

Data Science For Everyone Using Python

Manaranjan Pradhan

manaranjan@enablecloud.com

*This notebook is given as part of **Data Science for everyone** workshop.*

(Forwarding this document to others is strictly prohibited.)

Python Basics

Declaring Variables and printing

In [2]:

```
var1 = 2  
var2 = 5
```

In [3]:

```
var1
```

Out[3]:

2

In [4]:

```
print( var1 )
```

2

In [5]:

```
mystring = 'This is python'  
print( mystring )
```

This is python

In [6]:

```
print( var1, var2, mystring )
```

2 5 This is python

Operations on variables .. Arithmetic or logical

In [7]:

```
var1 + var2
```

Out[7]:

7

In [8]:

```
var1 * var2
```

Out[8]:

10

In [9]:

```
var1 == 2
```

Out[9]:

True

In [10]:

```
var1 == var2
```

Out[10]:

False

Built-in functions

In [11]:

```
round( 1.234 )
```

Out[11]:

1

In [12]:

```
# Round upto a number of decimal values  
round( 1.234, 2 )
```

Out[12]:

1.23

In [13]:

```
# Importing a math function  
In [14]:  
import math
```

```
math.ceil( 1.2 )
```

Out[14]:

2

In [15]:

```
math.floor( 1.2 )
```

Out[15]:

1

In [16]:

```
abs( -1.2 )
```

Out[16]:

1.2

In [17]:

```
# Get the variable type  
type( var1 )
```

Out[17]:

int

In [18]:

```
pow( var1 , 2 )
```

Out[18]:

4

In [19]:

```
## Generate a sequence number  
numbers = range( 1, 10 )
```

In [20]:

```
numbers
```

Out[20]:

```
range(1, 10)
```

In [21]:

```
type( numbers )
```

Out[21]:

range

In [22]:

```
for i in numbers:  
    print( i )
```

1
2
3
4
5
6
7
8
9

In [23]:

```
len( numbers )
```

Out[23]:

9

In [24]:

```
for i in numbers:  
    print(i , end = " ")
```

1 2 3 4 5 6 7 8 9

Control Flow Statements

In [25]:

```
if var1 > 1:  
    print( "Bigger" )
```

Bigger

In [26]:

```
if var1 > 5:  
    print( "Bigger" )  
else:  
    print( "Smaller" )
```

Smaller

In [27]:

```
x = 10
y = 12
if x > y:
    print ("x>y")
elif x < y:
    print ("x<y")
else:
    print ("x=y")
```

x<y

In [28]:

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

0
1
2
3
4

In [29]:

```
i = 1
while i < 5:
    print(i)
    i = i+1
print('Bye')
```

1
2
3
4
Bye

In [30]:

```
i = 1
while i < 5:
    print(i)
    i = i+1
    if i == 4:
        break
print('Bye')
```

1
2
3
Bye

In [31]:

```
i = 1
while i < 5:
    i = i+1
    if i == 3:
        continue
    print(i)
print('Bye')
```

```
2
4
5
Bye
```

Working with Data Structures

List - Collection of elements... (Elements can repeat)

In [32]:

```
## Create an empty list
a = []
```

In [33]:

```
fruits = ['apple', 'orange', 'banana', 'papaya']
```

In [34]:

```
fruits[0]
```

Out[34]:

```
'apple'
```

In [35]:

```
## Slicing an List
fruits[1:3]
```

Out[35]:

```
['orange', 'banana']
```

In [36]:

```
## Accessing the Last element
fruits[-1]
```

Out[36]:

```
'papaya'
```

In [37]:

```
# how many elements in the list  
len( fruits )
```

Out[37]:

4

In [38]:

```
seasonal_fruits = ['mango', 'cherry', 'watermelon']
```

In [39]:

```
all_fruits = fruits + seasonal_fruits
```

In [40]:

```
all_fruits
```

Out[40]:

```
['apple', 'orange', 'banana', 'papaya', 'mango', 'cherry', 'watermelon']
```

In [41]:

```
'banana' in all_fruits
```

Out[41]:

True

In [42]:

```
'grapes' in fruits
```

Out[42]:

False

In [43]:

```
all_fruits.index( 'banana' )
```

Out[43]:

2

In [44]:

```
all_fruits.append( 'grapes' )
```

In [45]:

```
all_fruits
```

Out[45]:

```
['apple',  
'orange',  
'banana',  
'papaya',  
'mango',  
'cherry',  
'watermelon',  
'grapes']
```

In [46]:

```
a = [1,1,2,4,5,6,7]
```

In [47]:

```
a
```

Out[47]:

```
[1, 1, 2, 4, 5, 6, 7]
```

In [48]:

```
min( a )
```

Out[48]:

```
1
```

In [49]:

```
max( a )
```

Out[49]:

```
7
```

In [50]:

```
## How many times an element exists in a list  
a.count( 1 )
```

Out[50]:

```
2
```

In [51]:

```
a.insert( 3, 3 )
```


In [52]:

```
a
```

Out[52]:

```
[1, 1, 2, 3, 4, 5, 6, 7]
```

In [53]:

```
a.reverse()
```

In [54]:

```
a
```

Out[54]:

```
[7, 6, 5, 4, 3, 2, 1, 1]
```

In [55]:

```
a.sort()
```

In [56]:

```
a
```

Out[56]:

```
[1, 1, 2, 3, 4, 5, 6, 7]
```

Tuples - Immutable List

In [57]:

```
tup1 = ( 1, 3, 'orange' )
```

In [58]:

```
tup1
```

Out[58]:

```
(1, 3, 'orange')
```

In [59]:

```
## It is not allowed t change the tuple elements..  
tup1[1] = 'a'
```

```
-----  
----  
TypeError                                Traceback (most recent call 1  
ast)  
<ipython-input-59-f67dd2a4584f> in <module>()  
      1 ## It is not allowed t change the tuple elements..  
----> 2 tup1[1] = 'a'
```

TypeError: 'tuple' object does not support item assignment

In [60]:

```
tupa = tuple( a )
```

In [61]:

```
tupa
```

Out[61]:

```
(1, 1, 2, 3, 4, 5, 6, 7)
```

Set - Order list of non-repeating items

In [62]:

```
b = set( [6,1,1,2,4,5] )
```

In [63]:

```
b
```

Out[63]:

```
{1, 2, 4, 5, 6}
```

In [64]:

```
b.add( 3 )
```

In [65]:

```
b
```

Out[65]:

```
{1, 2, 3, 4, 5, 6}
```

In [66]:

```
c = set( [2,4,6,7] )
```

In [67]:

```
c.union( b )
```

Out[67]:

```
{1, 2, 3, 4, 5, 6, 7}
```

In [68]:

```
b.intersection( c )
```

Out[68]:

```
{2, 4, 6}
```

In [69]:

```
c.difference( b )
```

Out[69]:

```
{7}
```

In [70]:

```
b.remove( 3 )
```

In [71]:

```
b
```

Out[71]:

```
{1, 2, 4, 5, 6}
```

In [72]:

```
b.clear()
```

In [73]:

```
b
```

Out[73]:

```
set()
```

Iterating through the elements in list or set

In [74]:

```
for i in a:  
    print( i * 2 )
```

```
2  
2  
4  
6  
8  
10  
12  
14
```

In [75]:

```
for i in b:  
    print( i )
```

Using a Dictionary

In [76]:

```
d0 = {}  
d1 = dict( { 'One': 1, 'Two':2 } )  
d1
```

Out[76]:

```
{'One': 1, 'Two': 2}
```

In [77]:

```
d0['One'] = 1  
d0['OneTwo'] = 12  
print( d0 )
```

```
{'OneTwo': 12, 'One': 1}
```

In [78]:

```
d0['One']
```

Out[78]:

```
1
```

In [79]:

```
# Join two lists and create an dictionary...  
names = ['One', 'Two', 'Three', 'Four', 'Five']  
numbers = [1, 2, 3, 4, 5]
```

In [80]:

```
d2 = dict( zip(names,numbers) )
```

In [81]:

```
print( d2 )
```

```
{'Two': 2, 'Five': 5, 'Four': 4, 'One': 1, 'Three': 3}
```

In [82]:

```
d2.keys()
```

Out[82]:

```
dict_keys(['Two', 'Five', 'Four', 'One', 'Three'])
```

In [83]:

```
d2.values()
```

Out[83]:

```
dict_values([2, 5, 4, 1, 3])
```

In [84]:

```
d2['six'] = 6
```

In [85]:

```
d2
```

Out[85]:

```
{'Five': 5, 'Four': 4, 'One': 1, 'Three': 3, 'Two': 2, 'six': 6}
```

In [86]:

```
# Remove an element and return it  
d2.pop( 'six' )
```

Out[86]:

```
6
```

In [87]:

```
d2
```

Out[87]:

```
{'Five': 5, 'Four': 4, 'One': 1, 'Three': 3, 'Two': 2}
```

Dealing with Strings

In [88]:

```
string0 = 'python'  
string1 = "Data Science"  
string2 = '''This is Data science  
            workshop  
            using Python'''
```

In [89]:

```
print( string0, string1, string2)
```

```
python Data Science This is Data science  
            workshop  
            using Python
```

In [90]:

```
string2.find( "Python" )
```

Out[90]:

```
53
```

In [91]:

```
string0.capitalize()
```

Out[91]:

```
'Python'
```

In [92]:

```
string0.upper()
```

Out[92]:

```
'PYTHON'
```

In [93]:

```
len( string2 )
```

Out[93]:

```
59
```

In [94]:

```
string2.split()
```

Out[94]:

```
['This', 'is', 'Data', 'science', 'workshop', 'using', 'Python']
```

In [95]:

```
string2.replace( 'Python', 'R')
```

Out[95]:

```
'This is Data science \n          workshop\n          using R'
```

Type *Markdown* and LaTeX: α^2

Functions in Python

In [96]:

```
def addElements( a, b ):
    return a + b
```

In [97]:

```
addElements( 2, 3 )
```

Out[97]:

5

In [98]:

```
addElements( 2.3, 4.5 )
```

Out[98]:

6.8

In [99]:

```
addElements( "python", "workshop" )
```

Out[99]:

'pythonworkshop'

In [100]:

```
def addElements( a, b ):
    return a, b, a + b
```

In [101]:

```
addElements( 2, 3 )
```

Out[101]:

(2, 3, 5)

In [102]:

```
addElements( 2.3, 4.5 )
```

Out[102]:

```
(2.3, 4.5, 6.8)
```

In [103]:

```
x, y, z = addElements( 4, 5 )
```

In [104]:

```
x
```

Out[104]:

```
4
```

In [105]:

```
def addElements( a, b = 4 ):
    return a + b
```

In [106]:

```
addElements( 2 )
```

Out[106]:

```
6
```

In [107]:

```
addElements( 2, 5 )
```

Out[107]:

```
7
```

In [108]:

```
def add_n(*args):
    sum = 0
    for arg in args:
        sum = sum + arg
    return sum
```

In [109]:

```
add_n( 1, 2, 3 )
```

Out[109]:

```
6
```


In [110]:

```
add_n( 1, 2, 3, 4, 5, 6 )
```

Out[110]:

21

In [111]:

```
add_n()
```

Out[111]:

0

Lambda Functions in Python

In [112]:

```
a = lambda x: x * x
```

In [113]:

```
a( 2 )
```

Out[113]:

4

In [114]:

```
a( 2 ) * a( 2 )
```

Out[114]:

16

In [115]:

```
mylist = [1,2,3,4,5,6,7,8,9]
```

In [116]:

```
xsquare = []  
  
for x in mylist:  
    xsquare.append( pow( x, 2 ) )  
  
print( xsquare )
```

[1, 4, 9, 16, 25, 36, 49, 64, 81]

In [117]:

```
map( lambda x: pow( x, 2 ), mylist)
```

Out[117]:

```
<map at 0x816278>
```

In [118]:

```
xsquare1 = list( map( lambda x: pow( x, 2 ), mylist) )
```

In [119]:

```
print( xsquare1 )
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81]
```

In [120]:

```
mylist1 = [1,2,3,4,5,6,7,8,9]
```

In [121]:

```
listprods = list( map( lambda x, y: x * y, mylist, mylist1 ) )
```

In [122]:

```
listprods
```

Out[122]:

```
[1, 4, 9, 16, 25, 36, 49, 64, 81]
```

In [123]:

```
list( filter( lambda x : x < 5, list1 ) )
```

```
-----
----
NameError                                Traceback (most recent call 1
ast)
<ipython-input-123-d2cc08ce53a1> in <module>()
----> 1 list( filter( lambda x : x < 5, list1 ) )
```

```
NameError: name 'list1' is not defined
```

Classes and Objects

In [124]:

```
class Student:
    workshop = 'python'
    def __init__(self,name,age):
        self.name = name
        self.age = age
    def describe( self ):
        print( self.name, " is ",
              self.age,
              " years old and participating in ",
              Student.workshop,
              " class ")
    return
```

In [125]:

```
dir( student1 )
```

```
-----
----
NameError                                Traceback (most recent call 1
ast)
<ipython-input-125-1ed96f2d105e> in <module>()
----> 1 dir( student1 )
```

NameError: name 'student1' is not defined

In [126]:

```
student1 = Student( "manaranjan", 39 )
```

In [127]:

```
student1.name
```

Out[127]:

```
'manaranjan'
```

In [128]:

```
student1.describe()
```

```
manaranjan is 39 years old and participating in python class
```

In [129]:

```
Student.workshop = "R"
```

In [130]:

```
student1.workshop
```

Out[130]:

```
'R'
```

Note: Make note of lessons learnt in this workshop