Python

Discover – Govido Vonn Rossum, 1989
First Version – 1991
python 1.0
=
=
Python 3.9.1

work on setl language
+
Drawbacks → It make notes

www.python.org

IDLE → Integrated development learning environment

IDE => Integrated development environment (pycharm)

to find out type – type()

<class int>
<class float>

% → modulus

remainder find out

10%2 → 0

// →

10//2 → 5

/ →

5/10 → 0.5

List in Python → collection of items

l = [1, 2, 3, 4, 5]
0 1 2 3 4

l[0] → first element

float → 5.73

int → 5, 13

Keywords → Python library

True, False → Bullean

Bool

True = 1
False = 0

Rules

1) variable → Type ✗

int a=5 ✗

a=5 ✓

2) net_amount , _abc

3) * %, ✗] इनसे शुरू नहीं होना चाहिए

4) numbers ✗

literals

Raw data given in a variable

a=5

- numeric literal
- string literals
- Boolean literals
- None

String :→ collection of characters

eg → Sumit , 'Sumit'

a → caracter

Sumit → String

Boleam literal

a = True / false

None literal

a = None
print(a) → None

list = [1, 2, 3, 4]

tl (touple) = (1, 2, 3, 4)

History (Monte Python Flying circus)

Constants :-

Pi = 3.14

Identifiers / Keywords / Reserved word

Variable

words
Predefined

Integers
X = 20 ✓
true = 30 X

Magic variables

__abc__

private variables

__abc

Collection of elements

List	Tuple
[ ]	( )
add, delete, sort	add, delete ✗
9-10 methods	count, Index
mutable	Immutables

simple if-else statement

one way ~~statement~~ decision

syntax

```
if <Conditon> :
    statement
```

WAP to print "Hello" message

a number is greater than 5

```
a= int(input("Enter a"))
if a>5
    print("Hello")
```

blank → Indent

WAP to print "Hello" message
a number is greater than 5. otherwise
print No

```
if a>5:          — True
    print("Hello")
else:            → false
    print("No")
```

WAP to find that the no is even or odd?
2) WAP to check that the number is +ve or -ve
3) ~~WAP to check~~ that a
3) WAP to find the Greatest Number b/w two numbers
4) WAP to check that a number is divisible by 5 or not

Ans 3

```
a = 5
b = 6
if a > b:
    print("a is greater")
else:
    print("b is greater")
```

== → comperision
= → assign

even or odd

even no - 2 divisible & remender = 0

```
n = int(input("Enter"))

if n%2 = 0:
        print("even no")

else:
        print("odd no")
```

Nasted if

a, b, c

```
if a>b:
        if a>c:
                print("a is greater")
        else:
                print("c is greater")
else:
        ~~print("b is greater~~
        if b>c:
                print("b is greater")
        else:
                print("c is greater")
```

a, b, c
a = a>b, a>c
b = b>c, b>a
c = c>a, c>b

Divisible by 5 or not

```
n = int(input("Enter no
if n % 5 == 0:
        print("no is divisible by 5")

else:
        print("no is not divisible by 5")
```

if elif statement

multiway decision

```
a, b, c   if a>b and a>c:
                print("a is greater")
a=a>b a>c
          elif b>c:
                print("b is greater")
          else:
                print("c is greater")
```

```
a = int(input("Enter a"))
b = int(input("Enter b"))
c = int(input("Enter c"))

if a>b and a>c:
        print(a, "is greater")
elif b>c:
        print(b, "is greater")
else:
        print(c, "is greater")
```

1. WAP to check that the given year is leap or not
2. WAP to check that the number is +ve, -ve or zero
3. WAP to check that the number is divisible by both 5 and 11 or not

Loops / Iterative statements

while loop
for loop

while loop :-

1) initialization / initial value
2) condition
3) iteration (increment / decrement)

WAP to print "Hello" message 5 times using while loop

```
i = 1          → initilization
while i <= 5 :  → condition
        print ("Hellos")
        i = i + 1  → interation
```

i=1	1<=5 ✓	Hello
i=2	2<=5 ✓	Hello
i=3	3<=5 ✓	Hello
i=4	4<=5 ✓	Hello
i=5	5<=5 ✓	Hello
i=6	6<=5 ✗	

WAP to print counting from 1 to 5

```
i=1
while i<=5:
    print (i)
    i=i+1
```

i=1 1<=5 ✓ 1
i=2 2<=5 ✓ 2
i=3 3<=5 ✓ 3
i=4 4<=5 ✓ 4
i=5 5<=5 ✓ 5
i=6 6<=x

WAP to find the sum of number b/w 1 to 5
1+2+3+4+5 = 15

```
i=1
sum=0
while i<=5:
    sum = sum+i
    i=i+1
print ("sum = " sum)
```

i = 1, 1<=5
sum = 0+1 = 1
i=2, 2<=5
sum = 1+2 = 3

i=3, 3<=5
sum = 3+3
i=4
sum = 6+4 = 10
i=10
sum = 10+5 = 15

1) WAP to print the counting from 10 to 1
2) WAP to print all even number b/w 1 to 10
3) WAP to print the sum of all even number b/w 1 to 10
2+4+6+8+10 = 30
4) WAP to find the factorial of no

5! = 5x4x3x2x1 = 120

all even no b/w 1 to 10

```
i = 2
while i <= 10 :
        print (i)
            i = i+2
```

factorial of No

```
n = int(input("Enter no")) -> 5
  i = n -> 5
while  i >= 1 :
            result = result * i ;
              i = i-1
print (result)
```

<u>For loop</u>

WAP to print 'Hello' message 5 times

```
for i in range (1, 6) :
                print ("Hello")
```

```
for i in range (1,5,1)
                print ("Hello")
```

```
for i in range(10,0,-1):
    print(i)
```

WAP to print even no b/w 50 to 100

WAP to print fibonacci series upto 10

i

0 + 1 = 1
1 + 1 = 2
2 + 1 = 3
3 + 2 = 5
5 + 3 = 8

factorial

```
n = int(input("Enter a no"))
result = 1
for i in rang(1, n+1):
    result = result * i
print("factorial =", result)
```

For loop	While Loop
How many loop will be executed (1 to 100) range	Condition we know X.

WAP to find sum of 1 to 5

```
result = 0
for i in range (1, 6, 1):
    result = result + i
print (result)
```

$\frac{27}{3} \times 100 = \frac{5}{100}$ 937.5

$\frac{1875}{2}$

fibonacci

```
a = 0
b = 1
for i in range (1,11,1):
    c = a+b
    print(c)
```

fibbonacci series upto 10

```
a = 0
b = 1
print(a)
print(b)
for i in range(1,9,1):
        c = a+b
        print(c)
        a = b
        b = c
```

even no in b/w (50 to 1)

```
for i in range (50,0,-2)
        print(i)
```

Most Imp

Reverse of a Number

123 → 321

I 123 % 10 = 3
123 / 10 = 12

II 12 % 10 = 2
12 / 10 = 1

III 1 % 10 = 1
1 / 10 = 0

$3 \times 100 + 2 \times 10 + 1$

n=123 result=0
whil(n>0):
rem = n%10
result = 0×10+rem (or result = result×10+rem)
n = n//10

rem=3
result=3
rem=2
result= 3×10+2
= 32
rem = 1
result = 32×10+1
= 321

```
n = int(input("Enter a number"))
result = 0
whil n>0:
    rem = n%10
    result = result×10 + rem
    n = n//10
print("Reverse = ", result)
```

1) WAP to find that the no is pallinderome or not pallinderome
Pallindrome No → 121 → 121
If Reverse of a Number is equal to original No

li = [1, 2, 3, 5, 4]

li.remove(4)
print(li)
[1, 2, 3, 5]

adding element in list

1) append()
2) extend()
3) insert()

remove element in list

1) pop()
2) remove()

ordering element in list

1) reverse()
2) sort()

li.reverse()
print(li)
[5, 3, 2, 1]

```
li.sort()
print(li)
[1, 2, 3, 5]
```

Q WAP to reverse a list in descending

Q traversing a list

```
li = [11, 23, 45, 67, 87]
print(li[0])
print(li[1])

print(len(li))
for i in li:
    print(li)

j = 0, len = len(li)
while j < len(li)
        print(li[j])
        j = j + 1

li.clear()
li
→ []
li.extend([1, 2, 33, 45])
li
→ [1, 2, 33, 45]
li.count(33)
→ 1
li.index(33)
→ 2
b = li.copy()  → Address is different
b
→ [1, 2, 33, 45]
```

→ address is same in this

```
a=li
a
→[1,2,33,45]
li.clear()
li
→[]
b
→[1,2,33,45]
a
→[]
```

Break
continue] → flow break

```
for i in range(1,11,1):
    if i==5:
        break
    print(i)
→ 1
  2
  3
  4
  5
```

```
for i in range(1,11,1):
    if i==5:
        continue
    print(i)
→ 1
  2
  3
  4
  6
  7
  8
```

→ 5 is not here

```
for i in range(1,11,1):
    if i==5:
        pass
    print(i)
```

→ 1
2
3
4
5
6
7
8
9
10

```
for i in range(1,11,1):
    if i%2 == 0
        pass
        print(i)
```

```
tup = (1,2,3,4,5)
tup
→ (1,2,3,4,5)

type(tup)
→ tuple

tup.index(5)
→ 4
tup.count(5)
→ 1
```

tuple is immutable → not change
list is mutable → changable

tuple(li)
→ ()
li
→ []

tp= tuple(b)
tp
→ (1, 2, 3, 4, 5)
print(tup*2)
(1, 2, 3, 4, 5, 1, 2, 3, 4, 5)

directories

d = {'car': 'swift', 'Phone': 'iphone'}
d
→ {'car': 'swift', 'phone': 'iphone'}

type(d)
→ dict
print(d['car'])
→ swift

print(d.get('car'))
swift

d['car'] = 'i20'
d
→ {'car': 'i20', 'phone': 'iphone'}

for i in d:
    print(i)
→ car
→ phone

print(d.value())
→ dict_value(['i20', 'iphone'])

~~for i in d:~~
~~print~~
d.clear

clear()
copy()
get()
value()
keys()
pop()
popitem()
del
sort()

d.keys()
→ dict_keys(['car', 'phone'])

d.pop('car')
→ 'i20'

d
→ {'phone': 'iphone'}

d['car'] = 'swift'
→ {'phone': 'iphone', 'car': 'swift'}

a = d.keys()
a
→ dict_keys(['phone', 'car'])

```
a1 = list(a)
a1.sort()
a1
['car', 'phone']
d[1] = 'harsha'
a = d.keys()
a
d.dict_keys(['phone', 'car', 1])

d
{'phone': 'iphone', 'car': 'swift', 1: 'harsha'}
print(list(d)[0])
→ phone

print(list(d)[2])
→ 1

print(list(d.value())[0])
→ iphone
```

Amstrong

$153 \rightarrow 1^3 + 5^3 + 3^3$

$1 + 125 + 27 \rightarrow 153$

$\begin{cases} 15 \% 10 = 3 \\ 153/10 = 15 \end{cases}$

$\begin{cases} 15 \% 10 = 5 \\ 5/10 = 1 \end{cases}$

$\begin{cases} 1 \% 10 = 1 \\ 1/10 = 0 \end{cases}$

```
n = 153
result = 0
while n>0:
    rem = n%10
    result = result + rem**3
    n = n//10
print(result)
if(result == temp):
    print("Amstrong no.")
else:
    print("Not Amstrong no")
```

Sets

```
s = {1, 2, 3, 45, 6}
s
→ {1, 2, 3, 6, 45}
type(s)
→ <class 'set'>

s.add(34)
s
→ {1, 2, 3, 34, 6, 45}
```

Time पर एक element होता है इसमें add में

```
s.clear
s
→ set()
s = {1, 2, 3, 4, 5}
s.pop()
→ 1
s[0]
→ Error

for i in s
    print(i)
→ 2
  3
  4
  5
```

s.remove(4)

s

→ {2, 3, 5}

s1 = {3, 4, 5, 6, 7, 8}

s1

→ {3, 4, 5, 6, 7, 8}

s.union(s1) → Merge two sets

→ {2, 3, 4, 5, 6, 7, 8}

s.intersection(s1)

→ (3, 5) → common element

~~copy~~

s.update(s1)

s

→ {2, 3, 4, 5, 6, 7, 8}

s.difference(s1)

→ {2} → not in s1

s1.difference(s)

→ set() → not in s

s.difference_update(s1)

s

→ {2}

WAP to convert decimal no into binary

Function

Block that can work independtly

- user defined function
- library / Pre-defined function

Library function

print()
input()
eval("12+12")
27

String functions

a = "Harsha" → collection of character
a.count(a) ⇒ 2 Ans

find (a) ⇒

rfind() → last prefrence

capitalize() → first letter bigger
title() → All first letter capatilize

upper() → All words capatil

~~swapcase() → reverse a word~~

islower() → if all letter is lower
it say true
else it say false

isupper → if all letter is capatilize
it say true
else it say false

swapcase() → it convert all first letter small & others capitilize

replace() → replace an a = "Hersha Mehta"

a.replace('mehta', 'Graba')

Harsha Graba

~~replace~~

strip() → a = " harsha "
print(a.strip())
→ harsha

it remove first & last spaces

lstrip() → it remove space from left

rstrip() → it remove space from right

partition () → it convert string into tupple

Join () → join all tupple & make a string

'@'.join(

tpl = ('a', 'b', 'xc')

~~print (tpl)~~

a = "#".join (tpl)

a

→ 'a#b#xc'

~~l = [1, 2, 3]~~

~~a = "*".join(l)~~

it only add strings not integer

l = ["1", "2", "3"]

a = "*".join(l)

a

→ '1*2*3'

a = " "

isspace → if string contain space it say true
else it say false

isalpha() → if string contain alphabet, number it say true
else it say false (space special character or)

isdigit() → if it contain integers say true
else it say false

isalnum() → if any string contain alpha numeric characters it say true
else it say false (space, *, /, #)

startwith() → if string starts with specified value it say true
else false

endwith() → if string ends with specified value it say true
else false

a = "I am a sumit"
print("am" in a)
→ True

user-defined function

```
def function reserve(parameter)
{
  body (statements)
  return statement
}
```

Math library

```
import math

math.pow(3, 3)
→ 27

math.ceil(4, 6)
→ 5

math.factorial(5)
→ 120

math.ceil(4.4)
→ 5
```

math.floor(3.4)
→ 3

math.floor(3.8)
→ 3

math.sqrt(4)
→ 2

math.trunc(6.7)
→ 6

math.trunc(6.9)
→ 6

math.round(6.9)
→ 7

random(0,1)
→

Alias Library

import math

math.power()
math.sqrt()

import math as m

m.power()
m.sqrt()

Date & Time Library

```
from datetime import date
datetime.now()
```

```
import datetime
```

User defined function

def

→ declaration
→ define
→ calling

Wtt to sum of two no's using user defined function

```
def add():
        a=int("input")
        b=int("input")
        c=a+b
        print(c)

add();
```

```
def factorial():
        n=5
        for i in range(1,6,1):
                result=result*i
        print("factorial=", result)

factorial()
```

→ 120

Variable :-

a=5

→ Local Variables

→ Global Variables

```
def add():
    a = ---
    b = ---
    c = a+b
    print(c)
```

Local variable

```
a=
b=
def sum():
    c=a+b
    print(c)
```

Global variable

Types of function

→ Taking nothing, Return Nothing

→ Taking nothing, Return something

→ Take somthing, Return Nothing

→ Take somthing, Return Something

Take → [parameter]

↓

return Statement

```
a=5
b=6
c=a+b
print(c)
```

Take nothing Return Nothing

```
def add():
    c=a+b
    return(c)
result add()
print(result)
```

Take nothing return something

```
def add(int a, int b):
    c=a+b
    print(c)
add(5,4)
```

Take something return nothing

```
def add(int a, int b)
    c=a+b
    return(c)
result = add(5,4)
print(result)
```

take something return something

firstly we one use open() function to open a file

operators

	modes
read →	"r"
write →	"w"
append →	"a"
	"x" → create

```
f = open("D:\\demo.txt", "r")
print(f.read())
```

```
f = open("D:\\demo.txt", "w")
~~print(f.write())~~
f.write("Harsha mehta")
print("Done")
f.close()
```

```
f = open("D:\\harsha.txt", "a")
f.write("Hello")
~~print~~ f.close()
```

readline

```
f = open("D:\\harsha.txt", "r")
print(f.readline(2))
f.close()
```

file pointer

tell() → current position of file pointer

seek() → It set the position of curs

Command line Arguments

```
from sys import argv
print(type(argv))
print("no of arguments", len(argv))
print(argv)
```

→ arguments

then open cmd
type py bin.py 10 20 30

it give length → 4
it make last type it in string

for i in argv:
print (i)

→ bin.py
10
20
30

print (argv(1))
a = argv (1:)
result = 0
for i in argv
result = result + int(i)

print ("Sum", result)

print (argv[20])

tell

f = open ("D:\\dd.txt", "w")
f.write ("harsha")
~~f.close()~~
prit(f.tell())
f.close()

```
def sum(a,b):
    c = a+b
    print(c)

def sam(a,b)
    c = a-b
    print(c)

def mul(a,b):
    c = a*b
    print(c)
```

make function in harsha.py file

```
import harsha as h
h.sum(2,3)
h.mul(4,5)
```

then call functions in another file

Array

```
import array as a
list = a.array('i',[1,2,3,4,5])
type(list)
→ <class 'array.array'>

list[0]
→ 1
```

list[3]
→ 4

list[2:]
→ array ('i', [3, 4, 5])

for i in list
    print(i)

→ 1
2
3
4
5

list.append
count
extend
index
insert
pop
remove
reverse
buffer_info

list.buffer_info()
→ (47016680, 5)

list.append(34)
→ array ('i', [1, 2, 3, 4, 5, 34])

```
list.insert(3, 67)
→ array('i', [1, 2, 3, 67, 4, 5, 34])

list[1]=98
list
→ array('i', [1, 98, 3, 67, 4, 5, 34])
```