basics of python programming

python type casting

.int

.float

.str

integers

In [2]:
```
x = int(1)
y = int(2.8)
z = int("3")
print(x)
print(y)
print(z)
```

```
1
2
3
```

floats:

In [3]:
```
x = float(1)
y = float(2.8)
z = float("3")
w = float("4.2")
print(x)
print(y)
print(z)
print(w)
```

```
1.0
2.8
3.0
4.2
```

strings:

In [4]:
```
x = str("S1")
y = str(2)
z = str(3.0)
print(x)
print(y)
print(z)
```

```
S1
2
3.0
```

boolean

In [5]:
```
bool('hai')
```

Out[5]:
```
True
```

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

Out[6]:
```
False
```

In [7]:
```
bool('6712')
```

Out[7]:
```
True
```

In [8]:
```
print(10>9)
print(10 == 9)
print(10 <9)
```

```
True
False
False
```

In [10]:
```
x = "hello"
y = 15

print(bool(x))
print(bool(y))
```

```
True
True
```

In [11]:
```
bool("abc")
bool(123)
bool(["apple","cherry","banana"])
```

Out[11]:
```
True
```

In [12]:
```
 2+ 3.4 + True + False + 55.21 -11
```

Out[12]:
```
50.61
```

In [13]:
```
int(55.45)
```

Out[13]:
```
55
```

slicing

In [1]:
```
s = 'Hi Everyone'
s
```

Out[1]:
```
'Hi Everyone'
```

In [2]:
```
len(s)
```

Out[2]:
```
11
```

In [18]:
```
s[0],s[1],s[2]
```

Out[18]:
```
('H', 'i', ' ')
```

In [19]:
```
s[0:2],s[8:],s[8:10],s[8:11]
```

Out[19]:
```
('Hi', 'one', 'on', 'one')
```

In [20]:
```
s[::2]
```

Out[20]:
```
'H vroe'
```

In [21]:
```
s[3: :3]
```

Out[21]:
```
'Ern'
```

negitive indexing

In [22]:
```
s[-1],s[-4],s[-1:-5:-1]
```

Out[22]:
```
('e', 'y', 'enoy')
```

In [23]:
```
s[::-1]
```

Out[23]:
```
'enoyrevE iH'
```

In [8]:
```
A = "IQRA is very beautiful girl"
A
```

Out[8]:
```
'IQRA is very beautiful girl'
```

In [9]:
```
len(A)
```

Out[9]:
```
27
```

In [10]:
```
A[::-1]
```

Out[10]:
```
'lrig lufituaeb yrev si ARQI'
```

In [11]:
```
A[-1]
```

Out[11]:
```
'l'
```

In [12]:
```
A[-2]
```

Out[12]:
```
'r'
```

In [13]:
```
A[:6]
```

Out[13]:
```
'IQRA i'
```

In [14]:
```
A[:8]
```

Out[14]:
```
'IQRA is '
```

operations

Arthematic operators

In [48]:
```
## + addition
## - subtraction
## /,//,% for division
## * multiplication
## ** power of
```

In [50]:
```
a = 5
b = 3
print(a+ b)
```

```
8
```

In [55]:
```
a-b
```

Out[55]:
```
2
```

In [54]:
```
a*b
```

Out[54]:
```
15
```

In [53]:
```
a/b
```

Out[53]:
```
1.6666666666666667
```

In [52]:
```
a%b
```

Out[52]:
```
2
```

In [51]:
```
a**b
```

Out[51]:
```
125
```

In [56]:
```
a//b # when you want to get integer output i.e it will ignore the decimal part
```

Out[56]:
```
1
```

In [57]:
```
a%b
```

Out[57]:
```
2
```

Assignment operators

In [51]:
```
x = 15
y = 25
```

In [52]:
```
print(x ,y)
```

```
15 25
```

In [78]:
```
x = 15
y = 25
x +=3
print(x)
```

```
18
```

In [54]:
```
print(y-3)
```

```
22
```

In [73]:
```
x = 15
x*=3
print(x)
```

```
45
```

In [72]:
```
y = 25
y%=3
print(y)
```

```
1
```

In [69]:
```
x = 15
x <<=3
print(x)
```

```
120
```

In [70]:
```
y = 25
y>>=3
print(y)
```

```
3
```

In [71]:
```
x = 15
x^=3
print(x)
```

```
12
```

In [79]:
```
x//=3
print(x)
```

```
6
```

comparision operators

In [12]:
```
x = 5
y = 3

print(x == y)
```

```
False
```

In [14]:
```
x = 5
y = 3

print(x != y)
```

```
True
```

In [15]:
```
x = 4
y =2

print(x > y)
```

```
True
```

In [16]:
```
x = 3
y = 2

print(x < y)
```

```
False
```

In [17]:
```
x = 5
y = 7

print(x >= y)
```

```
False
```

In [18]:
```
x = 7
y = 10

print(x <= y)
```

```
True
```

logical operators

In [22]:
```
x = 25
print(x > 10 and x < 100)
```

```
True
```

In [23]:
```
x =25
print(x > 10 or x < 100)
```

```
True
```

In [26]:
```
x = 25
print((not x > 10 and x < 100))
```

```
False
```

input functions

In [1]:
```
num = input('Enter a value')
num
```

```
Enter a value12
'12'
```

Out[1]:

In [3]:
```
type(num)
```

Out[3]:
```
str
```

In [4]:
```
num = float(input("Enter a value"))
num
```

```
Enter a value3.4
3.4
```

Out[4]:

In [5]:
```
num = int(input("Enter a value"))
num
```

```
Enter a value12
12
```

Out[5]:

In [6]:
```
a = 7
b = 10
```

- eval function

In [8]:
```
x = eval('a+b')
x
```

Out[8]:
```
17
```

In [9]:
```
x = eval('a*b')
x
```

Out[9]:
```
70
```

In [11]:
```
x = 'print(55)'
eval(x)
```

```
55
```

In [13]:
```
k = eval(x)
```

```
55
```

In [16]:
```
a = 5.5
b = 10
y = ('a+b')
y
```

Out[16]:
```
'a+b'
```

- place holder in python

In [23]:
```
num = 10
print("The value of a number=", num)
```

```
The value of a number= 10
```

In [38]:
```
num = 10
print("The value of a number=" + str(num))
```

```
The value of a number=10
```

In [24]:
```
num = 999
print("The value of a number=", num)
```

```
The value of a number= 999
```

In [25]:
```
name = "sajid"
print('My name is ',name)
```

```
My name is  sajid
```

In [5]:
```
name = "iqra"
age = 6
print("My  daughter name is {} and  her age is {}".format(name,age))
```

```
My  daughter name is iqra and  her age is 6
```

In [2]:
```
name = "sajid"
age = 25
print("My name is {1} and my age is {0} and my name is {1}".format(age,name))
```

```
My name is sajid and my age is 25 and my name is sajid
```

In [3]:
```
name = "sajid"
age = 25
role = "Data scientist"
print("My name is {0},and my age is {1} and my role is {2}".format(name,age,role))
```

```
My name is sajid,and my age is 25 and my role is Data scientist
```

In [7]:
```
name = "iqra"
age = 6
role = "Quran Enthusiast"
print("My  daughter name is {} she is an {} and  her age is {}".format(name,role,age))
```

```
My  daughter name is iqra she is an Quran Enthusiast and  her age is 6
```

conditional statements

In [6]:
```
## the conditional operators are
=  # operator to do assignment
== # equal to operator
> #greater than
< #less than
>= #greater than equal to
<= #less than  equal to
!= #not equal
```

```
  File "C:\Users\sajid\AppData\Local\Temp/ipykernel_18556/226900425.py", line 2
    1) =  # operator to do assignment
     ^
SyntaxError: unmatched ')'
```

In [3]:
```
a = 10
if (a ==10):
    print("A  value is 10")
    print("if block")
    print("continue")
    print("out of break")
```

```
A  value is 10
if block
continue
out of break
```

In [8]:
```
a = 10
if (a==5):
    print("A value is 10")
    print("If block")
    print("continue")
print("out of break")
```

```
out of break
```

In [11]:
```
a = 10
if(a==5):
    print("A value is 10")
    print("If block")
    print("continue")
else:
    print("if statement failed")
    print("Else statement excuted")
```

```
if statement failed
Else statement excuted
```

In [31]:
```
a = 23
b = 24
if a>b:
    print(" datamites is great")
else:
    print("iam a datascientist ")
    print("My name is sajid")
```

```
iam a datascientist
My name is sajid
```

In [15]:
```
a = 0
if (a !=0):
    print("A is positive")
else:
    print("A is negitive")
```

```
A is negitive
```

In [8]:
```
a =float(input("Enter the number"))
if(a>0):
    print("A is positive")
elif(a<0):
    print("A is negitive")
elif(a!=0):
    print("sajid is great")
else:
    print("A is zero")
```

```
Enter the number5
A is positive
```

In [2]:
```
sajid = float(input("enter the number"))
if(sajid>0):
    print("sajid is very good boy")
elif(sajid<0):
    print("sajid is interested in data analysis")
elif(sajid!=0):
    print("sajid is excellent")
else:
    print("sajid is good and decent")
```

```
enter the number-9
sajid is interested in data analysis
```

DATA STRUCTURES

List

- ordered hetrogenous data which indexed
- mutable
- operations
- []

In [28]:
```
l = [1,2,3,5, 'hi',4.7]
l
```

Out[28]:
```
[1, 2, 3, 5, 'hi', 4.7]
```

In [30]:
```
l2 =list([4,5,6,7,'True','False'])
l2
```

Out[30]:
```
[4, 5, 6, 7, 'True', 'False']
```

In [26]:
```
type(l2)
```

Out[26]:
```
list
```

In [34]:
```
L =[1,2,3,4]
L[2] = 48
L
```

Out[34]:
```
[1, 2, 48, 4]
```

In [36]:
```
#operations-> inserting,deleting,sort,slicing
l[2] #index value starts with 0
```

Out[36]:
```
3
```

In [38]:
```
l=[]
l
```

Out[38]:
```
[]
```

In [44]:
```
#Insertion ->  append,extend,insert
## append will append element at the end of list and can add only one element at a time
```

In [43]:
```
l1.append(5.6)
l1
```

Out[43]:
```
[5.6, 5.6, 5.6]
```

In [45]:
```
l2 =list([4,5,6,7,'True','False'])
l2
```

Out[45]:
```
[4, 5, 6, 7, 'True', 'False']
```

In [47]:
```
l2.append(6.5)
l2
```

Out[47]:
```
[4, 5, 6, 7, 'True', 'False', 6.5, 6.5]
```

In [49]:
```
# extend used for added multiple items at a end od list
```

In [48]:
```
l2.extend([7,8,9])
l2
```

Out[48]:
```
[4, 5, 6, 7, 'True', 'False', 6.5, 6.5, 7, 8, 9]
```

In [1]:
```
# create a list with Emp name,emp id,age,no.of exp, mob no
```

In [70]:
```
l = ["sajid",'067543',25,3,9100461127,'Btech']
l
```

Out[70]:
```
['sajid', '067543', 25, 3, 9100461127, 'Btech']
```

In [66]:
```
l[0],l[3]
```

Out[66]:
```
('sajid', 3)
```

In [67]:
```
l.insert(2,'Btech')
l
```

Out[67]:
```
['sajid', '067543', 'Btech', 25, 3, 9100461127]
```

In [18]:
```
# extend -> add /insert more than one element
```

In [68]:
```
l.extend([1,5,10])
l
```

Out[68]:
```
['sajid', '067543', 'Btech', 25, 3, 9100461127, 1, 5, 10]
```

In [23]:
```
# Deletion -> pop,remove and del
# pop-> deletes the last element and return that element
```

In [24]:
```
l.pop()
l
```

Out[24]:
```
['sajid',
 '067543',
 'Btech',
 'Btech',
 'B tech',
 'B tech',
 'B tech',
 'B tech',
 'B tech',
 'B tech',
 25,
 3,
 9100461127,
 1,
 5]
```

In [69]:
```
l.pop(4)
l
```

Out[69]:
```
['sajid', '067543', 'Btech', 25, 9100461127, 1, 5, 10]
```

In [71]:
```
# remove -> delete elements with the specific element
l.remove('Btech')
l
```

Out[71]:
```
['sajid', '067543', 25, 3, 9100461127]
```

In [36]:
```
# delete the entire list or specific value
```

In [72]:
```
del l[-2]
l
```

Out[72]:
```
['sajid', '067543', 25, 9100461127]
```

In [39]:
```
# slicing-> start,stop,interval
```

In [47]:
```
l3 = ["sajid",9,9100461127,'08916',25]
l3
```

Out[47]:
```
['sajid', 9, 9100461127, '08916', 25]
```

In [48]:
```
l3[2:4]
l3
```

Out[48]:
```
['sajid', 9, 9100461127, '08916', 25]
```

In [50]:
```
l3[-1]
l3
```

Out[50]:
```
['sajid', 9, 9100461127, '08916', 25]
```

In [53]:
```
# nested list
l3
```

Out[53]:
```
['sajid', 9, 9100461127, '08916', 25]
```

In [54]:
```
l3.insert(2,20)
l3
```

Out[54]:
```
['sajid', 9, 20, 9100461127, '08916', 25]
```

In [56]:
```
l3.insert(3,7730057539)
l3
```

Out[56]:
```
['sajid', 9, 20, 7730057539, 9100461127, '08916', 25]
```

In [58]:
```
l3[4] = [10,20,30]
l3
```

Out[58]:
```
['sajid', 9, [10, 20, 30], 7730057539, [10, 20, 30], '08916', 25]
```

Tuples

- ordered hetrogenous data which indexed
- immutable
- ()

In [75]:
```
# data can't be change
```

In [73]:
```
t1 = (2,3,4,"Hai",6)
t1
```

Out[73]:
```
(2, 3, 4, 'Hai', 6)
```

In [74]:
```
t1.count(4)
t1
```

Out[74]:
```
(2, 3, 4, 'Hai', 6)
```

In [76]:
```
type(t1)
```

Out[76]:
```
tuple
```

In [77]:
```
t1.index(6)
```

Out[77]:
```
4
```

In [82]:
```
t1[3]
t1
```

Out[82]:
```
(2, 3, 4, 'Hai', 6)
```

In [83]:
```
t1.index(3)
t1
```

Out[83]:
```
(2, 3, 4, 'Hai', 6)
```

Dictionary

In [ ]:
```
*key/value pairs,hetrogenous,unique,
```

In [86]:
```
d1 ={'Hi': 'old', 'test':'prin'}
d1
```

Out[86]:
```
{'Hi': 'old', 'test': 'prin'}
```

In [87]:
```
d2 = {"name":"sajid","age": "25", "college" : "kiet"}
d2
```

Out[87]:
```
{'name': 'sajid', 'age': '25', 'college': 'kiet'}
```

In [90]:
```
d3 = {"name":"sajid","age": "25", "college" : "kiet"}
d3
```

Out[90]:
```
{'name': 'sajid', 'age': '25', 'college': 'kiet'}
```

In [91]:
```
type(d3)
```

Out[91]:
```
dict
```

In [88]:
```
#Nested Dictionary
d1["data"] = [45,75,45,12]
d1
```

Out[88]:
```
{'Hi': 'old', 'test': 'prin', 'data': [45, 75, 45, 12]}
```

In [89]:
```
d1["data"] = (45,75,45,12)
d1
```

Out[89]:
```
{'Hi': 'old', 'test': 'prin', 'data': (45, 75, 45, 12)}
```

In [2]:
```
# create a dict with no.as key and no.name as value
d2 ={1:'one',5:'five',9:'Nine',4:'four'}
d2
```

Out[2]:
```
{1: 'one', 5: 'five', 9: 'Nine', 4: 'four'}
```

In [3]:
```
d2[9]= (99,109,119)
d2
```

Out[3]:
```
{1: 'one', 5: 'five', 9: (99, 109, 119), 4: 'four'}
```

sets

In [18]:
```
#unique,contains hetrogenous elements
```

In [19]:
```
s1 = {1,2,3,4,5,6}
s2 = {2,3,4,6,9,8}
s1
```

Out[19]:
```
{1, 2, 3, 4, 5, 6}
```

In [4]:
```
s2
```

Out[4]:
```
{2, 3, 4, 6, 8, 9}
```

In [20]:
```
s1,max
```

Out[20]:
```
({1, 2, 3, 4, 5, 6}, <function max>)
```

In [22]:
```
sajid = {3,6,9,12,15,18,21}
shalima={2,4,6,8,10,12,14,16,18}
```

In [23]:
```
sajid
```

Out[23]:
```
{3, 6, 9, 12, 15, 18, 21}
```

In [24]:
```
shalima
```

Out[24]:
```
{2, 4, 6, 8, 10, 12, 14, 16, 18}
```

In [25]:
```
sajid.union(shalima)
```

Out[25]:
```
{2, 3, 4, 6, 8, 9, 10, 12, 14, 15, 16, 18, 21}
```

In [26]:
```
sajid.intersection(shalima)
```

Out[26]:
```
{6, 12, 18}
```

In [27]:
```
sajid.add("bujji")
```

In [28]:
```
sajid
```

Out[28]:
```
{12, 15, 18, 21, 3, 6, 9, 'bujji'}
```

In [29]:
```
shalima.add("shannu")
```

In [30]:
```
shalima
```

Out[30]:
```
{10, 12, 14, 16, 18, 2, 4, 6, 8, 'shannu'}
```

In [8]:
```
#0perations ->union,intersection
```

In [9]:
```
s1.union(s2)
```

Out[9]:
```
{1, 2, 3, 4, 5, 6, 8, 9}
```

In [10]:
```
s1.intersection(s2)
```

Out[10]:
```
{2, 3, 4, 6}
```

In [11]:
```
s1.add("How r u")
s1
```

Out[11]:
```
{1, 2, 3, 4, 5, 6, 'How r u'}
```

In [12]:
```
s1.difference(s2)
s1
```

Out[12]:
```
{1, 2, 3, 4, 5, 6, 'How r u'}
```

In [13]:
```
s1.union(s2)
```

Out[13]:
```
{1, 2, 3, 4, 5, 6, 8, 9, 'How r u'}
```

In [14]:
```
s1 = {1,2,3,4}
s2 = {4,'one',6,'Hi'}
s1.union(s2)
```

Out[14]:
```
{1, 2, 3, 4, 6, 'Hi', 'one'}
```

In [15]:
```
s1.add(5)
s1
```

Out[15]:
```
{1, 2, 3, 4, 5}
```

Functions

- code reusability,decompose

In [18]:
```
def tree_n(l,b,h):
    v = l * b * h
    return v
vol= tree_n(5,6,7)
vol
```

Out[18]:
```
210
```

In [2]:
```
def area_m(l,b,h):
    area =1/2*l*b*h
    return area
area  = area_m(4,5,6)
area
```

Out[2]:
```
60.0
```

In [19]:
```
def sajid_s(h,w,a):
    s = h * w * a
    return s
s= sajid_s(5.4,65,25)
s
```

Out[19]:
```
8775.0
```

In [1]:
```
def shalima_s(h,w,a):
    s = h * w * a
    return s
s = shalima_s(5.3,55,25)
```

```
s
```

Out[1]:
```
7287.5
```

In [6]:
```
def car_c(windows, doors, petrolprice):
    d = windows * doors * petrolprice
    return d
c = car_c(4, 5, 121.3)
c
```

Out[6]:
```
2426.0
```

In [2]:
```
def avg(n1,n2,n3):
    return(n1+n2+n3)/3.0

print("Welcome")
sajid1 = avg(10,20,30)
result2 = avg(1,2,3)
result3 = avg(2.4,3.7,5.8)

print(sajid1)
print(result2)
print(result3)
```

```
Welcome
20.0
2.0
3.9666666666666663
```

In [3]:
```
def avg(sajid,shalima,anisa):
    return(sajid+shalima+anisa)/3.0

print("I love u ")
```

```
I love u
```

In [5]:
```
def display():
    print("sajid")
    print("have a nice day")

display()
```

```
sajid
have a nice day
```

loops

- for loop
- while loop

In [20]:
```
# while loop
```

In [8]:
```
a =1
while a<=10:
    print(a)
    a = a+1
```

```
1
2
3
4
5
6
7
8
9
10
```

In [1]:
```
# for loop
```

In [9]:
```
for i in range(1,5):
    print(i)
```

```
1
2
3
4
```

In [7]:
```
for i in range(1,6):
    print(i)
```

```
1
2
3
4
5
```

In [12]:
```
for  i in range (1,11):
    print(i, '*6=',i*6)
```

```
1 *6= 6
2 *6= 12
3 *6= 18
4 *6= 24
5 *6= 30
6 *6= 36
7 *6= 42
8 *6= 48
9 *6= 54
10 *6= 60
```

In [15]:
```
for i in range(1,11):
    print(i, ' 19 * =',i *19 )
```

```
1  19 * = 19
2  19 * = 38
3  19 * = 57
4  19 * = 76
5  19 * = 95
6  19 * = 114
7  19 * = 133
8  19 * = 152
9  19 * = 171
10  19 * = 190
```

In [15]:
```
for i in range(20,0,-4):
    print(i)
```

```
20
16
12
8
4
```

In [1]:
```
s1 = {1,2,4,5,8}
s2 = {2,3,4,6,7}
s1.union(s2)
```

Out[1]:
```
{1, 2, 3, 4, 5, 6, 7, 8}
```

In [2]:
```
s1.intersection(s2)
```

Out[2]:
```
{2, 4}
```

In [2]:
```
str1 = "programinghub3"
x = ""
for i in str1:
    x +=i
    print(x)
```

```
p
pr
pro
prog
progr
progra
program
programi
programin
programing
programingh
programinghu
programinghub
programinghub3
```

In [1]:
```
str2 =  "sajidlovesdatascience"
x = ""
for i in str2:
    x +=i
    print(x)
```

```
s
sa
saj
saji
sajid
sajidl
sajidlo
sajidlov
sajidlove
sajidloves
sajidlovesd
sajidlovesda
sajidlovesdat
sajidlovesdata
sajidlovesdatas
sajidlovesdatasc
sajidlovesdatasci
sajidlovesdatascie
sajidlovesdatascien
sajidlovesdatascienc
sajidlovesdatascience
```

In [3]:
```
x = 'community'
y = 'community'
print(x > y)
```

```
False
```

In [9]:
```
list = [1,2,3, "GFG", 2.4]
print(list)
```

```
[1, 2, 3, 'GFG', 2.4]
```

In [1]:
```
# while loop
wanna_play_again ='y'
while wanna_play_again =='y':
    wanna_play_again = input("Do you want to play again? (y/n) ? ")
else:
    print("we excited the Game !!")
```

```
Do you want to play again? (y/n) ? n
we excited the Game !!
```

In [2]:
```
num = int(input("Enter the number"))
```

```
Enter the number45
```

In [3]:
```
 for i in range(45):
        print(i)
```

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
```

In [ ]: