

```

# 1
def add(a,b):
    return a+b
def power(a,b):
    return a**b

"""'flag' is Nothing just a variable that is holding boolean
value"""
flag = True
print(type(flag))
print((add if flag else power)(3,2))
print((add if not flag else power)(3,2))

# 2
""" 'is' expressions evaluate to True if two variables point
to the same object.
'==' evaluates to True if the objects referred to by the
variables are equal. """
a = [1,2,4]
b = [1,2,4]
a = b
print(a is b) #True
print(a == b) #True

c = list(a)
print(a == c) #True
print(a is c) #False

# 3
"""What is memory view"""
def myfunc(a,b):
    return a+b
funcs = [myfunc]
print(funcs[0]) # <function myfunc at 0x0087AD68>
print(myfunc(2,3)) # 5

```

```
# 4
variable = ['start', 'end', 'skip']
string = "Hello python"
print(string[::-1]) # reversing
print(string[::-1][::-1]) # reversing reversed string
print(string[::2]) # skipping one character
print(string[::3]) # skipping two character
print(string[2::-1])
print(string[4::-1])
```

```
# 6
# You can use if or dict for solve same problem
```

```
def dispatch_if(operator, x,y):
    if operator == 'add':
        return x+y
    elif operator == 'sub':
        return x-y
    elif operator == 'mul':
        return x*y
    elif operator == 'div':
        return x/y
    else:
        return None

def dispatch_dict(operator, x,y):
    return{
        'add': lambda : x+y,
        'sub': lambda : x-y,
        'mul': lambda : x*y,
        'div': lambda : x/y,
    }.get(operator,lambda: None)()

print(dispatch_if('mul',2,3))
print(dispatch_dict('mul',2,3))
print(dispatch_dict('unknown',2,3))
print(dispatch_if('unknown',2,3))
```

```
# 7
# If you need to permutation something then
import itertools
for p in itertools.permutations('HT'):
    print(p)

# 8
# collections.Counter will help you to find the most common
elements
import collections
c = collections.Counter("Hi i'm shakil. Shoshi is my
friend.")
print(c)
print(c.most_common())
print(c.most_common(2))

# 9
my_list = [1,2,3,4,5]
del my_list[:]
# if you use [:] this it can't give an error
print(my_list)

# You can replace all elements of a list without creating a
new list object
a = my_list
my_list[:] = [7,8,9]
print(my_list) #[7,8,9]
print(a) #[7,8,9]
print(a is my_list) #True

# You can create a (shallow) copy of a list
b = my_list[:]
print(b)
print(b is my_list) #False
```

```

# 10
import itertools
""" itertools.combinations gets 2 args.
1. is iterable
2. is length of combinations"""
print(itertools.combinations([1,2,3],2))
print(list(itertools.combinations([1,2,3,4],2)))
print(list(itertools.combinations([1,2,3,4,5,6],3)))

# 11
"""n = int(input("no of levels: "))
def xmas(n):
    for i in range(n):
        print(" "*(n-i-1) + "*" * (2*i+1))
print(xmas(n))"""

# 12
p = ['a', 'b', 'c', 'd']
b = [1,2,3,4]
print(p+b)
print(max(p))

pb = ['a', 'b', 'c', 'd', '3', '66']
# data type should be same while sorting
# even special character can be sorted
print(max(pb))
print(sorted([3,7,5] + [5,4,9,7.4]))

# 13
c = ['a', 'b', 'c', 'd']
d = ['g', 'h', 'i', 'j']
print("+".join(['a', 'b', 'c', 'd']))
print(c.count('a'))
c.extend(d)
print(c)

```

```
# 14
def add(a):
    return 2+a
print(add(3))

# 15
i = 6
# get all range till specified number in list
print(list(range(1,i+1)))

# 16
from functools import reduce
# multiply every element of list
print(reduce(lambda a,b: a*b, list(range(1,i+1))))

# 17
a = list(range(1,16))
# return lambda value which is x % 2 != 0
odd = list(filter(lambda x: x%2,a))
print(odd)
# return lambda value which is x%2 == 0
even = list(filter(lambda x: x%2 == 0,a))
print(even)
```

```
import turtle
trt1 = turtle.Turtle()
turtle.bgcolor("white")
trt1.speed(0)
turtle.title("Design")

def drawRainbow():
    for i in range(300):
        trt1.color("green")
        trt1.backward(i)
        trt1.left(110)
    trt1.color("white")
    trt1.hideturtle()
    trt1.setpos((20,0))
drawRainbow()
```

Random module in python. Any randomization can be done here with methods

```
import random
print(random.randrange(6))
```