

JANUARY						
M	T	W	T	F	S	S
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		

JANUARY 2020

جمادی الاول ١٤٢١

Week :- 1 ( PYTHON BASICS )WEDNESDAY 15  
جمادی الاول ١٩PYTHON INTRO :-Pg WHAT IS PYTHON ?

→ Python is popular programming language

→ It is used for:-

★ web development      ★ software development

★ mathematics      ★ system scripting

WHAT CAN PYTON DO :-

→ used on server to create web application

→ used alongside software to create workflow

→ connect to database system - read and modify files.

→ used to handle big data and perform complex

mathematics

→ used for rapid prototyping

WHY PYTHON :-

→ Simple syntax

→ procedural way

→ On different platforms

→ Run on interpreter system.

THURSDAY 16  
جمادی الاول ٢٠ Evening

JANUARY 2020

جمادی الاول ١٤٢٠

JANUARY						
M	T	W	T	F	S	S
			1	2	3	4
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		

17 FRIDAY

جمادی الاول ٢٠

## Python Indentation :-

Refers to space at beginning of code line.

Example:-

1) if  $s > 5:$

    — print ("Five is greater than two!")

2) if  $s > 2:$

    ✓ — print ("Five is greater than two!")

    if  $s > 2:+$

X — print(Five is greater than two!)

Python comment:-

→ Used to explain python code

→ Used to prevent execution

18 SATURDAY

جمادی الاول ٢٢

19 SUNDAY

جمادی الاول ٢٣

→ Used to make code more readable

Single line:-

start with # and python will ignore them

Example:-

# This is a comment

print ("Hello, World!")

Multiline:-

Example:-

'''

This is a comment

written in more than just one line

print ("Hello world")

JANUARY						
M	T	W	T	F	S	S
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		

JANUARY 2020

الإثنين ٢٠ جانفي ٢٠٢٠

## Python variables:-

Variables are containers for storing data value.

### Creating variables:-

Assign value to it

x = 5

y = "John"

print(x)

print(y)

Note used to declare any particular type.

x = 4 # x is of type int

x = "Sally" # x is now of type str

print(x)

### Casting:-

Give some specific data type

x = str(3) # x will be '3'

y = int(3) # y will be 3

z = float(3) # z will be 3.0

Evening

### Get the type:-

Get the data type of variable with type() function

x = 5

y = "John"

print(type(x))

print(type(y))

Single or double quote:- (Same)

x = "John"

# is the same as

x = 'John'

### Case sensitive:-

Variable names are case sensitive

x = 4

X = "Sally"

# A(X) will not overwrite a(x)

Output:-

4

Sally

Evening

MONDAY  
January 20

January 20</

# JANUARY 2020

جاءی الاول ۱۴۴۱

JANUARY						
M	T	W	T	F	S	S
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		

22 WEDNESDAY جاءی الاول ۲۲

## Variable names :-

\* and y / age, carname, total-volume.

### Rules:-

- Start with letter or underscore character
- Cannot start with number
- Only contain alpha-numeric character or underscore.
- Are case sensitive (AGE and age)
- Cannot be any keyword.

Example:- \* myvar = "John" \* my\_var = "John"

\* -my\_var = "John" \* myVar = "John" \* MYVAR = "John"

\* myvar2 = "John"

### Camel case :-

myVariableName = "John"

### Pascal case :-

MyVariableName = "John"

23 THURSDAY جاءی الاول ۲۳

## Snake case :- my\_variable\_name = "John"

### Assign Multiple variable values :-

x, y, z = "Orange", "Banana", "Cherry"

print(x)

print(y)

print(z)

x = y = z = "Orange"

print(x)

print(y)

print(z)

### Unpack a list :-

fruits = ["apple", "orange", "cherry"]

x, y, z = fruits

print(x)

print(y)

print(z)

Evening

FEBRUARY						
M	T	W	T	F	S	S
				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	

FEBRUARY 2020

جمادى الثانى ١٣٢١

Output Variables:-

print() function

Example:-

→ `x = "Python is awesome"`  
`print(x)`

→ `x = "Python"`  
`y = "is"`  
`z = "awesome"`  
`print(x,y,z)`

→ `x = "Python"`  
`y = "is"`  
`z = "Awesome"`  
`print(x+y+z)`

→ `x = 5`  
`y = 10`  
`print(x+y) # 15`

Output:-

Python is awesome

Output :-

Python is awasome

Output:-

Python is awasome

Global variable :-

That are create outside of function

Example:-

→ `x = "awesome"`  
`def myfunc()`  
`print("Python is") + x`  
`myfunc()`

Create var outside and use inside

→ `x = "awesome"`  
`def myfunc()`

`x = "fantastic"`

`print("Python is" + x)`

`myfunc()`

`print("Python is" + x)`

Create var inside and use outside

FRIDAY 7  
جمادى الثانى

08

09

10

11

12

01

02

03

04

05

Evening

SUNDAY 9  
جمادى الثانى

06

07

08

09

10

11

12

01

02

03

04

05

Evening

M	T	W	T	F	S	S
					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	

## FEBRUARY 2020

جمادى الثاني ١٣٢١

10 MONDAY <sup>١٥ جمادى الثاني</sup> The global keyword :-

```

08 def myfunc():
09     global n
10     n = "fantastic"
11
12     myfunc()
13     print("Python is" + n)

```

01 PYTHON DATA TYPE :-02 Built in Data types :- (By default)

- Text type :- Str → Numeric types :- int, float, complex
- Sequence type :- list, tuple, range → Mapping types :- dict
- Set type :- set, frozenset → Boolean type :- bool
- Binary type :- bytes, bytearray, memoryview → None Type :- NoneType

Evening Setting Data types :-11 TUESDAY <sup>١٦ جمادى الثالثى</sup> n = "Hello World" #str

- 08 → n = 20 #int → n = ["apple", "banana", "cherry"] #list
- 09 → n = 20.5 #float → n = ("apple", "banana", "cherry") #tuple
- 10 → n = 1j #complex → n = range(6) #range
- 11 → n = True #bool → n = {"name": "John", "age": 36} #dict
- 12 → n = b"Hello" #bytes → n = {"apple", "banana", "cherry"} #set
- 01 → n = None #None type → n = memoryview(bytes(5)) #memoryview
- 02 → n = frozenset ({"apple", "banana", "cherry"}) #frozenset
- 03 → n = bytearray(5) #bytearray

04

05

Evening

FEBRUARY

M	T	W	T	F	S	S
					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	

FEBRUARY 2020

جمادى الثانى ١٣٢١

## PYTHON NUMBER :-

There are 3 numeric types in Python

### Int :-

Integer is whole, positive or negative, without decimal of unlimited length numbers.

#### Example :-

$x = 1$

$y = 3564312578943102$

$z = -3256785$

`print(type(x)) # int`

`print(type(y)) # int`

`print(type(z)) # int`

### Float :-

floating point number

$x = 1.0$

$y = 1.0$

$z = -35.59$

`print(type(x)) # float`

`print(type(y)) # float`

`print(type(z)) # float`

Can be scientific number with an

“e” to indicate

power of 10

$x = 35e3$

$y = 12E4$

$z = -87.7e100$

`print(type(x)(y)(z)) # float`

### Complex :-

Written with j as imaginary part

$x = 3 + 5j$

$y = 5j$

$z = -5j$

`print(type(x)) # complex`

`print(type(y)) # complex`

`print(type(z)) # complex`

WEDNESDAY 12  
جمادى الثانى

## FEBRUARY 2020

جمادى الثانى ١٣٢١

FEBRUARY						
M	T	W	T	F	S	S
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	

14 FRIDAY

١٩

Type conversion :-

```

08 x = 1 # int
09 y = 2.8 # float
10 z = 1j # complex
11 a = float(x) # int to float
12 b = int(y) # float to int
01 c = complex(x) # int to complex
02 print(a)
03 print(b)
04 print(c)
05 print(type(a))
Evening print(type(b))
print(type(c))

```

Output :-

1.0  
2  
(1+0j)  
class float  
class int  
class complex

15 SATURDAY ٢٠ جمادى الثانى

PYTHON CASTING :-Integer :-

```

08 x = int(1) # x will be 1
09 y = int(2.8) # y will be 2
11 z = int("3") # z will be 3

```

String :-

x = str("s1") # x will be 's1'  
y = str("2") # y will be '2'  
z = str(3.0) # z will be '3.0'

Float :-

```

01 x = float(1) # x will be 1.0
02 y = float(2.8) # y will be 2.8
03 z = float("3") # z will be 3.0
04 w = float("4.2") # w will be 4.2
05

```

Evening

FEBRUARY						
M	T	W	T	F	S	S
				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	

FEBRUARY 2020

IPM, Jalandhar

## Python string :-

### String :-

Single quotation mark , or double quotation mark.

#### Example:-

print ("Hello")

print ('Hello')

### Looping through String :-

for x in "banana":

    print (x)

### Quotes inside Quotes :-

#### Example:-

print ("It's alright")

print ("He is called 'Johnny'")

print ('He is called "Johnny"')

Output:-  
 b  
 a  
 n  
 a  
 n  
 a

### String length :-

a = "Hello, world!"  
 print (len(a)) #13

### Assign String to variable :-

#### Example:-

a = "Hello"

print (a)

### Check string :-

txt = "The best thing  
in life are free!"

print ("free" in txt)

#### if statement :-

txt = "The best thing --- free!"

if " free " in txt:

    print("Yes, 'free' is in text")

### Check is not :-

txt = "The best thing --- free!"

print ("expensive" not in txt)

#### if statement :-

if "expensive" not in txt:

    print("No, 'expensive' is not")

### Multiline String :-

#### Example:-

Use three double quotes

a = """ My name is Maryam,

doing software engineering,

In Comsts attack university. """

Can be in three single quotes

### Strings are array :-

a = "Hello, World!"

print [a[1])

Output:- e

MONDAY 17

Evening

08

09

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

01

02

03

04

05

06

07

08

09

10

11

12

13

14

</

FEBRUARY 2020

جمادى الثانى ١٤٢٠

FEBRUARY						
M	T	W	T	F	S	S
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	

19 WEDNESDAY

جمادى الثانى ١٤٢٠

## Slicing String :-

### Slicing :-

Range of character by using slice syntax

#### Example :-

b = "Hello, World!"

print(b[2:5]) # llo

### Slice from start :-

print(b[:5]) # Hello

### Slice to end :-

print(b[2:]) # llo, World!

### Modify Strings :-

#### Upper case :-

print("Hello,World!")

20 THURSDAY

جمادى الثانى ٢٠

print(a.upper())

#### Remove Whitespace :-

print(a.strip())

#### Split String :-

print(a.split(',')) # ['Hello', 'World']

#### String concatenation :-

We merge two string

a = "Hello"

b = "World"

c = a + b

print(c)

Evening

#### Negative indexing :-

print(b[-5:-2]) # orl

#### Lower case :-

print(a.lower())

#### Replace string :-

print(a.replace('H', 'J'))

#### To add space :-

a = "Hello"

b = "World"

c = a + " " + b

print(c)

FEBRUARY

M	T	W	T	F	S	S
					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	

FEBRUARY 2020

جمادى الثانى ١٣٢١

## Format String :-

Create f string

age = 36

txt = f "My name is John, I am {age}"

## Placeholders and Modifiers :-

price = \$9

txt = f "The price is {price} dollars"

print (txt)

## for modifying:-

text = f "The price is {price:.2f} dollars"

print (text)

## Contain python code:-

txt = f "The price is {20 \* 35} dollars"

## Escape characters:-

backslash (\) followed the character

## Example:-

txt = "We are the so-called \"Vikings\" from mouth."

Code      Result

\' Single Quote

\\" Backslash

\n New line

\r Carriage return

\t Tab

\b backspace

\f Form feed

\ooo octal value

\xhh hex value

SUNDAY 23 جمادى الثانى ٢٧ 22

Evening

SATURDAY 22 جمادى الثانى ٢٦

Evening

# FEBRUARY 2020

جاء في الثاني عشر

FEBRUARY						
M	T	W	T	F	S	S
					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	

24 MONDAY

جاء في الثاني عشر

## String method :-

Python has set of built in methods that you can use on string.

### Method

capitalize()

#### Description

Converts the first character to uppercase

casefold()

Converts string into lower

center()

Return number of times specified value occur in string

count()

Return a centered string

encode()

Return encoded version of string

endswith()

Return true if the string end with specified value

ends with()

Re — Same

Evening expandtabs()

Set a tab size of string

find()

Searches the string of specified value and

returns position of where it is found.

25 TUESDAY

جاء في الثاني عشر

format()

formats specified value in string.

format-map()

formats specified value in string

index()

Searches string for specified value and returns position of where it was found

isalnum()

Returns true if all characters in string are alphanumeric

isdigit()

Returns true if all character in string are decimal

isidentifier()

Returns true if string is identifier.

islower()

Return true if all characters in string are lowercase.

isnumeric()

// // // // // // // // numeric

isprintable()

// // // // // // // // are printable.

Evening is Space()

// // // // // // // // whitespaces

APRIL						
M	T	W	T	F	S	S
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			

APRIL 2020

شعبان ١٢٣١

WEDNESDAY 8

`istitle()`

Return true if string follows the rule of title.

`isupper()`

Return true if all characters in string are uppercase

`join()`

join element of iterable to end of string.

`ljust()`

Return left justified version of string.

`lower()`

convert string into lowercase

`lstrip()`

Return a left trim version of string

`maketrans()`

Return a translation table to be used in translation.

`partition()`

Return tuple where string is parted into 3 parts.

`replace()`

Value is changed with another specified value.

08

09

10

11

12

01

02

03

04

05

Evening

THURSDAY 9

08

09

10

11

12

01

02

03

04

05

Evening

APRIL 2020

شعبان ١٤٢١

APRIL						
M	T	W	T	F	S	S
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			

15 WEDNESDAY

## PYTHON BOOLEANS

Represent in two values

Example :-

```
print (10 > 9) # True
print (10 == 9) # False
print (10 < 9) # False
```

a = 200

b = 33

if b > a:

print ("b is greater than a")  
else:

print ("b is not greater than a")

True and False

In function :-

```
def myfunc():
    return True
if myfunc():
    print ("YES!")
else:
```

print ("No!")

Isinstance () :-

x = 200

- print (isinstance (x,int))

Evaluate Value and Variable :-

16 THURSDAY

print (bool ("Hello"))

print (bool (15))

## PYTHON OPERATORS:-

print (10+5) # 15

Arithmetic operator :-

+	Addition	$x+y$	=	Equal	$x=y$
-	Subtraction	$x-y$	!=	Not equal	$x \neq y$
*	Multiplication	$x*y$	>	Greater than	$x > y$
/	Division	$x/y$	<	Lesser than	$x < y$
%	Modulus	$x \% y$	>=	Greater and equal	$x \geq y$
**	Exponentiation	$x^{**}y$	<=	Less and equal	$x \leq y$
//	Floor division	$x//y$			

Comparison operator :-

APRIL						
M	T	W	T	F	S	S
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			

APRIL 2020

شعبان ١٤٣١

## Assignment operator:-

=	$x = 5$
+=	$x = x + 3$
-=	$x = x - 3$
*=	$x = x * 3$
/=	$x = x / 3$
%=	$x = x \% 3$
//=	$x = x // 3$
**=	$x = x ** 3$
/=	$x = x // 3$
!=	$x = x != 3$
==	$x = x == 3$
>=	$x = x >= 3$
<=	$x = x <= 3$
:=	$x = 3 \text{ print}(x)$

## Membership operator:-

in	True if sequence in	ning
		with specific value is
not in	Sequence with specific	x not in
	value is not present	

## Operator Precedence:-

( )	<< >>	not
**	^	and
+x -x ~x	^	or
* / // % .		
+ -	=, !=, >, >=, <, <=	is, is not, in, not, in

## Logical Operator:-

and	Return true if both are true	$x < 5 \& \& x < 10$
01	Return true if one of statement is true.	$x < 5 \text{ or } x < 10$
not	Reverse the result	$\text{not}(x < 5 \& x < 10)$
04	Identity Operators:-	
ps	True if both have same variable	$x \text{ is } y$
is not	Not same	$x \text{ is not } y$

SUNDAY 19 شعبان ٢٢ SATURDAY 18 شعبان ٢٣

## Bitwise operators:-

^	AND	08
	OR	09
^	XOR	10
~	NOT	11
<<	Zero fill left shift	12
>>	Signed right shift	13

APRIL 2020

جوان ۱۴۴۱

APRIL						
M	T	W	T	F	S	S
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			

20

MONDAY

PYTHON IF.... ELSE :-If statement :-

a = 33

b = 200

if b &gt; a :

print ("b is greater than a")

Else statement :-

a = 200

b = 33

if b &gt; a :

print ("b is greater than a")

elif b == a :

print ("a and b are equal")

21 TUESDAY

else:

print ("b is less than a")

Nested if :-

x = 41

if x &gt; 10 :

print ("Above total!")

if x &gt; 20

print ("Above 20!")

else:

print ("Not above 20!")

05

Evening

Elif statement :-

a = 33

b = 33

if b &gt; a :

print ("b is greater than a")

elif a == b :

print ("a and b are equal")

Short Hand if... else :-

a = 2

b = 330

print ("A") if a &gt; b else print ("B")

a = 33

b = 33

print ("A") if a &gt; b else print ("=") if a == b else print ("B")

Pass statement :-

a = 33

b = 200

if b &gt; a :

pass

APRIL

M	T	W	T	F	S	S
			1	2	3	4
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			

APRIL 2020

رمضان ١٤٢١

## PYTHON MATCH :-

### Syntax :-

match expression :

case x :

    code block

case y :

    code block

### Combine value :-

day = 4

match day :

case 1|2|3|4|5 :

    print ("Today is weekday") (if)

case 6|7 :

    print ("Today is weekend")

## PYTHON WHILE LOOP :-

i = 1

while i < 6 :

    print(i)

    i += 1

### Else statement :-

i = 1

while i < 6 :

    print(i)

    i += 1

else :

    print ("i is no longer less than 6")

### Example :-

day = 4

match day :

case 1 :

    print ("Monday")

case 7 :

    print ("Sunday")

MONDAY 27

## Break statement :-

i = 1

while i < 6 :

    print(i)

    if i == 3

        break (Same continuous)

i += 1

Evening

APRIL 2020

(رمضان ١٤٣١)

APRIL						
M	T	W	T	F	S	S
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			

29 WEDNESDAY

PYTHON FOR LOOP

fruits = ["apple", "banana", "cherry"]  
 for x in fruits:  
     print(x) # through String

Break statement:-

for x in fruits:  
     print(x)  
     if x == "banana":  
         break

Same for continue

Nested loop:-

Evening for x in adj:  
     for y in fruits:

Range function:-

for x in range(6):  
     print(x)  
     for x in range(2, 30, 3):  
         else:

Start from 2 and 3 gap

Pass statement:-

for x in (0, 1, 2):  
     pass

30 THURSDAY

رمضان ٦

print(x, y)

PYTHON FUNCTION

block of code which only runs when it is called

Example:-

def my\_function(): # function is created  
     print("Hello from a function")

my\_function() # function is called

Arguments:-

def myfunc(fname):  
     print(fname + "Refsnes")

myfunc("Email")

myfunc("Tobias")

myfunc("Linus")

Number of argument:-

def my\_func(fname, lname):  
     print(fname + " " + lname)

myfunc("Email", "Refsnes")

M	T	W	T	F	S	S
				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

MAY 2020

رمضان ١٤٣١

## Arbitrary argument :-

(args)

def my\_func(\*kids):

print("The youngest child is " + kid[2])

my\_func("Email", "Tobias", "Linus")

FRIDAY 1  
رمضان

## Keyword argument :-

key = value syntax

def my\_func(child3, child2, child1)

print("The youngest child is " + child3)

my\_func(child1="Email", child2="Tobias", child3="Linus")

## Arbitrary keyword argument :-

(\*\*kwargs)

def my\_func(\*\*kid):

print("His last name is " + kid["lname"])

my\_func(fname="Tobias", lname="Refsnes")

## Default Parameter Value :-

SUNDAY 3 SATURDAY 2  
رمضان ٢

def my\_func(country="Norway")

print("I am from " + country)

my\_func("Sweden")

my\_func("India")

my\_func()

my\_func("Brazil")

## Passing a list as argument :-

def my\_func(food):

for n in food:

print(n)

fruits = ["banana", "apple", "cherry"]

my\_func(fruits)

MAY 2020

رمضان ١٤٣١

MAY						
M	T	W	T	F	S	S
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

4 MONDAY  
ال一个星期一

Return value :-

```

08 def my_func(x):
09     return 5*x
10 print(my_func(3))
11 print(my_func(5))
12 print(my_func(9))

```

Keyword only argument :-

```

02 def my_func(*, x)
03     print(x)
04 my_func(x=3)
05

```

Evening

Recursion :-

def tri\_recursion(k):

if (k > 0):

result = k + tri\_recursion(k-1)

print(result)

else:

result = 0

return result

01 Output -

print ("Recursion example result: ")

02 1

tri\_recursion(6)

03 3

04 6

05 10

Evening 15

21

Pass statement :-

```

def myfunc():
    pass

```

Positional only argument :-

```

def my_func(x, /):
    print(x)

```

Combine Positional only and Keyword only :-

```

def my_func(a,b,/,*,c,d):
    print(a+b+c+d)
my_func(5,6,c=7,d=8)

```