In [6]:

```
# dictionary with duplicate values and keys
# keys (not applicable)
person = {
    "name": "virat",
    "international_team": "india",
    "national_team": "delhi",
    "nums": [1, 2, 3, 4]
}

print(person.get("nums")[2])
```

```
3
```

In [8]:

```
name = 'PIYUSH'
print(name[1::2])
```

```
IUH
```

In [16]:

```
# loops
# range -> by default it will take ending value(starting value = 0)
# print(range(5), type(range(5)))

print(list(range(5)))
print(list(range(2, 6)))
print(list(range(0, 30, 2)))
print(list(range(-8, -2)))
print(list(range(-8, -2, -1)))
print(list(range(-2, -8, -1)))
```

```
[0, 1, 2, 3, 4]
[2, 3, 4, 5]
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28]
[-8, -7, -6, -5, -4, -3]
[]
[-2, -3, -4, -5, -6, -7]
```

In [22]:

```
# for loop -> know the number of execution
# 1 to 10
for num in range(1, 11):
    print(num, end=" ")
```

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

In [24]:

```
print('ram' * 3)
```

```
ramramram
```

In [33]:

```
# string
name = 'Surya'
for letter in name:
    print(letter, end=' ')
```

```
S u r y a
```

In [57]:

```
# # list
# for num in [1, 2, 3]:
#     print(num)

# # tuple
# for num in (1, 2, 3, 4):
```

```
#     print(num)


# set
# chars = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O',
# for num in chars:
#     print(num)
# print(chars)


# dictionary
person = {
    "name": "virat",
    "team": "india",
    "national_team": "delhi",
}
# for key in person:
#     print(key, person.get(key))

# # keys
# for key in person.keys():
#     print(key, person.get(key))

# print(person.keys())

# values
# print(person.values())
# for value in person.values():
#     print(value)


# items
# print(person.items())
for key, value in person.items(): # unpacking
    print(f'{key} :  {value}')
```

```
name :  virat
team :  india
national_team :  delhi
```

In [63]:

```
# unpacking
a, b, c = [1, 2, 3]
print(a, b, c)
```

```
1 2 3
```

In [94]:

```
# 1 to 10 except 5 -> 1, 2, 3, 4, 6, 7, 8, 9, 10
# # continue
# for num in range(1, 11):
#     print(num, 'vasanth')
#     if num == 5:
#         continue
#     print(num)

# break
# 1 to 10 stop 5 -> 1, 2, 3, 4,
for num in range(1, 11):
    if num == 5:
        break
    print(num)
```

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

In [74]:

```
# nested loop
rows = 4
columns = 5
for row in range(1, rows):
    for col in range(1, columns):
        print(f'{row}{col}', end="  ")
    print()
```

```
11  12  13  14
21  22  23  24
31  32  33  34
```

In [92]:

```
# ***
# ***
# ***

# num = 5
# count = 0
# for row in range(num):
#     for col in range(num):
#         count += 1
#         print('*', end="")
#     print()


# print(count)

# optimised
# num = 5
# for row in range(num):
#     print("*" * 5)


# triangle
num = 5
```

```
for row in range(1, num+1):
    print(" " * (num-row), "*" * row)
```

```
     *
    **
   ***
  ****
 *****
```

In [96]:

```
# while loop -> condition check or entry checking loop
# 1 to 10

i = 1
while(i < 11):
    print(i)
    i += 1
```

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

In [100…

```
# terminal
while(True):
    command = input('c:/Desktop: ')
    if(command == 'add'):
        print('adding')
    elif(command == 'sub'):
        print('subtracting')
    elif(command == 'mul'):
        print('multiplying')
    elif(command == 'quit'):
        print('quitting...')
        print('thank you')
        break
    else:
        print(f'command: {command} not found')
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
Input In [100], in <cell line: 3>()
      1 # terminal
      2 while(True):
----> 3     command = input('c:/Desktop: ')
      4     if(command == 'add'):
      5         print('adding')

File ~\anaconda3\lib\site-packages\ipykernel\kernelbase.py:1075, in Kernel.raw_input
(self, prompt)
   1071 if not self._allow_stdin:
   1072     raise StdinNotImplementedError(
   1073         "raw_input was called, but this frontend does not support input reque
sts."
   1074     )
-> 1075 return self._input_request(
   1076     str(prompt),
   1077     self._parent_ident["shell"],
   1078     self.get_parent("shell"),
   1079     password=False,
   1080 )

File ~\anaconda3\lib\site-packages\ipykernel\kernelbase.py:1120, in Kernel._input_req
uest(self, prompt, ident, parent, password)
   1117             break
   1118 except KeyboardInterrupt:
   1119     # re-raise KeyboardInterrupt, to truncate traceback
-> 1120     raise KeyboardInterrupt("Interrupted by user") from None
   1121 except Exception:
   1122     self.log.warning("Invalid Message:", exc_info=True)

KeyboardInterrupt: Interrupted by user
```

In [107…

```
# functions -> verbs

# def get_sum(num1, num2):
#     print(num1 + num2)

# get_sum(4, 5)



def show_msg():
    print('Welcome!!! to intellipaat')

show_msg()
show_msg()
show_msg()
```

```
Welcome!!! to intellipaat
Welcome!!! to intellipaat
Welcome!!! to intellipaat
```

In [127…

```
# factorial -> 5! = 5*4*3*2*1 = 120

def get_factorial(num):
    fact = 1
    for i in range(1, num+1):
        fact = fact * i
    #     print(fact, i)
#     print(fact)
    return fact


# get_factorial(5)
# get_factorial(7)
# get_factorial(21)

fact_5 = get_factorial(5)
print(fact_5)
```

```
120
```

In [142…

```
# default values
def get_sum(n1 = 0, n2 = 0):
    return n1 + n2

print(get_sum(4, 5))
print(get_sum(4))
print(get_sum())

# print(get_sum(4, 3, 5))



# def get_3sum(n1, n2, n3):
#     return n1 + n2 + n3
```

```
9
4
0
```

In [152…

```
# args -> arguments
def get_sum(*nums): # *args(will be passed as tuple)
    total = 0
    for num in nums:
        total += num
    return total

print(get_sum(1, 3))
print(get_sum(1))
print(get_sum())
print(get_sum(1, 3, 4))
```

```
addding
addding
4
addding
1
0
addding
addding
addding
addding
addding
addding
33
```

In [162…

```
# def get_greeting_msg(name, roll_no):
#     print(f'Hi!!! {name}, Welcome to Intellipaat. Your Id is {roll_no}')
# # get_greeting_msg('vasi', 10)
# get_greeting_msg(roll_no = 10, name = 'vasi')



# **kwargs (key word arguments)
def get_greeting_msg(name = 'User', roll_no = 0):
    print(f'Hi!!! {name}, Welcome to Intellipaat. Your Id is {roll_no}')

# get_greeting_msg(roll_no = 10)
# get_greeting_msg()
# get_greeting_msg(roll_no = 10, name = 'vasi')

get_greeting_msg(roll_no=10, name='vasi', module='Python')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [162], in <cell line: 17>()
     11     print(f'Hi!!! {name}, Welcome to Intellipaat. Your Id is {roll_no}')
     13 # get_greeting_msg(roll_no = 10)
     14 # get_greeting_msg()
     15 # get_greeting_msg(roll_no = 10, name = 'vasi')
---> 17 get_greeting_msg(roll_no = 10, name = 'vasi', module='Python')

TypeError: get_greeting_msg() got an unexpected keyword argument 'module'
```

In [164…

```
# **kwargs (key word arguments)
def get_greeting_msg(**values):
    name = values.get('name')
    roll_no = values.get('roll_no')
    print(f'Hi!!! {name}, Welcome to Intellipaat. Your Id is {roll_no}')

get_greeting_msg(roll_no=10, name='vasi', module='Python')
```

```
Hi!!! vasi, Welcome to Intellipaat. Your Id is 10
```

In [166…

```
# def get_greeting_msg(name='user',roll_no):
#     print(f'hi {name}, welcome to intellipaat. your rollno. is {roll_no}')
# get_greeting_msg(roll_no=10)
```

```
  Input In [166]
    def get_greeting_msg(name='user',roll_no):
                                            ^
SyntaxError: non-default argument follows default argument
```

In [182…

```
# modules -> in built functions
import math

print(math.sqrt(3))
print(math.factorial(5))
print(math.ceil(3.567))
print(math.ceil(-3.567))
print(math.floor(3.567))
print(math.floor(-3.567))
print(math.pow(2.5, 3))


print(round(math.sqrt(3), 4))
```

```
1.7320508075688772
120
4
-3
3
-4
15.625
1.7321
```

In [188…

```
# def get_sum(n1, n2):
#     return n1 + n2

# lambda
# get_sum = lambda n1, n2: n1 + n2

# print(get_sum(4, 5))


# odd or even
is_even = lambda num: `

def is_even(num):
    if(num % 2 == 0):
        print('even')
    else:
        print('odd')

is_even(2)
is_even(3)
```

```
even
odd
```

In [191…

```
nums = [i for i in range(1, 101)]
print(nums)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 2
4, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 4
5, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 6
6, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 8
7, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100]
```

In [2]:

```
import arithmetic

total = arithmetic.get_sum(10, 15)
print(total)
```

```
25
```

In [ ]: