

Python

2017

Monday
(051-314) Week 8

20

New project

- Pycharm → create new project → location → change untitled to any desired name → CREATE
- Right click on the folder you created → New ↓
python file
- Save with extension '.py'

No header files

1) `print ("Janani")`

To run the program click RUN on the top

A little terminal output window with o/p Tuesday
(052-313) Week 8 21
opens.

Python executes line by line from top

`print ('*' * 10)`

output *****

In the above expression python evaluates and produce 10 asterisks

symbol within ' ' is considered a string

FEBRUARY 2017

S	M	T	W	T	F	S	S	M	T	W	F	S
1	2	3	4	5	6	7	8	9	10	11	12	13

February

22 Wednesday
(053-312) Week 8

Variables

price = 10

print('price')

Output is displayed as 10; here price is the variable

If
price = 10
price = 20
print('price')
output is 20, since
⇒ python executes the
program line by line

→ When keywords such as True or False is used the first letter is in capital, otherwise python will consider that as an error

Input

1. name = input('what is your name?')

23 Thursday
(054-311) Week 8

↓
tend to leave a space here

print('Hi ' + name)

Output What is your name? Janu
Hi Janu.

Program → Print Janu likes black

name = input('what is your name?')

color = input('what is your favourite color?')

print(name + ' likes ' + color)

Output Janu likes black.

Feb / Mar

For multiple lines triple quotes is used

27 Monday
(058-307) Week 9

```
course = "" multiple lines ""
```

```
print(course)
```

Positions

- ① course = "Python for beginners"
print(course[0])
Output - P (The index value is printed)
- ② If a negative index is given say [-1] it prints output as s (from the last)
- ③ course = "Python for beginners"
print(course[0:3])
Output = Prints PyE {Last character} {is excluded}
It leaves the last character in 0:3

28 Tuesday → If the last index value is not given
(059-306) Week 9 say [0:] ⇒ It prints all the letters

- If the input is given [:] it excludes the first character.
- If the input is given [:5] (without starting) index it assumes as zero.
- If the input is given as [:] it prints the whole word. Used for cloning

→ course = Janani's home

```
another = course[:]
```

```
print(another)
```

Jennifer
0 1 2 3 4 5 6 7 [: -1]

opennife

formatted strings

2017

first = 'janani'
 last = 'rengaraj'
 message = first + '[' + last + ']' is a reader!
 print(message) ↴ string
 ↓
 janani [rengaraj] is a reader!

Wednesday
(060-305) Week 9

01

Formatted string is prefixed with f

message = first + '[' + last + ']' is a reader'

msg = f'{first} {last} is a reader'

Here {} → whatever is written in it, the assigned term is displayed

String methods

1) course = 'Python'

print(len(course))

Thursday
(061-304) Week 9

02

output - 6 len ⇒ to know the length of the string

To access string functions dot operator is used

print(course.upper())

↳ string method

home = janani rengaraj

print(home.find('rengaraj'))

O/P ⇒ PYTHON

O/P = 7

print(course.lower())

index value

O/P ⇒ python

where r begins

print(course.find('p'))

O/P ⇒ 0 → index value is shown

print(course.find('s'))

O/P ⇒ -1 since s is not present

S	M	T	W	F	S	S	M	T	W	F	S
1	2	3	4	5	6	7	8	9	10	11	
12	13	14	15	16	17	18	19	20	21	22	23
26	27	28	29	30	31						

March

03 Friday (062-303) Week 9 for replacing string

home = janani janu

print ('janu')

print (home.replace('janu', 'jan'))

This is case-sensitive

for finding a string

course = Python for beginners

print ('Python' in course)

Output

→ operator

True (Boolean expression)

→ len
→ upper()
→ lower()
→ title()
→ find()
→ replace()

→ in operator

Arithmetic operations

① print (10 + 3) → +, -, *

② In case of division

print (10 / 3) = 3.333 (with floating point)

04 Saturday print (10 // 3) = 3 (only integer)

(063-302) Week 9 print (10 % 3) = 1 (remainder)

③ modulus →

④ exponent print (10 ** 3) = 1000

[i.e. 10^3]

Augmented operators

x = 10

x = x + 3

o/p = 13

print (x)

5 Sunday



x = 10

x += 3

print (x)

x = 13

x = 10

x = x + 3

x += 3

print (x)

o/p = 16

x -= 3

print (x)

x = 7

Operator precedence

2017

Monday
(065-300) Week 10 06

$$x = 10 + 3 * 2$$

print(x) O/P = 16

(BODMAS rule)

Order of operations

→ exponentiation

→ * /

→ + -

$$x = (2+3)*10 - 3$$

O/P: 47

Math functions

$$x = 2.9$$

print(round(x))

O/P = 3

$$x = 2.9$$

print(abs(-2.9)) abs = absolute function

O/P = 2.9

Tuesday
(066-299) Week 10 07

To import math functions use

import math

To use the functions a dot operator is placed

after math.ceil

math.floor

$$x = 2.9$$

print(math.ceil(x))

O/P = 3

$$x = 2.9$$

print(math.floor(x))

O/P = 2

S	M	T	W	T	F	S	S	M	T	W	T	F	S
							1	2	3	4	5	6	7
12	13	14	15	16	17	18	19	20	21	22	23	24	25
26	27	28	29	30	31								

March

08

Wednesday
(067-298) Week 10

If statements

is_hot = True

if is_hot:

print("It's a hot day")

Shift+Tab

print("enjoy your day")

O/P

It's a hot day

enjoy your day

To come
out of if
statement
press
shift+Tab

if incase is_hot = False

for the above program the output is
enjoy your day

eg

is_hot = True

is_cold = false

09

Thursday
(068-297) Week 10

if is_hot:

print("It's a hot day")

print("Drink plenty of water")

elif is_cold:

print("It's a cold day")

print("Wear warm clothes")

else:

print("enjoy your day")

O/P

It's a hot day

Drink plenty of water

2017

Friday
(069-296) Week 10 10

Eq
price = 1000000

has_good_credit = True

if has_good_credit:

down payment 0.1 * price

else:

down payment 0.2 * price

print ("Down payment : " + down_payment)

O/P Down payment : \$1000000

Logical operators

has_high_income = False

has_good_credit = True

if has_high_income **[and]** has_good_credit

Saturday

11

(070-295) Week 10

→ and - both

→ or - atleast one

→ not - inverses → has_good_credit = True

has_criminal_record = False

print if has_good_credit and not has
criminal record

print ("Eligible")

Sunday 12

O/P Eligible

MARCH 2017

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

March

13 Monday
(072-293) Week 11

comparison operators

>	greater	\geq greater than or equal to
<	lesser	\leq lesser than or equal to
\neq	not equal	$=$ equal

temperature = 30

if temperature > 30:

 print ("hot")

else:

 print ("not hot")

extra

weight converter

weight = input ("what is your weight?")

unit = input ("(L)bs or (K)gs")

if unit == (L) bs

 print (weight *)

 if unit.upper () == "L":

14 Tuesday
(073-292) Week 11

conv = int (weight) * 0.45

else:

 conv = int (weight) / 0.45

 print ("you are converted kgs")

while loops

i = 1

while i <= 5:

 print (i)

 i = i + 1

print ("Done")

what if i give as
① print ("i")

o/p i

② print ("* * * i")

* i
* * i
* * * i
* * * * i
* * * * *

2017

For loop

for item in "python":

 print(item)

o/p
p
y
t
h
o
n

Wednesday
(074-291) Week 11

15

List of names/ numbers can be printed

For large list of numbers - range function

for item in range(10)

 print(item)

i = i+1 | i+=1

price = [10, 20, 30]

total = 0

for price in price:

 total = price + total

print(f"The total price is {total}:")

Nested loops

① for x in range(2)

 for y in range(3)

 print(x, y)

consider

x y
(0 0)
(0 1)
(0 2)
(1 0)
(1 1)
(1 2)

Thursday
(075-290) Week 11

16

② numbers = [5, 2, 5, 2, 2]

for x_count in numbers

 print('x' * x_count)

numbers = [5, 2, 5, 2, 2]

o/p

xx xx x
x x
x x x x
x x
x x

→ (on for x_count in numbers)

output =

for count in range(x_count)

 or ← output += 'x'

output = output + x
print(output)

MARCH 2017

S	M	T	W	T	F	S	S	M	T	W	T	F	S
							1	2	3	4	5	6	7
12	13	14	15	16	17	18	19	20	21	22	23	24	25
26	27	28	29	30	31								

March

17

Friday
(076-289) Week 11

Lists

```
names = ['j', 'g', 'h']
```

```
print(names[-2])
```

↳ index value -2 or 2 = 4 or 1 ..

To print the largest number in list

```
numbers = [2, 4, 6, 8, 10]
```

```
max = numbers[0]
```

```
for number in numbers:
```

```
    if number > max:
```

```
        max = number
```

```
print(max)
```

2D Lists

```
matrix = [[1, 2, 3],
```

```
[4, 5, 6],
```

```
[7, 8, 9]]
```

18 Saturday
(077-288) Week 11

```
print(matrix) or print(matrix[0][1])
```

The value of the index can be changed by

```
matrix[0][1] = 15
```

```
for row in matrix:
```

```
    for item in row:
```

```
        print(item)
```

9 Sunday List methods

① numbers = [6, 9, 8, 0]

number.append(50)

print numbers

O/P 6, 9, 8, 0, 50

②

Insertion

numbers = [6, 9, 8, 0]

numbers.insert(0, 20)

print numbers

O/P 20, 6, 9, 8, 0

2017

③ removal

numbers = [6, 9, 8, 0]
numbers.remove(9)
print(numbers)
o/p 6, 8, 0

④ To clear list

numbers = [6, 9, 8, 0]

Monday
(079-286) Week 12

20

numbers.clear()

print(numbers)

o/p List is empty

⑤ To remove last item

numbers = [6, 9, 8, 0]
numbers.pop()
print(numbers)
o/p 6, 9, 8

⑥ Index method

numbers = [6, 9, 8, 0]
print(numbers.index(6))
o/p 0 (index value of 5)
(or)
print('6 in numbers')

⑦ To count the numbers

numbers = [6, 9, 8, 0]
print(numbers.count(5))
o/p 0

o/p True

⑧ To sort

numbers = [6, 9, 8, 0] Tuesday
(080-285) Week 12

21

numbers.sort()
print(numbers)

⑨ To copy

numbers = [6, 9, 8, 0] numbers.reverse() o/p 0, 6, 8, 9
numbers2 = numbers.copy() (for descending order)
print(numbers2)

To remove duplicate

numbers = [6, 9, 8, 0, 6]

unique = []

for number in numbers:

if number not in unique list

unique.append(number)

print(unique)

o/p 6, 9, 8, 0

MARCH 2017

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

March

22 Wednesday
(081-284) Week 12

Tuples - similar to lists, but cannot be modified

→ parenthesis () is used.

numbers = (1, 2, 3)

print (numbers[0])

two functions

count, index

Unpacking

coordinates = (1, 2, 3)

x = coordinates[0]

y = coordinates[1]

z = coordinates[2]

O/p = (2)

→ x, y, z = coordinates

print(y)

This can be used for both lists and tuples

Dictionaries - {} defines dictionaries

customer = {

23 Thursday "name": "janu",

"age": 19,

"is_verified": True,

print(customer.get("age"))

if not present

print(customer.get("birthdate", "Jan 1"))

To change customer["name"] = Janu

Program phone = input("phone: ")

digits = {

"1": "one",

"2": "two",

3

output = "" for ch in phone:

output += digits.get(ch, "!")

print(output)

2017

Friday
(083-282) Week 12

24

emoji converter
message = input('')
me eg words = message.split(' ')
| print (words)
o/p Good morning
→ ['good' 'morning']

use of split
function

Functions

To define a function \Rightarrow def. function_name():

```
def greet_user():
    print ('Hi There')
    print ('Warm welcome')
    print ("start")
greet_user()
print ("Finish")
```

Hi There
Warm welcome

Finish

Saturday
(084-281) Week 12 25

Parameters

```
← def greet_user(name)
    print ("Hi " + name + "!")
    print ("warm welcome")
    print ("start")
greet_user("Janu")
print ("finish")
```

→ parameter

o/p start
Hi Janu
Warm welcome
Finish Sunday 26
many arguments can be passed.

def greet_user (name, age):

MARCH 2017

S	M	T	W	T	F	S	S	M	T	W	F	S
1	2	3	4	5	6	7	8	9	10	11	12	13

March

27

Monday
(086-279) Week 13

keyword arguments

greet_user (last_name="rengaraj", first_name="janan")

If you are using both positional and keyword arguments, positional argument should precede the keyword arguments.

Return statement

```
def square(number):
    return number * number
result = square(3)
print(result)
```

creating a reusable function

28 Tuesday
(087-278) Week 13 — any function can be defined before and reused

Exceptions may:

```
age = int(input("age:"))
```

```
print(age)
```

```
except ValueError:
```

```
    print("Invalid")
```

{Value error}

Value error: 0 (correct)

| (error is present) / crashed

except zero division error:

```
Print("It cannot be zero")
```

2017

Wednesday
(088-277) Week 13

29

Comments

→ # function

It is used for explanations

classes

To define new types

Class - keyword

The first letter is capitalized, not separated by underscore for eg. Class EmailClient

class Point:

def move(self):

print("move")

def draw(self):

print("draw")

Thursday
(089-276) Week 13

30

point1 = Point()

point1.draw()

point1.x = 10

print(point1.x)

x is the attribute of point1

Constructors

class Point: (function) → parameter

def __init__(self, x, y):

self.x = x → initialization instead of
self.y = y point.x = 10

point1 = Point(10, 20)

print(point1.x)

S	M	T	W	F	S	S	M	T	W	F	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	

Mar / Apr

31

Friday
(090-275) Week 13

```
class Person:  
    def __init__(self, name):  
        self.name = name  
    def talk(self):  
        print("talk")  
  
person1 = Person("Janu")  
print(person1)  
person1.talk()
```

O/P Janu
talk

print(f"Hi I am {self.name}")

O/P Hi I am Janu

ST

Inheritance - To avoid repetition

class Dog:

def walk(self):

print("walk")

class Cat:

def walk(self):

print("walk")

defined the function once again



class Mammal

def walk(self):

print("walk")

02 Sunday

class Dog(Mammal):

→ Dog class will inherit the functions of Mammal class

← pass

class Cat(Mammal):

pass

dog1 = Dog()

dog1.walk()

Any functions can be defined here

e.g. def bark(self):

print("bark")

Satisfy Python because empty classes are not appreciated

2017

Modules

Monday

03

(093-272) Week 14

contains related functions and classes

import converters

converters. _____ (10)
(or)

from converters import kg_to_lbs
find

Packages

more than two modules present in it

for eg. Create a python package 'ecommerce' add two new files into it, shipping --init-- file

Now

shipping

```
def calc_shipping():  
    print("shipping")
```

In new prog

```
import ecommerce.shipping
```

```
ecommerce.shipping.calc_shipping()
```

(or)

```
from ecommerce.shipping import calc_shipping  
calc_shipping()
```

(or)

```
from ecommerce import shipping
```

```
shipping.calc_shipping()
```

Tuesday

04

(094-271) Week 14

S	M	T	W	F	S	S	S	M	T	W	F	S
					1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16	17	18	19	20	21
23	24	25	26	27	28	29	30					

April

15 Wednesday
(095-270) Week 14

Generating random values

random is a inbuilt package in Python

```
import random
```

```
for i in range(3)
```

```
    print(random.random())
```

O/P Prints three random values

What if I want to print random values b/w 10, 20

```
import random
```

```
for i in range(3)
```

```
    print(random.randint(10, 20))
```

Prog

```
import random
```

```
members = ['J', 'P', 'K', 'D']
```

```
print(random.choice(c))
```

O/P

5 Thursday

J

(096-269) Week 14

Dice prog

```
import random
```

```
class Dice
```

```
    def roll(self):
```

```
        first = random.randint(1, 6)
```

```
        second = random.randint(1, 6)
```

```
        return (first, second) → even if parenthesis are
```

dice = Dice() not given it is assumed,
in Tuples

```
print(dice.roll())
```

O/P

(4, 5)

2017

Friday 07
(097-268) Week 14

Files and directories

```
from pathlib import Path
path = Path("emails")
print(path.mkdir())
```

To search directory file

```
from pathlib import Path
path = Path()
```

```
print(path.glob(''))
```

↳ *: all

↳ everything

a new directory is created

rmdir - to remove directory

exist - to find whether file exists or not

Extra

len() ⇒ gives the length

numbers = [1, 2, 3]

print(len(numbers))

O/P 3

① Take a sequence of numbers and determine whether all are unique

def test(data)

if len(data) == len(set(data))

return True

else

return False

Set

=

Sunday 09

Set keeps unique elements

It returns the number of unique digits

print(test([1, 2, 3]))

(Ans)

if you need to get the input from user

data = input("Enter the numbers: ")

print(test(data))

S	M	T	W	F	S	S	M	T	W	F	S
					1	2	3	4	5	6	7
9	10	11	12	13	14	15	16	17	18	19	20

21 22

23 24 25 26 27 28 29 30

pril

0 Monday
(100-265) Week 15

Print aeiou randomly such as ieoua
import random
members = ['a', 'e', 'i', 'o', 'u']
random.shuffle(members)
print (''.join(members))

use of join function

list = [1, 2, 3]

s = '*'

print (s.join(list))

O/P

1 * 2 * 3

(for an iterable function join is used)

typecast

x, y = input().split()

1 Tuesday
(101-264) Week 15

x = int(x)

y = int(y)

print(x*y)

To remove a number from every 3rd position

def remove(int_list):

position = 2

idx = 0

len_list = len(int_list)

while idx > 0:

idx = (position + idx) // len_list

print(int_list.pop(idx))

len_list = len_list - 1

num = [1, 2, 3, 4]

remove(num)

16M 2017

Friday

(055-310) Week 8

24

Type conversion

age = 2020 - birth_year

string and integer cannot be subtracted

usage of int(), float(), bool() is required

```
birth_year = input('Birth year: ')
```

```
print(type(birth_year)) → To show what type it
```

age = 2020 - int(birth_year) ~~this step is not necessary in prog~~

```
print(type(age))
```

```
print(age)
```

User's weight (in pounds) → convert to kg → print.

```
pounds = input('What is your weight? ')
```

kilo = ^{float}int(pounds) * 0.45

```
print(kilo)
```

Strings

using double quotes to define a string

For eg. 'Python's course for beginners'

This is considered as a string to avoid
this error use double quotes

"Python's course for beginners"

If i want to add double quotes for
beginners

"Python's course for "beginners" "x

'Python's course for "beginners" ' ✓

Saturday

(056-309) Week 8

25