Variable and Constant

Variable

- A container that is going to store a particular value in it and we can change the store value in the variable at any point of time
- A variable is going to retain the last assigned in it till the next assignment is done
- A variable name should always be in lowercase (small letter)

In [1]:
```
a = 10
print(a)
```

```
10
```

In [2]:
```
b = 20
c = 2.0
d = 2+7j
e= 'Mayank'
```

In [3]:
```
a
```

Out[3]:
```
10
```

In [4]:
```
a = 12
print(a)
```

```
12
```

Constants

- Constants are variable only but we have set our mind that we will never change their value
- Constants should always be in uppercase (Capital letter)

In [6]:
```
PI = 3.14
r = int(input('Enter the radius'))
circumfernce = 2*PI*r
print(circumfernce)
```

```
Enter the radius14
87.92
```

In [7]:
```
2*3.14*14
```

Out[7]:
```
87.92
```

id() function

- It is used to tell me the memory space location/address which is associated with the value

In [8]:
```
a = 1
print(id(a))
```

```
140728745890600
```

In [9]:
```
print(a)
```

```
1
```

In [11]:
```
a = 1
b = 1
print(id(a),id(b))
```

```
140728745890600 140728745890600
```

Object Reusability / Memory Resuability

In [ ]:
```
int --- -5 to 256
Boolean - True or False
string --- normal string do follow object resulability but comlpex string
```

In [12]:
```
a = 10
b = 10
print(id(a), id(b))
```

```
140728745890888 140728745890888
```

In [13]:
```
a = 257
b = 257
print(id(a), id(b))
```

```
3075848147184 3075848155440
```

In [14]:
```
a = 256
b = 256
print(id(a), id(b))
```

```
140728745898760 140728745898760
```

In [15]:
```
a = -5
b = -5
print(id(a), id(b))
```

```
140728745890408 140728745890408
```

In [16]:
```
a = -6
b = -6
print(id(a), id(b))
```

```
3075848154960 3075848147056
```

In [19]:
```
a = 1
b = 2
print(id(a), id(b))
```

```
140728745890600 140728745890632
```

In [17]:
```
a = 'Mayank'
b = 'Mayank'
print(id(a), id(b))
```

```
3075819864944 3075819864944
```

Special Characters are !@#$%^&*()-+= and space

In [27]:
```
a = 'Mayank$Ghai'
b = 'Mayank$Ghai'
print(id(a), id(b))
```

```
3075848601136 3075849373104
```

In [28]:
```
a = 'Mayank_Ghai'
b = 'Mayank_Ghai'
print(id(a), id(b))
```

```
3075849313456 3075849313456
```

In [30]:
```
a = 'Mayankatulghai@'
b = 'Mayankatulghai@'
print(id(a), id(b))
```

```
3075849356592 3075849359664
```

Boolean -- True or False

In [32]:
```
1 bool(1)
```

Out[32]: True

In [33]:
```
1 bool(0)
```

Out[33]: False

In [38]:
```
1 a = bool(1)
2 b = bool(1)
3 print(id(a), id(b))
```

```
140728744421920 140728744421920
```

In [39]:
```
1 c =True
2 print(id(c))
```

```
140728744421920
```

In [40]:
```
1 bool(0)
```

Out[40]: False

In [41]:
```
1 bool(1)
```

Out[41]: True

In [42]:
```
1 bool(6)
```

Out[42]: True

In [43]:
```
1 bool(7)
```

Out[43]: True

In [44]:
```
1 a = bool(100)
2 b = bool(-100)
3 print(id(a), id(b))
```

```
140728744421920 140728744421920
```

In [45]:
```
a = True
b = bool(1)
c = 1
print(id(a),id(b),id(c))

```

```
140728744421920 140728744421920 140728745890600
```

Datatype

- A data-type is a type identifier of the value or information that is being stored in a variable

Different Types of Data

Basic Datatypes or Fundamental Datatypes or Primitive Datatypes

- int
- float
- complex
- string
- boolean
- None
- range

Derived Datatypes

- list
- tuples
- dictionary
- set
- frozenset

type() function

- It tells us about which kind of the datatype the variable has stored in it

Integer

- Any number +ve, -ve or 0 , without a decimal value is known as integer

In [46]:
```
a = 100
print(type(a))
```

```
<class 'int'>
```

In [47]:
```
a = -100
print(type(a))
```

```
<class 'int'>
```

In [48]:
```
a
```

```
<class 'int'>
```

float

- Any value +ve, -ve or 0 with decimal values is known as float datatype

In [49]:
```
a = 10.0
print(type(a))
```

```
<class 'float'>
```

In [50]:
```
b = 20.9
print(type(a))
```

```
<class 'float'>
```

In [51]:
```
a = 10.0
b = 10.0
print(id(a), id(b))
```

```
3075848144944 3075848156016
```

In [ ]:
```
10.0 ---- 100/10 ---- 100*10^-1 ---- 1*10^1 ----100e1
```

In [54]:
```
100e-1
Python stores the float values in exponential form and this makes them nu
```

Out[54]:
```
10.0
```

In [56]:
```
a = 10
b = 10.0
print(id(a), id(b))
print(type(a), type(b))
```

```
140728745890888 3075848143600
<class 'int'> <class 'float'>
```

Complex Datatype

- Any number stored in the form of a+bj or a+bJ will be considered as a complex datatype
- The number without the j or J is known as a real number and the part with the J or J in called the imaginary number

In [104]:
```
x = 6+5j #---> 6.0
print(type(x))
```

```
<class 'complex'>
```

In [105]:
```
id(x)
```

Out[105]:
```
3075849650480
```

In [106]:
```
y = 7-7J  #-------------- 7+(-7)j
print(type(y))
```

```
<class 'complex'>
```

In [107]:
```
x.real
```

Out[107]:
```
6.0
```

In [108]:
```
x.imag
```

Out[108]:
```
5.0
```

In [109]:
```
y.real
```

Out[109]:
```
7.0
```

In [110]:
```
y.imag
```

Out[110]:
```
-7.0
```

Precedence

Int<Float<Complex ----- Numeric datatypes

String

- Anything written inside '....', "....", '''.....''' or """....."""
- It can be a character, word, sentense, integer, float, or any other datatype

In [63]:
```
1 a = '1'
2 print(type(a))
```

```
<class 'str'>
```

In [65]:
```
1 a = "10.0"
2 print(type(a))
```

```
<class 'str'>
```

In [66]:
```
1 a = '''7+8j'''
2 print(type(a))
```

```
<class 'str'>
```

In [67]:
```
1 a = """Mayank"""
2 print(type(a))
```

```
<class 'str'>
```

In [68]:
```
1 a = '10.0'
2 b = '10.0'
3 print(id(a), id(b))
```

```
3075849371504 3075849310576
```

In [69]:
```
1 a = 'mayank112333435i3y6798569874589349'
2 print(type(a))
```

```
<class 'str'>
```

In [71]:
```
1 a = "I'm a data analyst"
2 print(a)
```

```
I'm a data analyst
```

In [72]:
```
a = '''My name is Mayank Atul Ghai
I am a python trainer at learnbay
I am having more than 7 years of experience'''
print(a, type(a))
```

```
My name is Mayank Atul Ghai
I am a python trainer at learnbay
I am having more than 7 years of experience <class 'str'>
```

'''....''' or """....."""" are used for Docstring or Also knowns as Documenatational Strings

In [75]:
```
def add(a, b):
    '''Functionality: Will take two numbers and return the sum of these t
        a = First Number
        b = second number
        c = result'''

    return c
```

In [ ]:
```
add() #shift +tab
```

Boolean Datatype

- Boolean means two
- It will store only two binary values in it i.e True or False

In [76]:
```
a = True
print(type(a))
```

```
<class 'bool'>
```

In [77]:
```
b = False
print(type(b))
```

```
<class 'bool'>
```

None Datatype

- It is used to represent missing values or a null value in Python
- It is space holder

In [78]:
```
a = None
print(type(a))
```

```
<class 'NoneType'>
```

In [80]:
```
Name = 'Mayank'
Age = 32
City = None
```

*None*_ is not used a variable name

In [82]:
```
id(a)
```

Out[82]: 140728744504048

In [83]:
```
id(City)
```

Out[83]: 140728744504048

Range

- It will give you a range between two specifed number
- the last number will not be included
- It is going to start by default by 0

In [84]:
```
range(5)
```

Out[84]: range(0, 5)

In [85]:
```
for i in range(5):
    print(i)
```

```
0
1
2
3
4
```

In [86]:
```
range(1,5)
```

Out[86]: range(1, 5)

In [88]:
```
for i in range(1,5):
    print(i)
```

```
1
2
3
4
```

In [89]:
```
a = 1
b = 20
c = range(a, b)
print(c)
```

```
range(1, 20)
```

In [90]:
```
range(-5,-1)
```

Out[90]:
```
range(-5, -1)
```

In [95]:
```
for i in range(-5, -1):
    print(i)
```

```
-5
-4
-3
-2
```

In [96]:
```
print(i)
```

```
-2
```

In [92]:
```
type(c)
```

Out[92]:
```
range
```

In [98]:
```
c = print('Mayank')
type(c)
```

Out[98]:
```
str
```

In [ ]:
```
range(starting value, ending_value, step_size)
by default starting_value will take 0

```

In [101]:
```
for i in range(10,20):
    #----> [10, 20)
    print(i)
```

```
10
11
12
13
14
15
16
17
18
19
```

In [111]:
```
2.0
```

Out[111]:
```
2.0
```

In [114]:
```
a = 2e0
```

In [115]:
```
id(a)
```

Out[115]:
```
3075848157840
```

In [116]:
```
a = 2
id(a)
```

Out[116]:
```
140728745890632
```

In [ ]:
```

```