Python History

REF - https://en.wikipedia.org/wiki/Python_(programming_language) REF - https://en.wikipedia.org/wiki/Guido_
https://en.wikipedia.org/wiki/Meertens_number REF - https://en.wikipedia.org/wiki/Richard_Bird_(computer_sc
https://en.wikipedia.org/wiki/Bird%E2%80%93Meertens_formalism REF - https://en.wikipedia.org/wiki/G%C3%
https://en.wikipedia.org/wiki/G%C3%B6del_numbering REF - https://en.wikipedia.org/wiki/Monty_Python%27s_
https://en.wikipedia.org/wiki/Amoeba_(operating_system)

National Research Institute for Mathematics and Computer Science

Richard Bird and Meertens formalism - BMF (Bird Meertens Formalism) - to handle map and reduce operations
number - `81312000 = 2^8 * 3^1 * 5^3 * 7^1 * 11^2 & 13^0 * 17^0 * 19^0`)

ABC - Lembert Meertens - Was in its initial stage

Guido van Rossum 1980 - took development of the language

Why the name Python?

After Monte Python Flying Circus

What is special in Monte Python Flying Circus

1. It ridiculed complexity of British life
2. Their team members are intellectuals and called themselves pythons.
3. Pythons they named themselves because they used sarcasm and double meaning to ridicule tight rule ba
4. Why Circus? Because their team tried to impress upon BBC by going around the office building, kind of ci
5. Why flying? Because the TV show is aired and it kind of flies wherever TV sets are installed
6. What is 'Monte'? Named after Lord Montgomery (because the 6 ppl are kind of rebels waging war agains
7. Why van Rossum had to adapt Python name? Because this language simplifies life.

Tenets of Python

1. Explicit > Implicit
2. Beautiful > Ugly
3. Simple > Complex
4. Complex > Complicated
5. Readability counts

```
print ('kali')
```

```
kali
```

```
! ls
```

```
sample_data
```

Basic Python

Online REF Materials

PDF - https://doc.lagout.org/programmation/python/Programming%20in%20Python%203_%20A%20Complete%20Int%282nd%20ed.%29%20%5BSummerfield%202009-11-22%5D.pdf

Google colab - https://colab.research.google.com/drive/1s_SZs0DTm9aFrdhiFDzZju3xbQY2n_uF?authuser=2#

Practice - https://colab.research.google.com/drive/1s_SZs0DTm9aFrdhiFDzZju3xbQY2n_uF?usp=sharing

1. https://www.w3schools.com/python/
2. https://www.w3resource.com/python-exercises/

Basic variables, Keyboard and Print

```python
x = 3

y = x +123

print (y)
```

```
126
```

```python
x = "kali"

print (x)
```

```
kali
```

```python
x = 3.14234

print (x)
```

```
3.14234
```

```python
i = input("enter string:")

print (i)
```

```
enter string:123
123
```

```python
j = i + 3

print (j)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-7-e22694a535d8> in <module>()
----> 1 j = i + 3
      2 
      3 print (j)

TypeError: must be str, not int
```

```python
j = int(i) + 3

print (j)
```

```
126
```

```python
j = int(i) + 4.5

print (j)
```

```
127.5
```

```python
j = float(i) + 4.5

print (j)
```

```
127.5
```

```python
print (i,j)
```

```
123 127.5
```

```python
print (i,'kali',j)
```

```
123 kali 127.5
```

```python
i = 3
j = 4.7

print ('i=%d, j=%f'%(i,j))

print ('i=%-6d, j=%-10.2f'%(i,j))

print ('i=%-6d, j=%5.2f, str=%s, hex=%x'%(i,j,'kali',(i+300)))

mystr = 'kali'

print ('i={0}, j={1}, str={2}, hex={3}'.format(i,j,mystr,hex(i+300) ) )

```

```
i=3, j=4.700000
i=3     , j=4.70      
i=3     , j= 4.70, str=kali, hex=12f
i=3, j=4.7, str=kali, hex=0x12f
```

Simple List operations

```python
x = ['kali','das']

print (x)
```

```
['kali', 'das']
```

```
print (len(x))
```

```
2
```

```
x = ['abc', 7.8, -3.4, 23, 'def', ['abc', 123]]

print (x, len(x))
```

```
['abc', 7.8, -3.4, 23, 'def', ['abc', 123]] 6
```

```
x = x + ['abc', 34.654]

print (x)
```

```
['abc', 7.8, -3.4, 23, 'def', ['abc', 123], 'abc', 34.654]
```

```
print (len(x))
```

```
8
```

```
print (7.7999 in x)
```

```
False
```

```
print ('abc' in x)
```

```
True
```

```
print ('ab' in x)
```

```
False
```

```
print (34.654 in x)
```

```
True
```

```
print (['abc', 123] in x)
```

```
True
```

```
print (['abc', 122.9999999991] in x)
```

```
False
```

```
print ([123,'abc'] in x)
```

```
False
```

```
print (x[0])
```

```
abc
```

```
print (x[1])
```

```
7.8
```

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

```
34.654
```

```
print (x[-3])
```

```
['abc', 123]
```

```
x = x - 34.654

print (x)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-30-a38c2ec4b983> in <module>()
----> 1 x = x - 34.654
      2 
      3 print (x)

TypeError: unsupported operand type(s) for -: 'list' and 'float'
```

SEARCH STACK OVERFLOW

```
x = x - [34.654]

print (x)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-31-f982fee32952> in <module>()
----> 1 x = x - [34.654]
      2 
      3 print (x)

TypeError: unsupported operand type(s) for -: 'list' and 'list'
```

SEARCH STACK OVERFLOW

```
print (x)

del x[3]

print (x)
```

```
['abc', 7.8, -3.4, 23, 'def', ['abc', 123], 'abc', 34.654]
['abc', 7.8, -3.4, 'def', ['abc', 123], 'abc', 34.654]
```

```python
print (x)

y = x.pop(0)

print (x,y)
```

```
['abc', 7.8, -3.4, 'def', ['abc', 123], 'abc', 34.654]
[7.8, -3.4, 'def', ['abc', 123], 'abc', 34.654] abc
```

```python
x = x + ['abc', 'abc', 'bcd', 'abc' ]

print (x)

x.remove('abc')

print (x)
```

```
[7.8, -3.4, 'def', ['abc', 123], 'abc', 34.654, 'abc', 'abc', 'bcd', 'abc']
[7.8, -3.4, 'def', ['abc', 123], 34.654, 'abc', 'abc', 'bcd', 'abc']
```

```python
x = "kalidas"

print (len(x))
```

```
7
```

```python
print (x[3])
```

```
i
```

```python
print("i" in x)
```

```
True
```

```python
print ("id" in x)
```

```
True
```

```python
print ("di" in x)
```

```
False
```

```python
x = "kalidas"

x.append("abcdef")

print (x)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-40-23b7a16411e3> in <module>()
      1 x = "kalidas"
      2 
----> 3 x.append("abcdef")
      4 
      5 print (x)

AttributeError: 'str' object has no attribute 'append'
```

```python
x = "kalidas"

x = x + "abcdef"

print (x)
```

```
kalidasabcdef
```

```python
x = []

x.append("abcdef")

x.append(12.45)

print (x)
```

```
['abcdef', 12.45]
```

```python
print (13 not in x)
```

```
True
```

```python
print (12.45 in x, "def" in x, "abcdef" in x)
```

```
True False True
```

extend

```python
x = ['kalidas','abcdef',12.45]

y = ['pqrst', 45]

x.extend(y)

print (x, len(x))
```

```
['kalidas', 'abcdef', 12.45, 'pqrst', 45] 5
```

append

```python
x = ['kalidas','abcdef',12.45]

y = ['pqrst', 45]

x.append(y)

print (x, len(x))
```

```
['kalidas', 'abcdef', 12.45, ['pqrst', 45]] 4
```

recursion

```python
x = ['kalidas','abcdef',12.45]

x.append(x)

print (x,len(x))
```

```
['kalidas', 'abcdef', 12.45, [...]] 4
```

```python
print (x[-1][-1][-1][-1][-1][-1])
```

```
['kalidas', 'abcdef', 12.45, [...]]
```

indexing

```python
x = ['kalidas','abcdef',12.45, [12, 45.62, 'pqr']]

print (x.index("kalidas"))

print (x.index("abcdef"))

print (x.index(12.45))
```

```
0
1
2
```

```python
x = ['kalidas','abcdef',12.45, [12, 45.62, 'pqr']]

print (x.index("12.45"))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-50-337c404c0588> in <module>()
      1 x = ['kalidas','abcdef',12.45, [12, 45.62, 'pqr']]
      2 
----> 3 print (x.index("12.45"))

ValueError: '12.45' is not in list
```

SEARCH STACK OVERFLOW

```python
x = ['kalidas','abcdef',12.45, [12, 45.62, 'pqr']]

print (x.index([12, 45.62, 'pqr']))
```

```
3
```

```python
x = ['kalidas','abcdef',12.45]

print (x)

x.reverse()

```

```python
print (x)

x.sort()

print (x)
```

```
['kalidas', 'abcdef', 12.45]
[12.45, 'abcdef', 'kalidas']
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-52-c99b8ecc6578> in <module>()
      7 print (x)
      8 
----> 9 x.sort()
     10 
     11 print (x)

TypeError: '<' not supported between instances of 'str' and 'float'
```

SEARCH STACK OVERFLOW

```python
x = ['kalidas','abcdef',"12.45"]

print (x)

x.sort()

print (x)
```

```
['kalidas', 'abcdef', '12.45']
['12.45', 'abcdef', 'kalidas']
```

unpacking

```python
x = [3,4]

print (x)
```

```
[3, 4]
```

```python
a,b = x

print (a)
print (b)
```

```
3
4
```

```python
a,b = [3,4]

print (a,b)

a,b = [b,a]

print (a,b)
```

```
3 4
4 3
```

```
x = [3, 4, "kali"]

y = [ "abcd", x]

print (y)

y = [ "abcd", *x]

print (y)
```

```
['abcd', [3, 4, 'kali']]
['abcd', 3, 4, 'kali']
```

```
x = "kalidas"

y = list(x)

print (y)
```

```
['k', 'a', 'l', 'i', 'd', 'a', 's']
```

comparison

```
x = [3,4]

y = [3,4]

print (x==y)
```

```
True
```

```
x = [3,4]

y = [4,1]

print (x < y)
```

```
True
```

```
x = [3,4]

y = [4,1]

print (x > y)
```

```
False
```

```
x = [3,4]

y = x

print (x == y)
```

```
True
```

Object reference check

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

```python
y = [3, 4]

print (x is y)
```

```
False
```

```python
x = [3, 4]

y = x

print (x is y)
```

```
True
```

If Statement

```python
i = 3

if i == 3 :
  print ("yes")
else :
  print ("no")
```

```
yes
```

```python
i = 4

if i == 1 :
  print ("one")
elif i == 2 :
  print ("two")
elif i == 3 :
  print ("three")
else :
  print ("none")
```

```
none
```

```python
i = "abc"

if i == "abc" :
  print ("yes")
else :
  print ("no")
```

```
yes
```

```python
i = "abc"

if i == "abc" :
  print ("yes")
```

```
yes
```

```python
i = "abc"

if i < "bcd" :
  print ("yes")
else :
  print ("no")
```

```
yes
```

```python
x = ["abc", 123, 23.45, -300]

if "abc" in x :
  print ("yes")
else :
  print ("no")
```

```
yes
```

```python
if True :
  print ("true")
```

```
true
```

```python
if 1 < 3 : print ("yes")
```

```
yes
```

```python
print ("yes") if 4 < 3 else print ("no")
```

```
no
```

While loop

```python
i = 3

n = 5

while n > 0 :
  print (n)
  n = n - 1
  i = 1
  i = 3
  # any number of statments

#any number of statements
```

```
5
4
3
2
1
```

```python
mysum = 0

n = 10

while n > 0 :
  mysum = mysum + n
  n = n - 1
  #
  #
  #

print (mysum)




```

```
55
```

```python
i = 3

n = 5

while n > 0 :

  if n == i :
    break

  print (n)
  n = n - 1
```

```
5
4
```

```python
i = 3

n = 5

while n > 0 :

  if n <= i :
    n = n -1 #try without this
    continue

  print (n)
  n = n - 1

print ("this is outside")
```

```
5
4
this is outside
```

```python
x = ["abc", "def", 12.3, 45.21, -800]

while len(x) > 0 :
  print (x)
  del x[0]

print (x)
```

```
['abc', 'def', 12.3, 45.21, -800]
['def', 12.3, 45.21, -800]
[12.3, 45.21, -800]
[45.21, -800]
[-800]
[]
```

For loop

```python
x = ["abc", "def", 12.3, 45.21, -800]

for ele in x :
  print (ele)
```

```
abc
def
12.3
45.21
```

self update issues

```python
x = ["abc", "abc", "def", 12.3, 45.21, 'abcd', -800, 23, 'kalidas']

for ele in x :
  print ('before removal:',ele, x)
  x.remove(ele)
  print ('after removal:',ele, x)

print ('final list:', x)
```

```
before removal: abc ['abc', 'abc', 'def', 12.3, 45.21, 'abcd', -800, 23, 'kalidas']
after removal: abc ['abc', 'def', 12.3, 45.21, 'abcd', -800, 23, 'kalidas']
before removal: def ['abc', 'def', 12.3, 45.21, 'abcd', -800, 23, 'kalidas']
after removal: def ['abc', 12.3, 45.21, 'abcd', -800, 23, 'kalidas']
before removal: 45.21 ['abc', 12.3, 45.21, 'abcd', -800, 23, 'kalidas']
after removal: 45.21 ['abc', 12.3, 'abcd', -800, 23, 'kalidas']
before removal: -800 ['abc', 12.3, 'abcd', -800, 23, 'kalidas']
after removal: -800 ['abc', 12.3, 'abcd', 23, 'kalidas']
before removal: kalidas ['abc', 12.3, 'abcd', 23, 'kalidas']
after removal: kalidas ['abc', 12.3, 'abcd', 23]
final list: ['abc', 12.3, 'abcd', 23]
```

nested for loops

```python
x = ["abc", "abc", "def", 12.3, 45.21, -800, [23, 67]]

pairs_list = []

for elem1 in x :
  for elem2 in x :
    pairs_list.append( (elem1,elem2) )

print (len(pairs_list))
```

```
49
```

(for clause) in lambda form

```python
x2 = [z for z in x]

print (x2)

print (x2 is x)

print (x2 == x)
```

```
['abc', 'abc', 'def', 12.3, 45.21, -800, [23, 67]]
False
True
```

```python
x = ["abc", "abc", "def", 12.3, 45.21, -800]

pair_list = []

print ([z for z in range(0,len(x))])

for i in range(0, len(x) ) :
  for j in range(i+1, len(x)) :
    pair_list.append( (x[i],x[j]) )

print (len(pair_list))
```

```
[0, 1, 2, 3, 4, 5]
15
```

Logical operators

```python
i = 1
j = 3

if i==1 and j==3 :
  print ("and")


if i==1 or j==4 :
  print ("or")


if i==2 or j==4 :
  print ("or 2")
```

```
and
or
```

```python
i = 1
j = 3
k = 0

print (i or j, j or i, k or i, not k or i, not j or i)


print (i and j, k and i, j and k, not k and i, i and not k)
```

```
1 3 1 True 1
3 0 0 1 True
```

Reading and writing from files

```python
fh = open("kalidas.txt","w")

fh.write("kali")

fh.close()
```

```python
fh = open("kalidas.txt","r")

x = fh.read()

print (x)

```

```
fh.close()
```

```
kali
```

```python
fh = open("kalidas.txt","w")

x = ["abc", 23.4, -800]

fh.write(x)

fh.close()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-88-ac92584c34c3> in <module>()
      3 x = ["abc", 23.4, -800]
      4 
----> 5 fh.write(x)
      6 
      7 fh.close()

TypeError: write() argument must be str, not list
```

SEARCH STACK OVERFLOW

```python
import json

x = ["abc", 23.4, -800]

fh = open("kalidas.txt","w")

json.dump(x,fh)

fh.close()
```

```python
import json

fh = open("kalidas.txt","r")

y = json.load(fh)

fh.close()

print (y)
```

```
['abc', 23.4, -800]
```

```
! cat kalidas.txt
```

```
["abc", 23.4, -800]
```

```
! echo "[1, 2, 3]" > kalidas.txt
```

```python
import json

fh = open("kalidas.txt","r")

y = json.load(fh)

fh.close()

print (y)
```

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

Basic Collections

Sets

```python
x = set([1,2,3, 1, 1, 1, "kalidas", "kali"])

print (x)
```

```
{1, 2, 3, 'kali', 'kalidas'}
```

```python
x = {1,2,3,"kali"}

print (x)
```

```
{1, 2, 3, 'kali'}
```

```python
x = {1,2,3,"kali"}

print ('before', x)

x.add('pqrst')
x.add(3.141234)
x.add(1)
x.add(2)

print ('after',x)
```

```
before {1, 2, 3, 'kali'}
after {1, 2, 3, 'pqrst', 'kali', 3.141234}
```

```python
x = {1, 2, 3, 3.141234, 'kali', 'pqrst'}

x.remove(1)

print (x)
```

```
{2, 3.141234, 3, 'kali', 'pqrst'}
```

```python
x = {1, 2, 3, 3.141234, 'kali', 'pqrst'}

x.remove(4)

print (x)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-98-ce754ef7528c> in <module>()
      1 x = {1, 2, 3, 3.141234, 'kali', 'pqrst'}
      2 
----> 3 x.remove(4)
      4 
      5 print (x)

KeyError: 4
```

```
x = {1, 2, 3, 3.141234, 'kali', 'pqrst'}

del x[0]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-8-1afddb017a6b> in <module>()
      1 x = {1, 2, 3, 3.141234, 'kali', 'pqrst'}
      2 
----> 3 del x[0]

TypeError: 'set' object doesn't support item deletion
```

SEARCH STACK OVERFLOW

```
x = {1, 2, 3, 3.141234, 'kali', 'pqrst'}

x.discard(4)

print (x)
```

```
{1, 2, 3.141234, 3, 'kali', 'pqrst'}
```

```
x = {1,2,3}

y = {2,3,4}

z = {5,6,7,8}

x2 = x.union({5,"kali",'das'})

print (x.intersection(y))

print (x.difference(y), y.difference(x))

print (x.union(y))

print (x.symmetric_difference(y))

print (x.isdisjoint(y), x.isdisjoint(z))

print (x.issubset(x2))
```

```
{2, 3}
{1} {4}
{1, 2, 3, 4}
{1, 4}
False True
True
```

Dictionaries (or Maps)

```python
x = {'abc': 11, 'bcd' : -3.4, 'cdef': "kalidas"}

print (x)
```

```
{'abc': 11, 'bcd': -3.4, 'cdef': 'kalidas'}
```

```python
x = {'abc': 11, 'bcd' : -3.4, 'cdef': "kalidas"}

print (x['abc'], x['cdef'])

key='bcd'

print (x[key])

print (x.get(key))
```

```
11 kalidas
-3.4
-3.4
```

```python
x = {'abc': 11, 'bcd' : -3.4, 'cdef': "kalidas"}

x['abc'] = 34

print (x)
```

```
{'abc': 34, 'bcd': -3.4, 'cdef': 'kalidas'}
```

```python
x = {'abc': 11, 'bcd' : -3.4, 'cdef': "kalidas"}

for ele in x :
  print (ele)
```

```
abc
bcd
cdef
```

```python
x = {'abc': 11, 'bcd' : -3.4, 'cdef': "kalidas"}

for val in x.values() :
  print (val)
```

```
11
-3.4
kalidas
```

```python
for k,v in x.items() :
  print ('key',k,'val',v)
```

```
key abc val 11
key bcd val -3.4
key cdef val kalidas
```

```python
x = {'abc': 11, 'bcd' : -3.4, 'cdef': "kalidas", 'pqrst' : 11, 'abc': 23}

print (len(x), x)
```

```
4 {'abc': 23, 'bcd': -3.4, 'cdef': 'kalidas', 'pqrst': 11}
```

```python
x = {'abc': 11, 'bcd' : -3.4, 'cdef': "kalidas", 'pqrst' : 11, 'abc': 23}

x['another'] = -900

print (x)
```

```
{'abc': 23, 'bcd': -3.4, 'cdef': 'kalidas', 'pqrst': 11, 'another': -900}
```

```python
x = {'abc': 11, 'bcd' : -3.4, 'cdef': "kalidas", 'pqrst' : 11, 'abc': 23}

z = x.pop('bcd')

print (z, x)
```

```
-3.4 {'abc': 23, 'cdef': 'kalidas', 'pqrst': 11}
```

```python
x = {'abc': 11, 'bcd' : -3.4, 'cdef': "kalidas", 'pqrst' : 11, 'abc': 23}

z = x.remove('bcd')

print (z, x)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-110-e58f37bc4d95> in <module>()
      1 x = {'abc': 11, 'bcd' : -3.4, 'cdef': "kalidas", 'pqrst' : 11, 'abc': 23}
      2 
----> 3 z = x.remove('bcd')
      4 
      5 print (z, x)

AttributeError: 'dict' object has no attribute 'remove'
```

SEARCH STACK OVERFLOW

```python
x = {'abc': 11, 'bcd' : -3.4, 'cdef': "kalidas", 'pqrst' : 11, 'abc': 23}

del x['bcd']

print (x)
```

```
{'abc': 23, 'cdef': 'kalidas', 'pqrst': 11}
```

```python
x = {'abc': 11, 'bcd' : -3.4, 'cdef': "kalidas", 'pqrst' : 11, 'abc': 23}

x.clear()

print (x)
```

```
{}
```

```python
x = {'abc': 11, 'bcd' : -3.4, 'cdef': "kalidas", 'pqrst' : 11, 'abc': 23}
```

```python

y = x

y['another'] = -900

print (x)
print (y)
```

```
{'abc': 23, 'bcd': -3.4, 'cdef': 'kalidas', 'pqrst': 11, 'another': -900}
{'abc': 23, 'bcd': -3.4, 'cdef': 'kalidas', 'pqrst': 11, 'another': -900}
```

```python
x = {'abc': 11, 'bcd' : -3.4, 'cdef': "kalidas", 'pqrst' : 11, 'abc': 23,
     'submap1' : {'abc': 11, 'bcd' : -3.4,
                  'subsubmap': {'abc': 11, 'bcd' : -3.4, 'cdef': "kalidas"} } }


import json

fh = open('kalidas.txt','w')
json.dump(x,fh)
fh.close()

!cat kalidas.txt
```

```
{"abc": 23, "bcd": -3.4, "cdef": "kalidas", "pqrst": 11, "submap1": {"abc": 11, "bcd'
```

```python
import json

fh = open('kalidas.txt','r')

x2 = json.load(fh)

fh.close()

print (x2)
```

```
{'abc': 23, 'bcd': -3.4, 'cdef': 'kalidas', 'pqrst': 11, 'submap1': {'abc': 11, 'bcd'
```

```python
x = {'abc': 11, 'bcd' : -3.4, 'cdef': "kalidas", 'pqrst' : 11, 'abc': 23}

y = x.copy()

y['another'] = -900

print (x)
print (y)
```

```
{'abc': 23, 'bcd': -3.4, 'cdef': 'kalidas', 'pqrst': 11}
{'abc': 23, 'bcd': -3.4, 'cdef': 'kalidas', 'pqrst': 11, 'another': -900}
```

```python
x = {'abc': 11, 'bcd' : -3.4, 'cdef': "kalidas", 'pqrst' : 11, 'abc': 23}

y = dict(x)

y['another'] = -900

print (x)
print (y)
```

```
{'abc': 23, 'bcd': -3.4, 'cdef': 'kalidas', 'pqrst': 11}
{'abc': 23, 'bcd': -3.4, 'cdef': 'kalidas', 'pqrst': 11, 'another': -900}
```

Function defintions

parameters

```
def f() :
  print ('kali')

f()
```

```
kali
```

```
def f() :
  return 1


print (f())
```

```
1
```

```
def f() :
  return 1

x = f()

print (x+2)
```

```
3
```

```
def f(a) :
  return 3*a


x = f(2)

print (x + 2)
```

```
8
```

```
def f(a,b) :
  tmp = a**2 + 2*a*b + b**2
  return tmp

x = f(2,3)

print (x)
```

```
25
```

```
def f(a) :
  tmp = [1,a,a*a,a*a*a]
  return tmp

x = f(2)

print (x)
```

```
[1, 2, 4, 8]
```

```
def f(a,b) :
  return a**3, 3* a**2 * b, 3*a * b**2, b**3

x = f(2,3)

print (x)

print (sum(x))
```

```
(8, 36, 54, 27)
125
```

```
def f(a,b) :
  tmp = a==b
  return tmp


print (f(2,3), f(2,2))
```

```
False True
```

recursion

```
def factorial(n) :
  if n==0 :
    return 1
  return factorial(n-1) * n

x = factorial(5)

print (x)
```

```
120
```

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

y = max(mylist)

mylist.remove(y)

z = [y] + mylist

print (y,mylist, z)
```

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

```
mylist = [1,2,3, [90000]]

y = max(mylist)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-128-630945db2818> in <module>()
      1 mylist = [1,2,3, [90000]]
      2 
----> 3 y = max(mylist)

TypeError: '>' not supported between instances of 'list' and 'int'
```

SEARCH STACK OVERFLOW

```python
def mysort(mylist) :
  if len(mylist) == 0 :
    return []

  y = max(mylist)

  mylist.remove(y)

  return [y] + mysort(mylist)


z = mysort([1,3,2,4,7,3,5])

print (z)
```

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

Lambda functions

simple lambda

```python
f = lambda x : x+2

print (f(2))
```

```
4
```

```python
f = lambda x : x[0] + x[1]*x[2]

print (f([2,1,3]))
```

```
5
```

```python
f = lambda : True

print (f())
```

```
True
```

```python
f = lambda x : True if x==1 else 45.2

print (f(1))

print (f(2))
```

```
True
45.2
```

```python
f = lambda x : [True, -4.65, 'kali'] if x==1 else 45.2

print (f(1))

print (f(2))
```

```
[True, -4.65, 'kali']
45.2
```

complex uses

```python
def myfun(a) :
  tmp = lambda x : x*a
  return tmp

doubler = myfun(2)

tripler = myfun(3)

print (doubler(4), tripler(4))
```

```
8 12
```

```python
tmp = [lambda x : x*2, lambda x : x*3, lambda x : x + 1000]

print (tmp[0](4), tmp[1](4) )

```

```
8 12
```

```python
tmp = [lambda x : x*2, lambda x : x*3, lambda x : x + 1000]

for myfun in tmp :
  print (myfun(12.3))
```

```
24.6
36.900000000000006
1012.3
```

filter, map, reduce functions

basic

```python
x = [5, 7, 22, 97, 54, 62, 77, 23, 73, 61]

y = list( filter(lambda a : a%2!=0, x) ) #filter-in (as against filter-out intuition)

print (y)
```

```
[5, 7, 97, 77, 23, 73, 61]
```

```python
x = [5, 7, 22, 97, 54, 62, 77, 23, 73, 61]

y = list( map(lambda a : a*2, x) )

print (y)
```

```
[10, 14, 44, 194, 108, 124, 154, 46, 146, 122]
```

```python
import functools

x = [5, 7, 22, 97, 54, 62, 77, 23, 73, 61]

y = functools.reduce( lambda a,b : a + b, x)

print (y)
```

```
481
```

popular lambda constructs using 'for' clause

```python
x = [5, 7, 22, 97, 54, 62, 77, 23, 73, 61]

y = [ a for a in x ]

print (y)

```

```
[5, 7, 22, 97, 54, 62, 77, 23, 73, 61]
```

```python
x = [5, 7, 22, 97, 54, 62, 77, 23, 73, 61]

y = [ a for a in x if a%2==1]

print (y)

```

```
[5, 7, 97, 77, 23, 73, 61]
```

```python
x = {'abc':1,'bcd':23,'cdef':-100}

y = {a:b for a,b in x.items() if b < 0}

print (y)
```

```
{'cdef': -100}
```

```python
x = {'a.b':1,'a.c':23,'b.a':-100}

y = {a:b for a,b in x.items() if a.startswith('a')}

print (y)
```

```
{'a.b': 1, 'a.c': 23}
```

Simple classes

```python
import math

class myclass :


  def __init__(self, x,y) :
    self.x = x
    self.y = y

  def mydist(self) :
    return math.sqrt(self.x **2 + self.y **2)


```

```python
a = myclass(1,2)
b = myclass(2,3)

print (a.mydist(), b.mydist())
```

```
2.23606797749979 3.605551275463989
```

```python
class myclass :
  def __init__(self,x,y) :
    self.x = x
    self.y = y

  def __str__(self) :
    return "x={0} y={1}".format(self.x,self.y)


a = myclass(2,3)

print (a)

```

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

```python
class myclass :
  def __init__(self,x,y) :
    self.x = x
    self.y = y

  def __repr__(self) :
    return "repr func: x={0} y={1}".format(self.x,self.y)
```

```python
a = myclass(1,2)

print (a)
```

```
repr func: x=1 y=2
```

```python
class myclass :
  def __init__(self,x,y) :
    self.x = x
    self.y = y

  def __repr__(self) :
    return "(x={0} y={1})".format(self.x,self.y)

  def __eq__(self,other):
    return self.x==other.x and self.y==other.y

  def __lt__(self,other) :
    if not self.__eq__(other) and self.x < other.x or self.y < other.y :
      return True
    return False

```

There is a bug in the logic here.. found?

```python
a = myclass(1,2)
b = myclass(1,2)

print (a is b, a==b, a<b, a>b)
```

```
False True False False
```

```
a = myclass(1,2)
b = myclass(1,3)

print (a<b)
```

```
True
```

```
a = myclass(1,2)
b = myclass(2,3)

print (a<b, a>b)
```

```
True False
```

```
a = myclass(1,3)
b = myclass(1,2)

print (a<b)
```

```
False
```

```

```

```
a = myclass(1,2)
b = myclass(1,3)
c = myclass(2,1)
d = myclass(2,2)
e = myclass(3,1)
a1 = myclass(1,2)

x = [e,d,c,b,a, a1]

print (x)

y = sorted(x)

print (y)

```

```
[(x=3 y=1), (x=2 y=2), (x=2 y=1), (x=1 y=3), (x=1 y=2), (x=1 y=2)]
[(x=1 y=2), (x=1 y=2), (x=1 y=3), (x=2 y=1), (x=2 y=2), (x=3 y=1)]
```

String functions

split function

```
x = "12,34,56"

y = x.split(',')

print (y)
```

```
['12', '34', '56']
```

```
x = "a.bc.de.efg.h.ijkl.m"
```

```
y = x.split('.')

print (y)
```

```
['a', 'bc', 'de', 'efg', 'h', 'ijkl', 'm']
```

```
x = "a.bc.de.efg.h.ijkl.m"

y = x.split('xyz') #try with ef

print (y)
```

```
['a.bc.de.efg.h.ijkl.m']
```

join function

```
x = [12, 34, 56]

y = ','.join(x)

print (y)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-152-8e8e926b90bb> in <module>()
      1 x = [12, 34, 56]
      2 
----> 3 y = ','.join(x)
      4 
      5 print (y)

TypeError: sequence item 0: expected str instance, int found
```

SEARCH STACK OVERFLOW

```
x = [12, 34, 56]

x2 = ['12','34','56']

y = ','.join(x2)

print (y)
```

```
12,34,56
```

```
print (','.join( [str(ele) for ele in x] ) )
```

```
12,34,56
```

Indexing

```
mystr = 'kalidas'

print (mystr[1:3])

print (mystr[:3])
print (mystr[[0,1]])
```

```
al
kal
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-158-600011cfc951> in <module>()
      5 print (mystr[:3])
      6 
----> 7 print (mystr[[0,1]])

TypeError: string indices must be integers
```

SEARCH STACK OVERFLOW

```
mystr = 'kalidas'

myindex_list = [0,1,3,2]

print ([mystr[i] for i in myindex_list])
```

```
['k', 'a', 'i', 'l']
```

Dynamic execution

```
text = "[1,2,3]"

print (eval(text))

x = eval(text)

print (type(x), x)
```

```
[1, 2, 3]
<class 'list'> [1, 2, 3]
```

```
text = "x = 3"

eval(text)
```

```
  File "<string>", line 1
    x = 3
      ^
SyntaxError: invalid syntax
```

SEARCH STACK OVERFLOW

```
text = "lambda x : x[0]**2 + x[1]**2"

x = eval(text)

print (x)

print ( x( (1,2) ) )
```

```
<function <lambda> at 0x7fa7821d48c8>
5
```

Run time

```python
x = 3
pass
y = 4

print (x,y)
```

```
3 4
```

```python
for i in range(4) :
  pass
  print (i)

print ('kali')
```

```
0
1
2
3
kali
```

```python
try :
  x = 1/0
  print (x)

except Exception as e :

  print ('Error', e)



```

```
Error division by zero
```

```python
try :
  x = {1, 2, 3}
  del x[0]

except Exception as e :

  print ('Error', e)
```

```
Error 'set' object doesn't support item deletion
```

```python
try :
  x = {'key1':'val1','key2':'val2'}

  del x['key1']

  print (x)

except Exception as e :

  print ('Error', e)
```

```
{'key2': 'val2'}
```

```python
try :
  x = {'key1':'val1','key2':'val2'}

  del x['key3']

  print (x)

except Exception as e :

  print ('Error', e)
```

```
Error 'key3'
```

```python
try :
  x = {'key1':'val1','key2':'val2'}

  del x['key3']

  print (x)

except Exception as e :

  pass

print ('as if nothing happened')
```

```
as if nothing happened
```

```python
def mydelete(x, key) :
  try :
    del x[key]
  except :
    pass

x = {'key1':'val1','key2':'val2'}

print (x)

mydelete(x,'key1')

print (x)

mydelete(x,'key3')

print (x)
```

```
{'key1': 'val1', 'key2': 'val2'}
{'key2': 'val2'}
{'key2': 'val2'}
```