

# Python for Data Science

The course is mainly provided in to main three parts:

1. General term of data science
2. Python introduction
3. Library for data science

## 1. General term of data science

This section is mainly focused on the environment setup for data science work, the way to use anaconda, markdown and anaconda tool.

## 2. Python Introduction

This section is described the general term of python programming include:

- i. Python variable and data type
- ii. Python control structure (selection structure and repetition structure)
- iii. python function (1. Built-in, 2. recursion, 3. Lambda and 4. user define function)

## 3. Library for data science

This section various library has picked up for the discussion:

- i. Numpy
- ii. Pandas
- iii. Matplotlib

## 2. Python Introduction

### i. Python variable and data types

In [2]:

```
#int variable
X=7
x=10
print(X)
print(x)
name="Cambodia"
print(name)
```

```
7
10
Cambodia
```

In [3]:

```
age =42
print(id(age)) #identify
print(type(age)) #type
print(age) #value
```

```
2123219562064
<class 'int'>
42
```

In [4]:

```
str_name= """ My Contry is "Cambodia" """
print(str_name)
name1='I like coding '
name2="I like Math " # string concate
print(name1 + name2)
```

```
My Contry is "Cambodia"
I like coding I like Math
```

In [5]:

```
mylist = ["apple", "banana", "cherry"] #index [0] array
print(id(mylist))
mylist.append('Durian') #myList=["apple", "banana", "cherry", durian]
print(id(mylist))
```

```
2123301160832
2123301160832
```

In [6]:

```
name1='23 '
print(id(name1))
name1='I like Math '
print(id(name1))
```

```
2123300853744
2123300855216
```

In [7]:

```
# data type of sequential type: List, string, tuple, set , dictionaries
# list [ .....],(contained diferent data types), order, changable, duplicate value
country= ['Cambodia', " Thailand", " Vieatnam", " Laos", " Malaysia", " Japan"]
print(country)
```

```
['Cambodia', ' Thailand', ' Vieatnam', ' Laos', ' Malaysia', ' Japan']
```

In [8]:

```
print(country[1:4]) # 4 not count
```

```
[' Thailand', ' Vieatnam', ' Laos']
```

In [9]:

```
print(len(country))
```

6

In [10]:

```
print(country[-1])
```

Japan

In [11]:

```
print(country[-5])  
print(country[3:])
```

Thailand  
[' Laos', ' Malaysia', ' Japan']

In [12]:

```
print(country[-1])
```

Japan

In [13]:

```
country[1:3]=["French", " England"]  
print(country)
```

['Cambodia', 'French', ' England', ' Laos', ' Malaysia', ' Japan']

In [14]:

```
country[4]=["Cambodia"]  
print(country)
```

['Cambodia', 'French', ' England', ' Laos', ['Cambodia'], ' Japan']

In [15]:

```
# append add item to the last index  
country.append("Russia")  
print(country)
```

['Cambodia', 'French', ' England', ' Laos', ['Cambodia'], ' Japan', 'Russia']

In [16]:

```
#insert specific index  
country.insert(2, "Swiss")  
print(country)
```

['Cambodia', 'French', 'Swiss', ' England', ' Laos', ['Cambodia'], ' Japan', 'Russia']

In [17]:

```
#extend concate two List (combine 2 Lists)
fruit=["Mango", " Cherry", "potato"]
country.extend(fruit)
print(country)
```

```
['Cambodia', 'French', 'Swiss', ' England', ' Laos', ['Cambodia'], ' Japan',
'Russia', 'Mango', ' Cherry', 'potato']
```

In [18]:

```
country[5]= "Cambodia"
print(country)
```

```
['Cambodia', 'French', 'Swiss', ' England', ' Laos', 'Cambodia', ' Japan',
'Russia', 'Mango', ' Cherry', 'potato']
```

In [19]:

```
#removed item from List: remove, pop(), del()
country.remove("Mango")
print(country)
```

```
['Cambodia', 'French', 'Swiss', ' England', ' Laos', 'Cambodia', ' Japan',
'Russia', ' Cherry', 'potato']
```

In [20]:

```
country.pop(5)#index
print(country)
```

```
['Cambodia', 'French', 'Swiss', ' England', ' Laos', ' Japan', 'Russia', ' C
herry', 'potato']
```

In [21]:

```
del country[8]
print(country)
```

```
['Cambodia', 'French', 'Swiss', ' England', ' Laos', ' Japan', 'Russia', ' C
herry']
```

In [22]:

```
#indentation
newlist=[]
for x in country:
    if "a" in x:
        newlist.append(x)
print(newlist)
```

```
['Cambodia', ' England', ' Laos', ' Japan', 'Russia']
```

In [23]:

```
# sort
country.sort()
print(country)
```

```
[' Cherry', ' England', ' Japan', ' Laos', 'Cambodia', 'French', 'Russia',
'Swiss']
```

In [24]:

```
num=[45,78,1,45,67,9,3]
print(num)
num.sort()
print(num)
```

```
[45, 78, 1, 45, 67, 9, 3]
[1, 3, 9, 45, 45, 67, 78]
```

## String

In [25]:

```
# string can store: ' ', " ", "" ""
name= " I love Cambodia!"
print(name)
```

```
I love Cambodia!
```

In [26]:

```
print(name[1:7])
```

```
I love
```

In [27]:

```
# modify (upper() and lower())
print(name.upper())
```

```
I LOVE CAMBODIA!
```

In [28]:

```
#replace() build-in
print(name.replace("o", "W"))
```

```
I lWve CambWdia!
```

In [29]:

```
#combine 2 strings
name1=" I love coding"
me=name+name1
print(me)
```

I love Cambodia! I love coding

In [30]:

```
a="23"
b1="20"
d1=a+b1# 2320
print(d1)
print(type(a))
b=int(a)
c=30
d=b+c
print(type(b))
print(d)
```

```
2320
<class 'str'>
<class 'int'>
53
```

## Tuple, set and dictionary

In [31]:

```
# list [ .....],(contained diferent data types), order, changable, allow duplicate value
# tuple (.....), (contained different tada type), order, Unchange, allow duplicate valu #
# set {.....}, (contained different tada type), unordered, Unchange,unidex
# Dictionary in key:value pairs, ordered, changeable and do not allow duplicates.

country= ("Cambodia", 'French', 'Swiss', ' England', ' Laos', ' Japan', 'Russia')

print(country)
```

('Cambodia', 'French', 'Swiss', ' England', ' Laos', ' Japan', 'Russia')

In [32]:

```
print(country[2])
#-3
# 2==>5
```

Swiss

In [33]:

```
#update tuple
My_country_li= list(country)
My_country_li[4]= "Brunei"
country=tuple(My_country_li)
print(country)
```

('Cambodia', 'French', 'Swiss', ' England', 'Brunei', ' Japan', 'Russia')

In [34]:

```
#unpack
(Romdol,a,b,v,n,h,o) = country
```

In [35]:

```
print(Romdol)
```

Cambodia

In [37]:

```
i=0
while i<len(country):
    print(country[i])
    i +=2
```

Cambodia  
Swiss  
Brunei  
Russia

## Set

In [38]:

```
book = {"Mein Kampf", "Think and grow rich", "Follow your heart", "Being happy", "7 habits"}
print(book)
```

{'Mein Kampf', 'Being happy', '7 habits', 8, 'Follow your heart', 'Think and grow rich'}

In [39]:

```
book = {"Mein Kampf", "Think and grow rich", "Follow your heart", "Being happy", "7 habits"}
print(book)
```

{'Mein Kampf', 'Being happy', '7 habits', 8, 'Follow your heart', 'Think and grow rich'}

In [40]:

```
book = {"Mein Kampf", "Think and grow rich", "Follow your heart", "Being happy", "7 habits"}
print(book)
print(book)
```

```
{'Mein Kampf', 'Being happy', '7 habits', 8, 'Follow your heart', 'Think and grow rich'}
{'Mein Kampf', 'Being happy', '7 habits', 8, 'Follow your heart', 'Think and grow rich'}
```

## Dictionary

In [42]:

```
dict={
    "Hello":"សួស្តី",
    "Love" : "ស្រឡាញ់",
    "Age" : 20}
print(dict)
```

```
{'Hello': 'សួស្តី', 'Love': 'ស្រឡាញ់', 'Age': 20}
```

In [43]:

```
print(dict["Hello"])
```

សួស្តី

In [44]:

```
for key in dict :
    print (key)
```

```
Hello
Love
Age
```

In [45]:

```
t=dict["Age"]#20
print(t)
```

20

In [46]:

```
k=dict.keys()
print(k)
```

```
dict_keys(['Hello', 'Love', 'Age'])
```



In [47]:

```
dict["Like"] ="ចូលចិត្ត"  
print(dict)
```

```
{'Hello': 'សួស្តី', 'Love': 'ស្រឡាញ់', 'Age': 20, 'Like': 'ចូលចិត្ត'}
```

In [48]:

```
#data structure : Link-List, array, #algorithm: sorting  
#pop  
dict.pop("Age")  
print(dict)  
#popitem  
#del
```

```
{'Hello': 'សួស្តី', 'Love': 'ស្រឡាញ់', 'Like': 'ចូលចិត្ត'}
```

In [49]:

```
dict.clear()  
print(dict)
```

```
{}
```

## ii. Control structure

In this section the control structure in Python will be discussed on:

- a. repetition structure (for)
- b. selection structure (if, if/elif)

### Repetition structure

In [50]:

```
#for ... in ...(string, list,tuple, dictionary, set, Array,...)  
#for ... in range(0,11) #for(i=0; i<=10,i++)  
fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
    print(x)
```

```
apple  
banana  
cherry
```

In [51]:

```
fruits = {"apple", "banana", "cherry"}  
for x in fruits:  
    print(x)
```

```
banana  
apple  
cherry
```

In [52]:

```
fruits = ("apple", "banana", "cherry")
for x in fruits:
    print(x)
```

apple  
banana  
cherry

In [53]:

```
for x in range(0,11):
    print(x)
```

0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10

In [54]:

```
for x in range(0,11,2): #for(i=0,i<=10,i=i+2)
    print(x)
```

0  
2  
4  
6  
8  
10

In [73]:

```
for i in range (1,101):
    if (i ==10 or i ==20):
        continue
    print(i,end=' ')
```

1 2 3 4 5 6 7 8 9 11 12 13 14 15 16 17 18 19 21 22 23 24 25 26 27 28 29 30 3  
1 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56  
57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 8  
2 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

In [74]:

```
for i in range (1,101):  
    print(i,end=' ')  
    if (i ==11):  
        break;
```

1 2 3 4 5 6 7 8 9 10 11

## Selection structure

In [56]:

```
fruits = ["apple", "banana", "cherry",12,78,"Cambodia"]  
for x in fruits:  
    if x == 12:# skip banana  
        continue  
    print(x)
```

In [57]:

```
name=["LALA","Baba","toto",34,18,'ee']  
for x in name:  
    if x== 34:  
        print(x)
```

34

In [58]:

```
name=["LALA","Baba","toto",34,18,'ee']  
for x in name:  
    if x== "ee":  
        print(x)
```

ee

In [59]:

```
name=["LALA","Baba","toto",34,18,'ee']  
for x in name:#last value ee  
    if x== 34:  
        break  
    print(x)
```

LALA

Baba

toto

In [60]:

```
a=2001
b=2002
if a>b:
    print("A>B")
if a<b:
    print("A<B")
```

A<B

In [61]:

```
a=2001
b=2001
if a>b:
    print("A>B")
elif a<b:
    print("A<B")
else:
    print("A = B")
```

A = B

In [62]:

```
a=2001
b=2001
if a>=b:
    print("A>=B")
else:
    print("A<B")
```

A>=B

In [63]:

```
i = 1 #int i=1
while i < 6: #while(i<6){ }
    print(i)
    i += 1
else:
    print("i is no longer less than 6")
```

```
1
2
3
4
5
i is no longer less than 6
```

## Some important projects

password Authentication using python

In [64]:

```
import getpass
database = {"sophal":"1234", "senglay":"6789","rith":"asdfg"}# key:value
username = input("Enter username: ") # sophal
password = getpass.getpass("Enter password: ") #12345678
for i in database.keys():
    if username == i:
        while password != database.get(i):
            password = getpass.getpass("Enter password again: ")
        break
print("logged in!!")
```

```
Enter username: sophal
Enter password: .....
Enter password again: .....
logged in!!
```

### iii. Python function

There are four main function type in python:

1. Built-in function, refers to the function that already has in python programing
2. User defined function
3. Recursion function
4. Lambda function

### User defined function

In [66]:

```
#function def (return function = fruitful function )
def rectangle(w,h):
    return w*h

area=rectangle(20,5) #20*5
#enter from user
print(area)
```

100

In [67]:

```
# void function
def myself(name,p_number):
    print("My name is " + name)
    print("My phone number is "+ p_number)

myself("Messi", "09876541")
```

```
My name is Messi
My phone number is 09876541
```

In [71]:

```
#watt = amp * volt

def watt(amp, volt): #None
    return amp * volt
a=67
b=30
#input amp
#input volt
print("The value of watt is ", watt(30,2))
```

The value of watt is 60

In [ ]:

```
##Do by yourself
#watt= amp * volt // key argument
solv(watt=None, amp=None, volt=None):
    watt=None, return amp*volt
    amp=None, return watt/volt
    volt=None, return watt/amp

***Which parameter do you want to find?***
1.find watt
2.find amp
3.find volt
chice: 1
    enter amp:
    enter volt:
    watt=??
```

In [75]:

```
#return function have keyword return
#paramednter vs argument
def sum_3_num(a, b, c):
    return a+b+c

#print(sum_3_num(45,6,7))

def triangle_area(a,b):
    area=0.5 *a*b
    return area
a =triangle_area(6,9)
#print(a)

def far_to_cel(far):
    return (far-32)*(5/9)

a=float(input("Enter Farenheit temperature: "))
temp_cel=far_to_cel(a)

print(f"{a} degree farenheit = {temp_cel:.2f} degree celius")
print(a, " degree farenheit = ",round(temp_cel,3), " degree celius")

#3 return function : circle area, Multiple of 3 number, Fahrenheit to celcius
```

```
Enter Farenheit temperature: 4
4.0 degree farenheit = -15.56 degree celius
4.0 degree farenheit = -15.556 degree celius
```

In [76]:

```
def invest_fv(pv,rate,years):
    for i in range(1,years+1):
        pv=pv*(1+rate/100)
        print(f"year {i} : ${pv:.2f}")

pv=int(input("Enter your money $: "))
rate=float(input("Enter a yearly rate: "))
years=int(input("Enter number of years: "))

invest_fv(pv,rate,years)
```

```
Enter your money $: 567800
Enter a yearly rate: 23
Enter number of years: 5
year 1 : $698394.00
year 2 : $859024.62
year 3 : $1056600.28
year 4 : $1299618.35
year 5 : $1598530.57
```

## Lambda function

In [72]:

```
def rectangle(b,h):  
    return b * h  
  
rec = lambda b,h:b*h  
triangle = lambda b,h:0.5 * b * h  
  
def area(func,b,h):  
    return func(b,h)  
  
if __name__=="__main__":#double underscore  
    print("hi")  
    print(rectangle(6,8))  
    print(rec(5,9))  
    print(triangle(5,10))  
    print(area(rec,11,3))  
    print(area(triangle,67,2))
```

```
hi  
48  
45  
25.0  
33  
67.0
```

In [77]:

```
# lambda argument: expression  
def area_rec(w,h):  
    return w*h  
  
x = lambda w , h : w*h  
far_to_cel = lambda far: ((far-32)*(5/9))  
cel_to_far = lambda cel: ((cel * 9/5) + 32)  
print(area_rec(20,45))  
print(x(20,45))  
print(far_to_cel(59))  
print(cel_to_far(34))
```

```
900  
900  
15.0  
93.2
```



In [78]:

```
#def far_to_cel(far):
#    return (far-32)*(5/9)
x=lambda far:(far-32)*(5/9)
#def cel_to_far(cel):
#    return (cel*9/5)+32
y=lambda cel:(cel*9/5)+32

print("Welcome to temperature coversion program")
print("1. Celious to farenheit")
print("2. farenheit to celious")
n=int(input("Choise: "))
if(n==1):
    a=float(input("Enter celious degree: "))
    b=x(a)
    print(f"The farenheit degree at {a} celious degree is: {b:.3f} ")
#elif (n==2):v
```

```
Welcome to temperature coversion program
1. Celious to farenheit
2. farenheit to celious
Choise: 1
Enter celious degree: 56
The farenheit degree at 56.0 celious degree is: 13.333
```

## Recursion function

In [79]:

```
#4!=4*3*2*1
#f(n)=f(n-1)*f(n-2)*f(n-3)*...*f(1)
def factory(n):
    if n<0:
        return "Error: only possitive accepted"
    elif n==0:
        return 1
    else:
        return n *factory(n-1) #4*f(3)=4*3*f(2)=4*3*2*f(1)=4*3*2*1*f(0)=4*3*2*1*1

print(factory(4))
```

In [80]:

```
def countdown_num(n):
    if n==0:
        return 0
    elif n<0: #-10+1
        print(n)
        return countdown_num(n+1)
    else:
        print(n)
        return countdown_num(n-1)

num=int(input("Enter number: "))
print(countdown_num(num))
```

Enter number: 8

8  
7  
6  
5  
4  
3  
2  
1  
0

In [81]:

```
# exp_num(5,2): 50= 5* 5**2
def exp_num(n,m):
    if m==0:
        return n
    else:
        return n*exp_num(n,m-1)
print(exp_num(5,2))
#5*exp(5,1)=5*5*exp(5,0)=5*5*5=125
```

125

## some important tips:

1. python zip()
2. python filer()
3. python map()

python zip()

In [82]:

```
#BMI = w/(h/100)**2 by using ndarray
def test():
    import numpy as np
    w=np.array([60,70,80,65,98])
    h=np.array([156,170,165,168,180])
    BMI=w/(h/100)**2
    print(BMI)

test()
```

```
[24.65483235 24.22145329 29.38475666 23.03004535 30.24691358]
```

In [83]:

```
#BMI=[] by using (List comprehension)
def test1():
    w=[60,70,80,65,98]
    h=[156,170,165,168,180]
    bmi=[]
    for i in range(len(w)):
        bmi.append(w[i]/(h[i]/100)**2)
    return bmi

print(test1())
```

```
[24.654832347140037, 24.221453287197235, 29.384756657483933, 23.030045351473
927, 30.24691358024691]
```

In [84]:

```
#by using zip()
def test2():
    weight=[60,70,80,65,98]
    height=[156,170,165,168,180]
    bmi=[]
    for w,h in zip(weight, height):
        bmi.append(w/(h/100)**2)
    return bmi

#float_arr=np.array(test2())
#np.set_printoptions(precision=2)
#print(float_arr)
print(test2())
```

```
[24.654832347140037, 24.221453287197235, 29.384756657483933, 23.030045351473
927, 30.24691358024691]
```

In [85]:

```
def test3():
    weight=[60,70,80,65,98]
    height=[156,170,165,168,180]
    bmi=[]
    return [ w/(h/100)**2 for w,h in zip(weight, height)]

print(test3())
```

```
[24.654832347140037, 24.221453287197235, 29.384756657483933, 23.030045351473
927, 30.24691358024691]
```

In [86]:

```
def test4():
    weight=[60,70,80,65,98]
    height=[156,170,165,168,180]
    name=["Dara", "Pisey", "Theary", "Piseth", "Vireakbot"]
    bmi=[]
    return [{n:w/(h/100)**2} for w,h,n in zip(weight,height,name)]

print(test4())
```

```
[{'Dara': 24.654832347140037}, {'Pisey': 24.221453287197235}, {'Theary': 29.
384756657483933}, {'Piseth': 23.030045351473927}, {'Vireakbot': 30.246913580
24691}]
```

In [87]:

```
def test5():
    weight=[60,70,80,65,98]
    height=[156,170,165,168,180]
    name=["Dara", "Pisey", "Theary", "Piseth", "Vireakbot"]
    gender=['M', 'F', 'F', 'M', 'M']
    return [{n:w/(h/100)**2} for w,h,n,g in zip(weight,height,name, gender) if g=='M']

print(test5())
```

```
[{'Dara': 24.654832347140037}, {'Piseth': 23.030045351473927}, {'Vireakbot':
30.24691358024691}]
```

## Python map()

In [88]:

```
def map_sample():
    flower=['jasmine','lutus','lily','sunflower','rose']
    flower1=list(map(str.capitalize,flower))
    print(flower)
    print(flower1)

map_sample()
```

```
['jasmine', 'lutus', 'lily', 'sunflower', 'rose']
['Jasmine', 'Lutus', 'Lily', 'Sunflower', 'Rose']
```

In [89]:

```
def usd_2_riel(usd):  
    return usd*4000  
def map_sample1():  
    price_usd=[56,89,23,67,30]  
    price_riel=list(map(usd_2_riel,price_usd))  
    price_riel1=[x*4000 for x in price_usd]  
    print(price_riel)  
    print(price_riel1)
```

map\_sample1()

```
[224000, 356000, 92000, 268000, 120000]  
[224000, 356000, 92000, 268000, 120000]
```

In [91]:

```
# 1 hexta= 10000m^2  
#1 Ah=1000m^2  
#hexta-Ah-Meter: 5-3-20 (5*10000+ 3*1000+20)  
def area():  
    s=input("Hexta-Ah-meter : ").split('-')  
    #print(s)  
    hexta,ah,meter=list(map(int,s))  
    total_area=hexta*10000+ah*1000+meter  
    print(total_area)
```

area()

```
Hexta-Ah-meter : 12-4-9  
124009
```

In [92]:

```
def area1():  
    hexta,ah,meter=list(map(int, input("Hexta-Ah-Meter : ").split('-')))  
    return hexta*10000+ah*1000+meter
```

area1()

```
Hexta-Ah-Meter : 34-6-1
```

Out[92]:

```
346001
```

**Python filter()**

In [93]:

```
#funct test() has list age =[10,34,55,23,78,77,45], age_e=filter()
def test():
    age=[10,34,55,23,78,77,45]
    age_e=filter(lambda x:x%2==0,age)
    print(type(age_e))
    print(age_e)
    print(list(age_e))

test()
```

```
<class 'filter'>
<filter object at 0x000001EE5EA82820>
[10, 34, 78]
```

In [94]:

```
def test1():
    age=[10,34,55,23,78,77,45]
    age1=lambda x:x%2 ==0
    age2=lambda x:x%2 !=0
    age_o=filter(age2,age)
    print(list(age_o))

test1()
```

```
[55, 23, 77, 45]
```

In [95]:

```
#list country =['Cambodia','Lao','Malaysia','Indonesia','Singapore','Qata','Brunei'] filter
def test2():
    country =['Cambodia','Lao','Malaysia','Indonesia','Singapore','Qata','Brunei']
    country1=filter(lambda x:'s' in x ,country)
    print(list(country1))

test2()
```

```
['Malaysia', 'Indonesia']
```

In [96]:

```
#test3() country =[['Cambodia',49],['Lao',40],['Malaysia',60],['Indonesia',53],['Singapore'  
#filter country which has english score less than 60  
def test3():  
    country =[['Cambodia',49],['Laos',40],['Malaysia',60],['Indonesia',53],['Singapore',65]  
    country_low_score=filter(lambda x:x[1]<60,country)  
    country2=filter(lambda x:len(x[0])<5,country)  
    l = [x for x in country if(len(x[0]) <5)]  
    l1 =[x[0] for x in country if (len(x[0])>5)]  
    print(list(country_low_score))  
    print(list(country2))  
    print(l)  
    print(l1)  
  
test3()
```

```
[['Cambodia', 49], ['Laos', 40], ['Indonesia', 53]]  
[['Laos', 40], ['Qata', 70]]  
[['Laos', 40], ['Qata', 70]]  
['Cambodia', 'Malaysia', 'Indonesia', 'Singapore', 'Brunei']
```

### 3. Library for data science

1. Numpy
2. Pandas
3. Matplotlib

In [ ]: