
```

```
if 1>2:
    print('first')
else:
    print('last')
```

```
last
```

```
if 1 == 2:
    print('first')
elif 3==3:
    print('middle')
else:
    print('Last')
```

```
middle
```

```
seq = [1,2,3,4,5]
```

```
for item in seq:
    print(item)
```

```
1
2
3
4
5
```

```
for item in seq:
    print('Yep')
```

```
Yep
Yep
Yep
Yep
Yep
```

```
for jelly in seq:
    print(jelly+jelly)
```

```
     2
     4
     6
     8
     10
```

```
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
```

```
range(5)
```

```
     range(0, 5)
```

```
for i in range(5):
    print(i)
```

```
     0
     1
     2
     3
     4
```

```
list(range(5))
```

```
     [0, 1, 2, 3, 4]
```

```
x =[1,2,3,4]
```

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

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

```
[item**2 for item in x]
```

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

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

```
<function __main__.my_func(param1='default')>
```

```
my_func('new param')
```

```
new param
```

```
my_func(param1='new param')
```

```
new param
```

```
def square(x):
    return x**2
out = square(2)
print(out)
```

```
4
```

```
def times(var):
    return var*2
times(2)
```

```
4
```

```
lambda var:var*2
```

```
<function __main__.<lambda>(var)>
```

```
seq = [1,2,3,4,5]
```

```
map(times, seq)
```

```
<map at 0x7f902fa63cd0>
```

```
list(map(times,seq))
```

```
[2, 4, 6, 8, 10]
```

```
list(map(lambda var: var*2,seq))
```

```
[2, 4, 6, 8, 10]
```

```
filter(lambda item: item%2 == 0, seq)
```

```
<filter at 0x7f900f4df490>
```

```
list(filter(lambda item: item%2==0, seq))
```

```
[2, 4]
```

```
st = 'hello my name is Sam'
st.lower()
```

```
'hello my name is sam'
```

```
st.upper()
```

```
'HELLO MY NAME IS SAM'
```

```
st.split()
```

```
['hello', 'my', 'name', 'is', 'Sam']
```

```
tweet = 'Go Sports! #Sports'
```

```
tweet.split('#')
```

```
['Go Sports! ', 'Sports']
```

```
tweet.split('#')[1]
```

```
     'Sports'
```

```
d
```

```
     {'key1': 'item1', 'key2': 'item2'}
```

```
d.keys()
```

```
     dict_keys(['key1', 'key2'])
```

```
d.items()
```

```
     dict_items([('key1', 'item1'), ('key2', 'item2')])
```

```
lst = [1,2,3]
```

```
lst.pop()
```

```
     3
```

```
lst
```

```
     [1, 2]
```

```
'x' in [1,2,3]
```

```
     False
```

```
'x' in ['x','y','z']
```

```
     True
```

```
#IV . Python basic
```

```
7 ** 4
```

```
2401
```

```
s = "Hi there Sam!"
words = s.split()
words[-1] = "dad!"
result = words[:2] + [words[-1]]
print(result)
```

```
['Hi', 'there', 'dad!']
```

```
planet = "Earth"
diameter="12742"
print("The diameter of {} is {} kilometers.".format(planet, diameter))
```

```
The diameter of Earth is 12742 kilometers.
```

```
my_list = [11, 2, [3, 4], [5, [100, 200, ('hello', 11, 23, 111, 1, 71)]]]
word = my_list[3][1][2][0]
print(word)
```

```
hello
```

```
d = {'kI*': (1, 2, 3, {'tricky': ('oh', 'man', 'inception', {'target': (1, 2, 3, 'hello')})})}
word = d['kI*'][3]['tricky'][3]['target'][3]
print(word)
```

```
hello
```

```
#Tuple is immutable
```

```
def get_domain(email):
    return email.split('@')[1]
```

```
get_domain('user@domain.com')
'domain.com'
```

```
'domain.com'
```

```
def findDog(string):
    return 'dog' in string.lower().split()
```

```
findDog("Is there a dog there?")
```

```
True
```

```
def count_dog(input_str):
    return input_str.lower().count('dog')
```

```
count_dog("This dog run faster than the other dog dude !")
```

```
2
```

```
['soup', 'salad']
```

```
def caught_speeding(speed, is_birthday):
    if is_birthday:
        speed -= 5
    if speed <= 60:
        return "No Ticket"
    elif speed >= 61 and speed <= 80:
        return "Small Ticket"
    else:
        return "Big Ticket"
```

```
print(caught_speeding(81, True))  # "Small Ticket"
print(caught_speeding(81, False))  # "Big Ticket"
```

```
Small Ticket
Big Ticket
```

```
print(caught_speeding(81, True))
```

```
Small Ticket
```

```
print(caught_speeding(81, False))
```

```
Big Ticket
```