

Headings:

python programming

python programming

python programming

python programming

python programming

python programming

```
In [3]: print("Hello!!!")
```

Hello!!!

List of programming languages:(Ordered list)

1. python
2. java
3. c
4. c++
5. .NET

List of programming languages:(Unordered list)

- python
 - core python(tab space)
 - Adv python
- java
- c
- c++
- .NET

- **Python Programming**
- *Python Programming*

[click here](#)



1. Different paradigms
2. Interpreted PL
3. Open source
4. Object Oriented PL
5. Dynamic language

Guido Van Rossum

```
In [ ]:
```

Numeric Data types:

1. int
2. float
3. complex

```
In [8]: a = 10 #integer
b = 23.34 #float
c = 12+3j #Complex number
print(type(a),type(b),type(c))

<class 'int'> <class 'float'> <class 'complex'>
```

```
In [ ]: c1=4+5j
c2=complex(6,8)
print(c2)
```

```
In [1]: a=int(input("enter a"))
b=int(input("enter b"))
print(a+b)
print(a-b)
print(a*b)
print(a/b)
```

enter a10
enter b20
30
-10
200
0.5

```
In [4]: a=int(input("enter a"))
b=int(input("enter b"))
print(a+b,end=" ")
print(a-b,a*b,a/b,sep=",")
```

enter a10
enter b20
30 -10,200,0.5

Operators:symbol which perform some specific task

Operands:values given to operator are known as operands

```
In [5]: #Arithmetic operators:+,-,*,/,%,/(floor division),**(exponent)
a = int(input("enter a value"))
b = int(input("enter b value"))
print(a + b)
print(a - b)
print(a * b)
print(a / b)#float quotient
print(a % b)#Remainder
print(a // b)#int quotient
print(a ** b)# 2^5
```

enter a value2
enter b value5
7
-3
10
0.4
2
0
32

```
In [7]: #Relational operators:<,>,>=,<=,!=,==
a = int(input("enter a value"))
b = int(input("enter b value"))
print(a > b)
print(a < b)
print(a >= b)
print(a <= b)
print(a != b)
print(a == b)
```

enter a value2
enter b value5
False
True
False
True
True
False

```
In [9]: #Logical operator: and,or,not
print(help("keywords"))
```

Here is a list of the Python keywords. Enter any keyword to get more help.

False	class	from	or
None	continue	global	pass
True	def	if	raise
and	del	import	return
as	elif	in	try
assert	else	is	while
async	except	lambda	with
await	finally	nonlocal	yield
break	for	not	

None

```
In [11]: #Logical operator: and,or,not--> return type -boolean
#print(help("keywords"))
print(a < b and a > 1)
print(a < b or a > 1)
print(not a < b )
```

True
True
False

```
In [14]: #Membership operators : in,not in
list1 = [12,23,34,45,56,67]
print(12 in list1)
12 not in list1
```

True

Out[14]: False

```
In [17]: #Conditional statements: if,else,elif
if a < b:
    print("a is less than b")
else:
    print("b is less than b")
```

a is less than b

```
In [18]: if a < b:
        print("a is less than b")
elif a>b:
        print("b is less than a")
else:
        print("a equals to b")
```

a is less than b

```
In [19]: a = int(input("enter a value"))
b = int(input("enter b value"))
c = int(input("enter c value"))
if a>b and a>c:
    print("a is largest")
elif b>a and b>c:
    print("b is largest")
else:
    print("c is largest")
```

enter a value2
enter b value1
enter c value5
c is largest

```
In [22]: #Loops : while,for
# print 1 to n values
n=int(input("enter n value"))
i=1
while i<=n:
    print(i,end=" ")
    i=i+1
```

enter n value5
1 2 3 4 5

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```