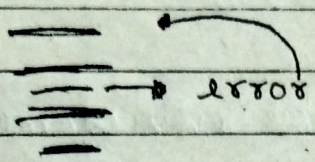
Date 10/04/23
Page No.

PYTHON — 1991

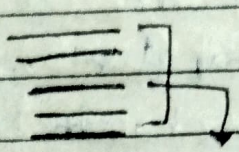
Interpreter



line by line code check

Ex: JS
Python

Compiled



Whole code check at once.

Ex: TS.
C++

```
a = 5
b = 7
print(a+b)    // Used for output
```

* Indentation

4 spaces
int main() ✗ Not in python

```
    code
```

4 space

{ } curly braces
Not used in Python

Date / /
Page No.

▶ Where we used python :

- IDLE : Integrated development learning environment
- VS CODE - IDE : Integrated development environment
- Spider
- Jupyter
- Pycharm - Used to make environment.

▶ Variables

```
a = 5
print (type(a))
```

→ int

▶ Keywords : Predefined words

→ ~~33~~ 35

▶ Data Types

32 = int
1.5 = float
"c" = string
"dog" = string
True/False = boolean [capital T & F]
→ Bool
Complex = real + Imaginary
2 + 5j
[j, J]

Date / /
Page No.

Comments

Single line Comment : #

Multiline Comment : """ """ / ''''''' '''''''

- Take Input from user // Always return string

```
a = input("Enter your name")

print(a)
print(type(a)) ⟶ return string

a = int(input("Enter your age"))

print(type(a)) ⟶ return int

a = complex(input("Enter age"))

print(type(a)) ⟶ return _+0j
                       (complex)
```

- If we used float ⟶ return _.0
                          (float)

a = 5 (Assign)
a == 5 (compare)

Date ./ ./
Page No.

Conditional statement : if elif else

```
a = 5
if (a>5):
    print ("a is greater than 5")
elif (a<5):
    print ("a is smaller than 5")
else:
    print ("a is equals to 5")
```

Indentation

Loops [initialisation, condition, updation]

Types of Loop
- while loop & for loop

- while

```
i = 0
while ( i <= 10):
    print(i)
    i = i+1
```

- for

```
for i in range (2, 21, 2):
    print(i)
```

- range (0, 21, 1)
  - Starting pt
  - jump

Date / /
Page No.

Questions

1. Python program to find the sum of first 10 Natural numbers?
2. Multiplication table of user given number?
   Hint: 2 * 1 = 2
3. Python program to take the age of user as input and check whether the user is school student, college student & working professional.
4. From 1 to 100 find the number which are divisible by 5 & 7?

Operators

1) Assignment : +, -, *, /, %, **

2) Comparison : ==, <>, >=, <, <= (return True or False)

3) Logical : and, or, not

4) Ternary operator:

```
a = 12
ticket = 20 if (a > 6) else 10
print(ticket)
```

→ truncate // floor division //

print(22//7) 3.14

return 3

Memory location & value

check only one value

Date / /

Page No.

- Membership in , not in

```
a = 1
b = 1,2,3,4
print (a in b)
```

return True or False

- Identity is , is not

```
a = 1,2,3
b = 1,2,3
print (a is b)
```

return True or False

▶ functions : Block of code

- Predefined
- User defined
  - parametrized
  - non-parametrized

def Keyword used for making userdefined function.

-

```
def sach():
    print("")

sach()
```

```
def sach()
    print("")

sach();
```

- parametrized

```
def sach(a,b):
    print(a+b)
sach(5,5)
```

parameter (pass fun definition time).

argument (pass calling time)

Date 12/04/23
Page No.

Questions

1) Take two no. in an array & swap the two numbers?
2) Take three no. input from user & find the largest no?
3) Program to reverse a string. take 5 no. input from user & multiply using the function.
4) Take string input from the user & a character & check whether the character is present in the string or not?

Recursion → function call itself again & again

```
def happy(a)
    if(a>0):
        return a + happy(a-1)
    else:
        return 0
print(happy(5))
```

→ Sum of 10 Natural No:-

```
sum = 0
for i in range(1, 11):
    sum = sum + i
print(sum)
```

Date / /
Page No.

→ Multiplication Table.

```
a = int(input("Enter any number:"))

for i in range(1, 11)
    print(a, "x", i, "=", a*i)
```

→ school, college or working professional.

```
a = int(input("Enter your age :"))
if (a>0 and a<=20):
    print("You are school student")
elif(a>20 and a<=26):
    print("You are college student")
elif(a>26 and a<=50):
    print("You are working professional")
else:
    print("You are to old now take retairment")
```

→ 1 to 100 find divisible 5 or 7.

```
print("Divisible by 5 and 7 between 1 to 100:")
for i in range(1,101):
    if(i%5==0 and i%7==0)
        print(i)
```

→ Take two no. input in array & swap :-

```
a = [10, 20]
temp = a[0]
a[0] = a[1]
a[1] = temp
print(a)
```

Date / /
Page No.

→ largest no. b/w three no.s

```
a = int(input("Enter first number:"))
b = int(input("Enter second number:"))
c = int(input("Enter third number:"))

def check(a,b,c)
 if(a>b):
    print("First number is largest:",a)
    if(a>c):
      print("First number is largest:",a)
    else:
      print("Third number is largest:",c)
 elif(b>c):
    print("second number is largest:",b)
 else:
    print("Third number is largest:",c)

check(a,b,c)
```

→ Reverse a string

```
a = input("Enter your name:")
l = len(a)
for i in range(l-1,-1,-1):
    print(a[i])
```

→ Take 5 no. from user & multiply

```
a = int(input("Enter 1st number:"))
b = int(input("Enter 2nd number:"))
c = int(input("Enter 3rd number:"))
d = int(input("Enter 4th number:"))
e = int(input("Enter 5th number:"))
def cal(a,b,c,d,e)
  print("Total is :",a*b*c*d*e)

cal(a,b,c,d,e)
```

Date / /
Page No.

→ Take string & check character present or not.

```
a = input("Enter your name :")
b = input("Enter any character:")

for i in a :
    if (i==b):
        print("character is present")
```

→ Check length of string:

```
a = input("Enter your name :")
count = 0
for i in a
    count = count + 1
print(f"character comes {count} times")
```

→ Reverse 50 to 1

```
for i in range(50, 0, -1):
    print(i)
```

→ Reverse a number.

```
a = [1,2,3,4,5]
t = len(a)
print(t)
for i in range(t-1, -1, -1):
    print(a[i])
```

[array] stores specific datatype (Homogeneous datatype)

Date

Page No.

Data Structures

- List [ ] → stores all datatyes [Mutable] ← can be changed
- Tuple
- Dictionary
- Set { }

Empty list

- li = [ ]
- a = list( ) → only one argument. → List constructor

a = list([1,2,3,4]) ✓

a = list(1,2) ✗

Methods in List

- slicing

a[1 : 3] (n-1) upper

1 → starting value

[1 : 3 : _] jump

[: : 2] 2 ka jump

- append [Add data last of list]

a.append( )

for i in range(1, 101):

a.append(i)

- copy [copy data]

b = a.copy

- count [1 कितनी बार है]

a.count(1)

- extend b = [ ]

a.extend(b) → a.extend([6])

print(a).

- Index (check index of data)

a.index(2)

Bake index shift हो जाएगा।

Date / /
Page No.

- Insert (specific index value add)

a.insert (, ,)
index item

- pop (remove last element)

a.pop() print(a.pop())
return which value delete

- remove (specific element remove)

a.remove (,)
1 argument value [1,2] delete

- reverse

a.reverse()

- sort

a.sort (reverse = True)
False

a.sort ( )

▶ **Tuple** : immutable , Hetrogeneous data , Indexing

- ()

- a=() → Empty tuple
- a = tuple() → constructor tuple

▶ 1st Method
convert tuple to list for adding data

a = (1,2,3,4,5,6)
b = list(a)
print(b)

b = (45) return int

b = (45,) return tuple

Date / /

Page No.

print (a[0])

Slicing

print (a[3:4])

2nd Method { Jugad }

a = (1, 2, 3, 4, 5, 6)

[:2] [2:]

1, 2, 99, 3, 4, 5, 6

print (a[:2] + (99,) + a [2:])

- Methods of tuple

- count // count no. of occurence of value

print (a.count(4))

- Index // Given index of value

print (a.index(3))

Questions

1) Python program to interchange first & last element in the list

```
a = [1, 2, 3, 4, 5]
temp = a[0]
a[0] = a[-1]
a[-1] = temp
for i in a:
    print(a[i])
```

Date / /
Page No.

2) Find length of list without using len().

```
a = [1,2,3,4,5]
count = 0
for i in a:
    count = count + 1
print(f"Length of list is : {count}")
```

3) Sum of all integer elements in the list

```
a = [1,2,3,4,5]
sum = 0
for i in a:
    sum = sum + i
print(f"sum is : {sum}")
```

4) Take list & check occurrence of at least one element

```
a = [1,1,2,2,3,4,4,3,5]
b = int(input("Enter any number :"))
print(f"Element is : {b}")
count = 0
for i in a:
    if(i==b):
        count = count + 1
print(f"Number of occurence : {count}")
```

Decending order : a[j] < a[K]

Accending Order

```
l = []
for i in range(1,6):
    a = int(input(f"Enter {i} number : "))
    l.append(a)

for j in range(0, len(l)):
    for K in range(j+1, len(l)):
        if(l[j] > l[K]):
            temp = l[j]
            l[j] = l[K]
            l[K] = temp
print(f"After sorting : {l}")
```

duplicate

```
for j in range(0, len(l)):
    for K in range(j+1, len(l)):
        if(l[j] == l[K]):
            print(l[K])
```

2nd smallest or largest

```
print(l[K-1])
```

- for largest No : l[j]
- for smallest No : decending order + l[j]

▶ Dictionary { } , Mutable , follow Indexing

- Works on Key & value [key should not same]

dict = { "Ram" : 45, "Name" : "sach" }

▶ Empty Dict

Dict = { }

b = dict()

print(dict["key"])

dict = { "Name": ["a", "b", "c"], "class": [1, 2, 3, 4, 5] }

Nested Dictionary

a = { "class1": ["Preeti", "Pooja"],
"class2": ["Anshul": 24, "sach": 31]

▶ **Sets** : { } , unordered, immutable,

```
a = {1, 2, 3, 4, 5, 8}
b = {1, 2, 3, 4, 5, 99}

print(a)
```

set Iteration

```
for i in a:
    print(i)
```

Methods of Sets :

(1) Pop() // delete any element

```
print(a.pop())
```

(2) remove() // delete by value

```
a.remove(3)
print(a)
```

(3) discard() // delete by value

```
a.discard(i)
print(a)
```

(4) clear() // delete all value & return set()

```
a.clear()
print(a)
```

Date / /
Page No.

(5) add() // Add value in set anywhere

```
a.add(6)
print(a)
```

(6) union() // Add two or more set

```
print(a.union(b))
```

(7) intersection() // Common element comes

```
print(a.intersection(b))
```

(8) issubset() // a ~~के सभी~~ element b ~~में है~~ return True or False

```
print(a.issubset(b))
```

(9) ~~[illegible]~~ issuperset() //

```
print(a.issuperset(b))
```

(10) update() // add set b in set a

```
a.update(b)
print(a)
```

(11) difference() // a-b return 8

```
c=a.difference(b)
print(c)
```

(12) symmetric_difference // a-b return 8,99

```
c=a.symmetric_difference(b)
print(c)
```

Date / /

Page No.

* 4 Number methods

56.51 → 57

- print(round(56.50)) // return 56 roundoff value
- print(abs(-234)) // give +ve value (234)
- print(divmod(35,3)) // Give reminder
- print(pow(3,3)) // $(3)^3$
- print(3**3) // $(3)^3$

* lambda function // lambda (keyword), small function

```
x = lambda y: y*y
print(x(6))
```

return 36

* Input 2 or mor numbers without using loop

```
a,b,c,d = map(int, input("enter No :").split())
print(a,b,c,d)
```

* List comprehension

Print value in list[]

```
a = [b for b in range(0,10)]
print(a)
```

* dict comprehension (Key value pair)

```
a = {i: i*i for i in range(0,10)}
print(a)
```

return 0:0, 1:1, 2:4, 3:9

* String formating

```
name = "sach"
rollno = 34
print(f"My name {name} and Roll No is {34}")
```

Date / /
Page No.

print("My name {a} & roll no is {b}".format(a= name, b=rollno))

print("My name {0} & roll no is {1}".format("sach", 34))

* Call by value // value change Nahi Hota

```
def sum(a):
    a=a+10
    print(a)      // 15
a=5
print(a)          // 5
sum(a)
print(a)          // 5
```

* Call by reference // value change ho jaega.

```
def sum(a):
    a.append(86)
    print(a)      // -
a=[1,2,3]
print(a)
sum(a)
print(a)          // [1,2,3,86]
```

* assert // Testing technique

assert(16>9), "the value is greater than 6"

If false Hua toh

return assertion error: the value is greater than 6

Date / /
Page No.

* random (Module)

```
import random as r
print(random.randint(1,100)

a = ["apple", "grapes", "banana", "grapes", 1, 2, 3]

r.shuffle(a)

print(r.choice(a))      // Random element निकलेगा
print(r.sample(a,5))    // Randomly 5 element Niklaga
print(a)
```

* filter

```
def check(a):
    if(a%2==0):
        return a

print(list(filter(check, [1,2,3,4,5,6])))
```

return [2, 4, 6]

* enumerate

```
a = ["sach", "kiara", "kirti"]

for i, j in enumerate(a,1):
    print(i, j)
```

return 1 sach, 2 kiara

* reduce // like Recursion

```
from functools import reduce
print(reduce((lambda x, y: x*y), [1,2,3,4,5,6]))
```

(function, iterable)

J.S, Python Concepts

Date / /

Page No.

* zip

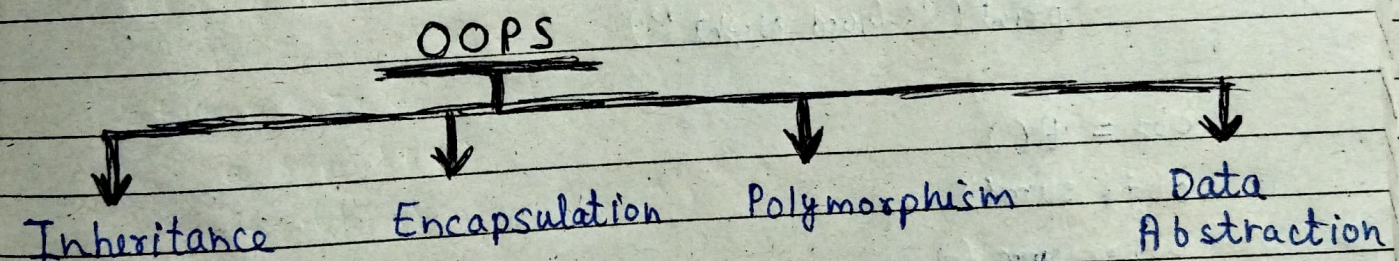
```
a = [1, 2, 3, 4]
b = ["sach", "Kiara", "Anu"]
print (list (zip (a,b)))
```

// Both list have same no. of element

* Unzip

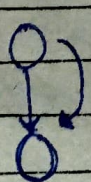
```
a = [(1, "sach"), (2, "Kiara"), (3, "Anu")]

z, x = list (zip (*a))
print (z, x)
```

▶ OOPS (Object oriented Programming structure)

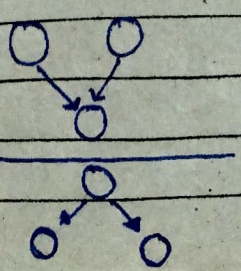


- Single
- Multiple
- Multi level
- ~~Hybrid~~
- ~~Hiarchial~~

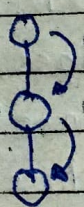
Single



Multiple



Multilevel



Date / /
Page No.

(कुछ भी लिख सकते हैं)

```
class anshul:
    def sum(self):          // self (instance)
        print("hi block-chain")

s = anshul()
s.sum()
```

* Single Inheritance

```
class A:
    def sum():
        print("Good Morning")

class B(A):
    def sum1():
        print("Good Night")

obj = B()
obj.sum()
obj.sum1()
```

* Multiple Inheritance

```
class A:
class B:
class C(A,B):
```

* Multilevel Inheritance

```
class A:
class B(A):
class C(B):
```

Date / /
Page No.

Encapsulation

- Public // Access anywhere
- Private // Access only with in its own class
- Protected // Access by Inheritance.

* How to declare private
  prefix -- (underscore)

```
__a = 5
```

* How to declare Protected
  prefix _ (underscore)

```
_a = 5
```

▶ Constructor // call automatically with the help of object creation.
(in python)

* How to define constructor

```
def __init__()
```

```
class Sach:
    def __init__(self, a, b):
        print("I am constructor", a, b)

obj = Sach(2,3)
```

* constructor used only in Class.

* Method used inside class as well as outside class.

- Non-local variable
  - Defined Inside Nested function. Scope only Inside Nested function.
- local variable
  - Defined Inside function & scope only used in function
- Global variable
  - Declare anywhere in program & scope used anywhere in program.