014-351

22 23 24 25 26 27 28 29 30 31
M T W T F S S M T W T F S

बुधवार / WEDNESDAY

Appointments

*** SYLLABUS ***

CORE PYTHON

1. INTRODUCTION
2. VARIABLES AND DATA TYPES
3. OPERATOR
4. CONDITIONAL STATEMENT
5. LOOPS
6. CONTROL STATEMENT
7. STRING MANIPULATION
8. LIST
9. TUPLE
10. DICTIONARIES
11. FUNCTIONS
12. MODULES
13. INPUT / OUTPUT
14. EXCEPTION HANDLING

जनवरी / JANUARY '15

						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											

FEBRUARY

						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											

M T W T F S S M T W T F S S M T W T F S S

015-350

15

गुरुवार / THURSDAY

Appointments Wk - 3

INTRODUCTION OF PYTHON

Python is a general purpose high level programming language.

It can be used for —

- Console app
- Desktop application
- Web app
- Mobile app
- Machine learning
- IoT application.

↳ And it was created by Guido Vann Possum, in 1980.

↳ It works on different plateforms (Windows, Mac, Linux etc)

↳ Python has a simple Syntax ~~to~~ similar to English language.

16
शुक्रवार / FRIDAY
016-349

Appointments

Wk - 3

VARIABLES

Variables are containers for storing data values.

$x = 5$

↳ variable

DATA TYPES

Variables can store data of different types.

1. Txt type → str
2. Numeric types → int, float, complex
3. Sequence types → list, tuple, range
4. Mapping type → dict
5. Set type → set, frozenset
6. Boolean type → bool
7. Binary type → byte, bytearray, memoryview

16	17	18	19	20	21	22	23	24	25	26	27	28																															

FEBRUARY: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

M T W T F S S M T W T F S S M T W T F S S

017-348

17
शनिवार / SATURDAY

Appointments Wk - 3

- str → x = "Hello world"
- int → x = 10
- float → x = 11.1
- complex → 1j
- list → { x = ["apple", "banana", "mango"]
- tuple → x = ("apple", "banana", "mango")
- range → range(6)
- dict → x = {"name" : "mansi", "age": 20 }
- set → x = {"apple", "mango", "banana"}
- bool → True / False
- frozenset → frozenset({"apple", "banana"})
- bytes → b"Hello"

14

15

018-347

18
रविवार / SUNDAY

16

17

19 सोमवार / MONDAY

019-346

DECEMBER / जनवरी / JANUARY '15

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											
M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S

Appointments

Wk - 4

8

OPERATORS

Operators are used to perform operations on variable and values.

Python divides the operators in the following group.

- Arthmetic operator
- Assignment operator
- Comparision operator.
- Logical operator
- Identity operator
- Membership operator
- Bitwise operator

ARTHIMETIC OPERATOR :—

+	add	x+y
-	sub	x-y
*	mul	x*y
/	div	x/y
%	module	x%y
**	Exponentiation	x**y
//	floor division	x//y

ASSIGNMENT OPERATOR :—

=	x = 5	x = 5
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 5	x = x * 5
/=	x /= 5	x = x / 5
%=	x %= 5	x = x % 5
//=	x //= 5	x = x // 5
**=	x **= 5	x = x ** 5
&=	x &= 5	x = x & 5
|=	x |= 5	x = x | 5
^=	x ^= 5	x = x ^ 5
>>=	x >>= 5	x = x >> 5
<<=	x <<= 5	x = x << 5

COMPARISON OPERATOR :—

==	equal	x == y
!=	not equal	x != y
>	greater than	x > y
<	Smaller than	x < y
>=	greater than or equal to	x >= y
<=	less than or equal to	x <= y

8 LOGICAL OPERATOR

9 and return true if both r true x<5 and x<10
or R. true if 1 of stat. is true x<5 or x<
10 not reverse the result not (x<5 and x

11 —— CONDITIONAL STATEMENT ——

12 ↳ if
↳ if else
↳ if elif else
↳ Nested if

14 if statement → if statement

15 if expression
16 statement } if else
else
17 statement

18 else statement →

19 else

20 statement

elif →

```
if (x<y):
    statement 1,
elif (x==y):
    statement 2
else:
    statement 3
```

Loops

Python has 2 primitive commands

- while loop
- for loop

While loop →

```
i = 1
while i<6:
    print(i)
    i+=1
```

While loop is used to execute a block of statements repeatedly until a given a condition is satisfied.

शुक्रवार / FRIDAY

M T W T F S S M T W T F S S M T W T

Appointments Wk - 4

for loop →

```
fruits = ["apple", "mango", "banana"]
for x in fruits:
    print(x)
```

range() function —

Return sequence of no., it starts 0 by default.

Example —

```
for x in range(6):
    print(x)
```

Appointments

Wk - 4

SATURDAY

CONTROL STATEMENT

Control Statements are used to control the order of execution of the program based on the value and logic.

There are 3 types of control Statements

- Continue
- Break
- Pass

#1 continue Statement :

```
for char in 'Python'
    if (char == 'y'):
        continue
    print ('letter', char)
```

o/p

```
p
t
h
o
n
```

025-340

25

रविवार / SUNDAY

#2 Break Statement :

```
for char in 'python'
    if (char == 'h'):
        break
    print (char)
```

o/p. p
y
t

#3 Pass Statement :

Pass statement is a null operation

```
for char in 'python':
    if (char == 'h'):
        pass
    print (char)
```

o/p p
y
t
h
o
n

Wk - 5

STRING MANIPULATION

A string is a list of characters in order.

like - "Hello world"

To manipulate strings, we can use some of pythons built-in methods.

↳ Creation : word = "Hello World"
print(word)

o/p → Hello world

↳ Accessing : Use [] to access characters in a string

word = "Hello world!"
letter = word[0]
print letter

o/p H

बुधवार / WEDNESDAY

Appointments Wk - 5

↳ Length :
word = "Hello world"
len(word)

o/p 11

↳ finding : word = "Hello world"
print word.find("H")

o/p → 0

↳ count : S = "count, the no. of spaces"
print S.count(' ')

o/p → 8

↳ Split : word = "Hello word"
word.Split(' ')

o/p ['Hello', 'world']

↳ Repeat : print "." *10
print "," *10
.....

o/p .....

↳ Replacing : word = "Hello world"

word.replace ("Hello", "Goodbye")

o/p Goodbye world

Appointments Wk - 5

LIST

List are used to store multiple items in a single variable.

List are created using square brackets

Example:

```
list = ["apple", "mango", "banana"]
print(list)
```

→ List items are ordered, changeable, and allow duplicate values.

→ List items are indexed the first item has index [0], the second item has [1] etc. and so on.

→ The word changeable means that we can change, add, and remove items in a list after it has created.

List length :- To determine how many items a list has, use the len() function

example - list = ["Ram", "Mohan", "shyam"]
print (len(list))

o/p - 3

A list contain different data type

eg - list = ["abc", 34, True, !male"]

Tuples

Tuples are used to store multiple items in a single variable.

A tuples is collection which is ordered and unchangeable.

Tuples are written with round brackets.

Example →

```
tuple = ("Rani", "Mohan", "Shyam")
print (tuple)
```

→ Tuple items are ordered, unchangeable and allow duplicate values.

→ Tuple items are indexed, the first item has [0] and then so on.

Unchangeable means that cannot be change, add and remove items after created.

Tuple length : to determine how many items a tuple has, use the len() function

example :

```
tuple =("apple", "banana", "Mango")
print (tuple)
print (len (tuple))
```

tuple item can be of any data type :

example -

```
tuple = ("abc", 34, True, 40, "male")
```

Appointments Wk - 6

DICTIONARIES

Dictionaries are use to store data in key.

A dictionary is a collection which is unordered, changeable and does not allow duplicate.

Dictionaries are written with curly brackets.

Example:

```
dict = {"brand": "Ford",
        "model": "Mustage",
        "year": 1964
       }
print(dict)
```

Accessing Items →

```
dict = {"brand": "ford",
        "modal": "Mustage",
        "year": 1964}
x = dict["model"]
```

o/p → Mustage

M T W THURSDAY

Appointments Wk - 6

* length → print(len(dict))

the value in dictionary items can be of any data type.

FUNCTIONS

A function is a block of code which ~~ot~~ only runs when it is called.

You can pass data, known as 'parameter'

In python function is defined using 'def' keyword.

example → def my_function():
print("Hello function")

Appointments Wk - 6

Calling a function :- To call a function, use the function name followed by parenthesis

Example → def my_fun(): } defination
print("Hello")

my_fun() → calling

ARGUMENTS :- ——

Information can be passed into fun. as argument

ex → def my_fun(fname):
print(fname + "Refsnes")

my_fun("Email")
my_fun("Linus")

Passing a list as an Argument :

You can send any data type of argument to a function.

ex :

```
def myfun(food):
    for x in food:
        print(x)

fruits = ["apple", "banana", "mango"]
myfun(fruits)
```


Return Values :

```
def my_fun(x):
    return 5 * x

print(my_fun(3))
print(my_fun(2))
```

Appointments WK - 7

Modules

Consi

A modules is a file containing defination and statement. Python

Module are prewritten. python libraries which. helps you. to complete. a specific task.

A module. can define. functions, classes and variables. Grouping related code. into a module makes. the code. easier to understand. and use. It also makes the code logically organized.

Appointments Wk 7

Example — # simple module .{calc.py}
def add(x, y)
return (x+y)

import Statement/module

import calc
print (add (10, 2))

EXCEPTIONS HANDLING

Python has many built-in exceptions that are raised when your program encounters an error (something in the program goes wrong)

Exceptions can be handled using a try statement.

The critical operation which can raise an exception is placed inside the try. The code that handle the exception is written in the except.

ex:-

```
list = ['a', 0, 2]

for entry in list:
    try:
        print("The entry is", entry)
        r = 1/int(entry)
        break
    except:
        print("oops!")
        print("Next entry")
```