Contents

[1 Session 1 1](#_Toc522875717)

[2 Session 2 2](#_Toc522875718)

[3 Session 3 4](#_Toc522875719)

[4 Session 4 Working with List, Dictionary ,Tuple and Library 5](#_Toc522875720)

[4.1 List 5](#_Toc522875721)

[4.2 Dictionary 7](#_Toc522875722)

[4.3 Tuple 8](#_Toc522875723)

[4.4 Library 9](#_Toc522875724)

[5 Session 5 11](#_Toc522875725)

[5.1 Working with File 11](#_Toc522875726)

# Session 1

Values are like 2,4,5,.. which are integers and strings which are in this format ‘ ‘

type() gives you the type of value which are in these three formats (maybe more): intg., string and float (2.2)

print ()

Variables like x

capital letters and small letters are different in phython

u can’t define a variable starting whith number

there are python reseved kewords that can’t be used as a name of variable

Statatement: every executable part in python like print(x) and x=1

operators: +, -, \*, /, \*\* (2\*\*3=8), // gives the quotient and % gives he reminder

expresssion: every combination of values, variables and operators

print(myname+ “ “+ myfamily)

print(jadi\*5)

input() always gives a string

you can change the type by this function int()

use # for comments

# Session 2

Boolean (“bool”) is another type of python values which are in form of True or False.

type(True)=”bool”

>, <, >=, <=, == (a = 3: put 3 into a, a ==3: check whether a is 3 or no)

for conditional statemnts: and, or, not

if [conditional statement]:

[4 spaces] command

[4spaces]command

…

else:

[4 spaces] command

[4spaces]command

…

or

if [conditional statement]:

[4 spaces] command

[4spaces]command

…

elif [conditional statement2]:

[4 spaces] command

[4spaces]command

…

defining function:

def [functionname](varaibles):

[4spaces] what function does

[4spaces] return folan

there are bunch of built-in functions in python

you can also call different libraries and use their functions also.

for example *sin* function is in *math* library, so first you *import* math library and then you call *math.sin* function.

**booleans another type of variable True False**

**while …:**

**[4spaces] commands**

**for variablename in range(,):**

**[4spaces] commands**

if ….:

[4spaces]break

[] makes lists for example: [hasan, ali, amir]

saving in the list ->

list = []

listLen = 10

for i in range(listLen):

list.append(input())

# Session 3

a = ‘salam’

a[0] = s

a[-1] = m

a[:2] = sa

a[2:-1] = la

b = ‘Mo Sa’ + a[2:-1] + ‘h’

b = ’Mo Salah’

len (a) = 5

for harf in ‘salam’:

print(harf)

‘ad’ in ‘jadi’ -> True

‘bia berim’ > ‘addade aval’ -> True (because b is bigger than a)

b is bigger than B

**methods** for **classes** *like* **functions** for **libraries**

dir (‘jadi’) -> gives all the **methods** available for ‘jadi’’s class (which is string)

then yiu can use method by typin ‘jadi’.mehodname()

s = ‘hello’

s.upper() = HELLO

s.lower() = hello

s.find(‘l’) = 2

‘ sdfsdffs ’.strip() = ‘sdfsdffs’

‘hello’.startswith(‘h’) -> True

‘Hello’.startswith(h) -> False

‘Hello’.lower.startswith(h) -> True

name = ’asghar’

print (‘hello %s chetori?’ % name) -> hello asghar chetori?’

print (‘hello %s chetori? midooni %s salet shode?’ % (name, sen) )

# Session 4 Working with List, Dictionary ,Tuple and Library

## List

l = [ 1, 4, 5, ‘jadi’]

l[0]

l[:4] 🡪 gives all up to the 4th element

l[-2]

we can put everythin in the lists even lists themselves! 🡪 l2 = [ l, 5, [1,2]]

Lists are MUTABLE

l[0] = 8

l = [8, 4, 5, ‘jadi’]

for i in range (0, len(l)) : for item in l

l1 + l2

l2\*3

type(l1) 🡪 <class ‘list’>

dir (l1) 🡪 **gives all methods available for l1**

**SOME METHODs for STRINGS: .append, .extend, .sort, .pop**

l = [1,2]

l.append(3)

l = [1,2,3]

l.extend([2,3]) 🡪 like + but redefine the list (no need to make new variable or define l again)

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

l.sort()

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

l.pop() 🡪 removes the last member of the list and redefine the list

l.pop(2) 🡪 removes the third (2+1) member of the list and redefine the list

del l[2]

l = [1,2,3,3]

l.remove(2)

max(l) min(l) sum(l)

s = ‘some new words’

a = s.split(‘o’) = [‘s’, ‘me new w’, ‘rds’]

‘-’.join(a) = ‘s-me new w-rds’

## Dictionary

new kind of variable

d [key] = value

they can transform things

f2t = dict() or f2t = {}

f2t[‘yek’] = ‘bir’

f2t [‘do’] = ‘ikki’

f2t 🡪 {‘do’: ‘ikki’, ‘yek’: ‘bir’}

so we can define a dictionary like this:

ghad = {‘jadi’: 180, ‘hasan’: 200, ‘zahra’: 167}

ghad[‘jadi’] 🡪 180

\*order is not important for dictionary

list(ghad.keys()) = [‘jadi’,’hasan’,’zahra’]

list(ghad.values()) = [180, 200, 167]

\*in in dictioinary works much faster than list

‘jadi’ in ghad 🡪True

180 in ghad 🡪 False

dictionaryname.get(keyname)

return keyname value and if it does not exist return nothing

dictionaryname.get(keyname,0)

return keyname value and return 0 if it does not exist

3 methods for dictionary we learnt:

-.vlues

-.keys

-.get

## Tuple

like lists but they are immutable

tuple with () list with [] dic with {}

type(4,) 🡪 tuple

t = (0, 1, 2, 3, 4)

type(t) 🡪 tuple

a = [1,2]

x, y =a

or

x, y = 1, 2

🡪

x = 1

y = 2

email = ’[jadijadi@gmail.com](mailto:jadijadi@gmail.com)’

email.split(‘@’) = [‘jadijadi’, ‘gmail.com’]

name, domain = email.split(‘@’)

name = ’jadijadi’

domain = ‘gmail.com’

how tuples work with dicts?

dictname.items() 🡪 gives atems of a dict (item = key + value) in a tuple

then we can have list (dictname.items())

for x, y in list (dictname.items()) :

print ( ‘this item consists of %s and %s’ %(x, y))

when to use tuple when to use dictionary?

tuples:

using functions and paramethers

phone = dict()

phone[‘jadi’] = ‘0912’

phone [‘jadi’, ‘home’] = ‘0912’, ‘021’ 🡪 tuple

## Library

import …

someone has written some programs and others found it useful, so he has saved hem so that others can use it whenever they want

there are LOTS OF libraries

most powerful part of python

There are two types of liibraries:

standard (they are on your python by default)

* core
* built-in ffunctions: print, len
* import: import random, import math

search python standard libraries and review them

non-standards

import random

random.randint(1,10)

from random import randint

randint(1,10)

how to import non-standandard libraries?

pip install libraryname

some important libraries:

datetime --- datetime.date.today()

**pandas** --- python data analysis library

Numpy ---

ipython

# Session 5

## Working with File

creating a handle – open, read, write, close

f = open (‘filename’) 🡨 f is handle

type(f) 🡪 file

f.read()

for line in f:

print ()

f.readlines() 🡪 gives a list

how to write in a file?

f = open(‘filename’, **‘w’**)

f.write()

f.close()

## CSV

comma separated values

# Project

hash is a function: x 🡪 y

all paswords are: 0000 – 9999

open file

separate two parts

for each row 🡪 name + password

for 0 to 9999, make each of them string, find its sha 256

and then check whether it is equal to string name

sha 256 python 🡪 library

encode

for 0-9999: make a dictionary of the number and its hash

values are 0 to 9999, keys are their hash