Nama : Muhammad Fahri Ramadhan
NPM : 1194055
Kelas : D4TI-3B

1 Cara memilih item secara acak dari list

Gunakan random.choice():
Contoh:

```
import random

foo = [’a’, ’b’, ’c’, ’d’, ’e’]
print(random.choice(foo))
```

2 Import files dari folder berbeda

```
# some_file.py
import sys
# insert at 1, 0 is the script path (or ’’ in REPL)
sys.path.insert(1, ’/path/to/application/app/folder’)

import file
```

3 Menulis data JSON ke file

```
import json
with open(’data.json’, ’w’) as f:
json.dump(data, f)
```

4 Menginstall package menggunakan pip berdasarkan requirements.txt

```
pip install -r /path/to/requirements.txt
```

5 UnicodeEncodeError: ’ascii’ codec can’t encode character u’\xa0’ in position 20: ordinal not in range(128)

Pada dasarnya, berhenti menggunakan str untuk mengonversi dari unicode ke teks/byte yang diencode.
Sebagai gantinya, gunakan .encode() dengan benar untuk mengencode string:

```
p.agent_info = u’ ’.join((agent_contact,
    agent_telno)).encode(’utf-8’).strip()
```

6 Memastikan apakah direktori ada di Python

```
>>> import os
>>> os.path.isdir(’new_folder’)
True
>>> os.path.exists(os.path.join(os.getcwd(), ’new_folder’,
    ’file.txt’))
False
```

7 Cara merujuk ke objek null pada Python

```
if foo is None:
```

8 Cara cek nilai NaN pada Python

```
>>> import math
>>> x = float(’nan’)
>>> math.isnan(x)
True
```

9 Menghapus duplikasi pada list

```
>>> t = [1, 2, 3, 1, 2, 5, 6, 7, 8]
>>> t
[1, 2, 3, 1, 2, 5, 6, 7, 8]
>>> list(set(t))
[1, 2, 3, 5, 6, 7, 8]
>>> s = [1, 2, 3]
>>> list(set(t) - set(s))
[8, 5, 6, 7]
```

10 Sort dictionary berdasarkan key

```
In [1]: import collections

In [2]: d = {2:3, 1:89, 4:5, 3:0}

In [3]: od = collections.OrderedDict(sorted(d.items()))

In [4]: od
Out[4]: OrderedDict([(1, 89), (2, 3), (3, 0), (4, 5)])
```

11 Melakukan Reverse pada list

```
array=[0,10,20,40]
for i in reversed(array):
    print(i)
```

12 Cara mendapatkan nilai ASCII dari suatu karakter

```
>>> ord('a')
97
>>> chr(97)
'a'
>>> chr(ord('a') + 3)
'd'
>>>
```

13 Melakukan pengecekan ketersediaan variabel

Untuk memeriksa keberadaan variabel lokal:

```
if 'myVar' in locals():
# myVar exists.
```

Untuk memeriksa keberadaan variabel global:

```
if 'myVar' in globals():
# myVar exists.
```

Untuk memeriksa apakah suatu objek memiliki atribut:

```
if hasattr(obj, 'attr_name'):
# obj.attr_name exists.
```

14 Ekstrak nama file dari path, apa pun format os/path-nya

```
import os
print(os.path.basename(your_path))
```

15 Mencetak exception dengan Python?

```
except Exception as e: print(e)
```

16 Cara mendapatkan waktu saat ini dengan Python

```
>>> import datetime
>>> datetime.datetime.now()
datetime.datetime(2009, 1, 6, 15, 8, 24, 78915)

>>> print(datetime.datetime.now())
2009-01-06 15:08:24.789150
```

17 Membuat daftar semua file direktori

```
from os import listdir
from os.path import isfile, join
onlyfiles = [f for f in listdir(mypath) if isfile(join(mypath, f))]
```

18 Sort dictionary berdasarkan nilai

```
>>> x = {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}
>>> {k: v for k, v in sorted(x.items(), key=lambda item: item[1])}
{0: 0, 2: 1, 1: 2, 4: 3, 3: 4}
```

atau

```
>>> dict(sorted(x.items(), key=lambda item: item[1]))
{0: 0, 2: 1, 1: 2, 4: 3, 3: 4}
```

19 Menambah key baru pada dictionary

```
d = {'key': 'value'}
print(d) # {'key': 'value'}

d['mynewkey'] = 'mynewvalue'

print(d) # {'key': 'value', 'mynewkey': 'mynewvalue'}
```

20 Menggabungkan dua buah list

```
listone = [1, 2, 3]
listtwo = [4, 5, 6]

joinedlist = listone + listtwo
```

21 Melakukan pengecekan list kosong

```
if not a:
print("List is empty")
```

22 Mendefinisikan array dua dimensi

```
# Creates a list containing 5 lists, each of 8 items, all set to 0
w, h = 8, 5
Matrix = [[0 for x in range(w)] for y in range(h)]
```

coba tambahkan nilai seperti berikut

```
Matrix[0][0] = 1
Matrix[6][0] = 3 # error! range...
Matrix[0][6] = 3 # valid
```

23 Convert date ke datetime pada python

```
from datetime import date
from datetime import datetime

dt = datetime.combine(date.today(), datetime.min.time())
```

24 Cara mendapatkan hostname system

```
import socket
print(socket.gethostname())
```

25 Membuat list kosong dengan besaran yang ditentukan

```
>>> l = [None] * 10
>>> l
[None, None, None, None, None, None, None, None, None, None]
```

26 Menambah key baru pada dictionary

```
d = {’key’: ’value’}
print(d) # {’key’: ’value’}

d[’mynewkey’] = ’mynewvalue’

print(d) # {’key’: ’value’,
   ’mynewkey’: ’mynewvalue’}
```

27 Cara memilihi item secara acak dari daftar

```
import random

foo = [’battery’, ’correct’, ’horse’, ’staple’]
secure_random = random.SystemRandom()
print(secure_random.choice(foo))
```

28 Mengimpor file dari folder berbeda

```
import os, sys

from os.path import dirname, join, abspath
sys.path.insert(0, abspath(join(dirname(__file__), ’..’)))

from root_folder import file_name
```

29 Membaca JSON yang memiliki banyak items

```
data JSON dari API jsonplaceholder dengan endpoint URL:
    https://jsonplaceholder.typicode.com/posts

Buatlah program baru dengan nama list_artikel.py, kemudian isi
    dengan kode berikut:

import json
from urllib import request

url = "https://jsonplaceholder.typicode.com/posts"

# lakukan http request
response = request.urlopen(url)

# parsing data json
data = json.loads(response.read())

# gunakan perulangan untuk menampilkan data
for i in range(len(data)):
print(f"{i}. {data[i][’title’]}")
```

30 Cara menampilkan Input dari Keyboard

```
Python sudah menyediakan fungsi input() dan raw_input() untuk
    mengambil inputan dari keyboard.

Cara pakainya:
```

```
nama_varabel = input("Sebuah Teks")

# Mengambil input
nama = raw_input("Siapa nama kamu: ")
umur = input("Berapa umur kamu: ")

# Menampilkan output
print "Hello",nama,"umur kamu adalah",umur,"tahun"
```

31 Cara menampilkan output

```
nama = "Muhammad Fahri Ramadhan"
print "Hello",nama
```

32 Menggunakan string Formatting

```
nama = raw_input("Inputkan nama: ")
umur = input("Inputkan umur: ")
tinggi = input("Inputkan tinggi badan: ")

print "Hello %s, saat ini usiamu %d tahun dan tinggi badanmu %f
    cm" % (nama, umur, tinggi)
```

33 Menggunakan Perulangan for

```
item = ['kopi','nasi','teh','jeruk']

for isi in item:
print(isi)
```

34 Menggunakan Perulangan while

```
jawab = 'ya'
hitung = 0

while(jawab == 'ya'):
hitung += 1
jawab = input("Ulang lagi tidak? ")

print(f"Total perulagan: {hitung}")
```

35 Cara pakai operator relasi sama dengan

```
lulus = raw_input("Apakah kamu lulus? [ya/tidak]: ")

if lulus == "tidak":
print("Kamu harus ikut ujian")
```

36 Penggunaan If/Else

```
umur = input("Berapa umur kamu: ")

if umur >= 18:
print("Kamu boleh membuat SIM")
else:
print("Kamu belum boleh membuat SIM")
```

37 Membuat Program Dengan list

```
# Membuat list kosong untuk menampung hobi
hobi = []
stop = False
i = 0

# Mengisi hobi
while(not stop):
hobi_baru = raw_input("Inputkan hobi yang ke-{}: ".format(i))
hobi.append(hobi_baru)

# Increment i
i += 1

tanya = raw_input("Mau isi lagi? (y/t): ")
if(tanya == "t"):
stop = True

# Cetak Semua Hobi
print "=" * 10
print "Kamu memiliki {} hobi".format(len(hobi))
for hb in hobi:
print "- {}".format(hb)
```

38 Pengambilan panjang Tuple

```
# Membuat Tuple
hari = ('Senin', 'Selasa', 'Rabu', 'Kamis', 'Jum\'at', 'Sabtu',
    'Minggu')
```

```
# Mengambil panjang tuple hari
print("Jumlah hari: %d" % len(hari))
```

39 Mengakses nilai item dari dictionary

```
# Membuat Dictionary
mahasiswa = {
   "nama": "Muhammad Fahri Ramadhan",
   "umur": 22,
   "hobi": ["coding", "membaca", "belajar"],
   "menikah": False,
   "sosmed": {
      "facebook": "fahri-r",
      "twitter": "@fahri_r"
   }
}

# Mengakses isi dictionary
print("Nama saya adalah %s" % mahasiswa["nama"])
print("Twitter: %s" % mahasiswa["sosmed"]["twitter"])
```

40 Membuat fungsi mengembalikan nilai

```
def luas_persegi(sisi):
luas = sisi * sisi
return luas

# pemanggilan fungsi
print "Luas persegi: %d" % luas_persegi(6)
```

41 Cara menggunakan lambda expression

```
greeting = lambda name: print(f"Hello, {name}")
```

42 Penggunaan dari *args dan **kwargs

```
# membuat fungsi dengan parameter *args
def kirim_sms(*nomer):
print nomer

# membuat fungsi dengan parameter **kwargs
def tulis_sms(**isi):
print isi

# Pemanggilan fungsi *args
```

```
kirim_sms(123, 888, 4444)

# pemanggilan fungsi **kwargs
tulis_sms(tujuan=123, pesan="apa kabar")
```

43 Parsing XML di Python

```
import xml.dom.minidom as minidom

def main():
# gunakan fungsi parse() untuk me-load xml ke memori
# dan melakukan parsing
doc = minidom.parse("mahasiswa.xml")

# Cetak isi doc dan tag pertamanya
print doc.nodeName
print doc.firstChild.tagName


if __name__ == "__main__":
main()
```

44 Cara baca dan Parse file CSV di Python

```
import csv

with open('contacts.csv') as csv_file:
csv_reader = csv.reader(csv_file, delimiter=",")
print(csv_reader)
for row in csv_reader:
print(row)
```

45 Menulis data dictionary ke CSV

```
import csv

with open('contacts.csv', mode='a') as csv_file:
# menentukan label
fieldnames = ['NO', 'NAMA', 'TELEPON']

# membuat objek writer
writer = csv.DictWriter(csv_file, fieldnames=fieldnames)

# menulis baris ke file CSV
writer.writeheader()
writer.writerow({’NO’: ’10’, ’NAMA’: ’Nanda’, ’TELEPON’:
    ’02109999’})
writer.writerow({’NO’: ’11’, ’NAMA’: ’Fahriza’, ’TELEPON’:
    ’02148488888’})

print("Writing Done!")
```

46 Membuat fungsi lihat data

```
def show_contact():
clear_screen()
contacts = []
with open(csv_filename) as csv_file:
csv_reader = csv.reader(csv_file, delimiter=",")
for row in csv_reader:
contacts.append(row)

if (len(contacts) > 0):
labels = contacts.pop(0)
print(f"{labels[0]} \t {labels[1]} \t\t {labels[2]}")
print("-"*34)
for data in contacts:
print(f’{data[0]} \t {data[1]} \t {data[2]}’)
else:
print("Tidak ada data!")
back_to_menu()
```

47 Menggunakan fungsi create

```
def create_contact():
clear_screen()
with open(csv_filename, mode=’a’) as csv_file:
fieldnames = [’NO’, ’NAMA’, ’TELEPON’]
writer = csv.DictWriter(csv_file, fieldnames=fieldnames)

no = input("No urut: ")
nama = input("Nama lengkap: ")
telepon = input("No. Telepon: ")

writer.writerow({’NO’: no, ’NAMA’: nama, ’TELEPON’: telepon})
print("Berhasil disimpan!")

back_to_menu()
```

48 Membuat fungsi search

```
def search_contact():
clear_screen()
contacts = []

with open(csv_filename, mode="r") as csv_file:
csv_reader = csv.DictReader(csv_file)
for row in csv_reader:
contacts.append(row)

no = input("Cari berdasrakan nomer urut> ")

data_found = []

# mencari contact
indeks = 0
for data in contacts:
if (data[’NO’] == no):
data_found = contacts[indeks]

indeks = indeks + 1

if len(data_found) > 0:
print("DATA DITEMUKAN: ")
print(f"Nama: {data_found[’NAMA’]}")
print(f"Telepon: {data_found[’TELEPON’]}")
else:
print("Tidak ada data ditemukan")
back_to_menu()
```

49 Membuat fungsi edit

```
def edit_contact():
clear_screen()
contacts = []

with open(csv_filename, mode="r") as csv_file:
csv_reader = csv.DictReader(csv_file)
for row in csv_reader:
contacts.append(row)

print("NO \t NAMA \t\t TELEPON")
print("-" * 32)

for data in contacts:
print(f"{data[’NO’]} \t {data[’NAMA’]} \t {data[’TELEPON’]}")

print("------------------------")
no = input("Pilih nomer kontak> ")
```

```
nama = input("nama baru: ")
telepon = input("nomer telepon baru: ")

# mencari contact dan mengubah datanya
# dengan data yang baru
indeks = 0
for data in contacts:
if (data[’NO’] == no):
contacts[indeks][’NAMA’] = nama
contacts[indeks][’TELEPON’] = telepon
indeks = indeks + 1

# Menulis data baru ke file CSV (tulis ulang)
with open(csv_filename, mode="w") as csv_file:
fieldnames = [’NO’, ’NAMA’, ’TELEPON’]
writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
writer.writeheader()
for new_data in contacts:
writer.writerow({’NO’: new_data[’NO’], ’NAMA’: new_data[’NAMA’],
    ’TELEPON’: new_data[’TELEPON’]})

back_to_menu()
```

50 Menggunakan fungsi delete

```
def delete_contact():
clear_screen()
contacts = []

with open(csv_filename, mode="r") as csv_file:
csv_reader = csv.DictReader(csv_file)
for row in csv_reader:
contacts.append(row)

print("NO \t NAMA \t\t TELEPON")
print("-" * 32)

for data in contacts:
print(f"{data[’NO’]} \t {data[’NAMA’]} \t {data[’TELEPON’]}")

print("-----------------------")
no = input("Hapus nomer> ")

# mencari contact dan mengubah datanya
# dengan data yang baru
indeks = 0
for data in contacts:
if (data[’NO’] == no):
```

```
contacts.remove(contacts[indeks])
indeks = indeks + 1

# Menulis data baru ke file CSV (tulis ulang)
with open(csv_filename, mode="w") as csv_file:
fieldnames = [’NO’, ’NAMA’, ’TELEPON’]
writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
writer.writeheader()
for new_data in contacts:
writer.writerow({’NO’: new_data[’NO’], ’NAMA’: new_data[’NAMA’],
    ’TELEPON’: new_data[’TELEPON’]})

print("Data sudah terhapus")
back_to_menu()
```

51 Menggunakan main loop

```
if __name__ == "__main__":
while True:
show_menu()
```

52 Menangani exception

```
# import modul sys untuk memperoleh jenis eksepsi
import sys

lists = [’a’, 0, 4]
for each in lists:
try:
print("Masukan:", each)
r = 1/int(each)
break
except:
print("Upps!", sys.exc_info()[0], " terjadi.")
print("Masukan berikutnya.")
print()
print("Kebalikan dari ", each, " =", r)
```

53 Cara menghitung tanggal dan waktu :

```
from datetime import datetime
now = datetime.now()
mm = str(now.month)
dd = str(now.day)
yyyy = str(now.year)
hour = str(now.hour)
```

```
mi = str(now.minute)
ss = str(now.second)
print (mm + "/" + dd + "/" + yyyy + " " + hour + ":" + mi + ":" +
    ss)
```

54 Cara menghitung 2 angka yang di inputkan :

```
number1 = input("Angka Pertama: ")
number2 = input("\nANgka Kedua: ")

sum = float(number1) + float(number2)

print("Penjumlahan dari {0} dan {1} adalah {2}" .format(number1,
    number2, sum))
```

55 Cara menampilkan nama :

```
# Python program showing
# a use of raw_input()
g = raw_input("Enter your name : ")
print g
```

56 Cara menampilkan kalkulator :

```
# Program make a simple calculator that can add, subtract,
    multiply and divide using functions
# This function adds two numbers
def add(x, y):
return x + y
# This function subtracts two numbers
def subtract(x, y):
return x - y
# This function multiplies two numbers
def multiply(x, y):
return x * y
# This function divides two numbers
def divide(x, y):
return x / y
print("Select operation.")
print("1.Add")
print("2.Subtract")
print("3.Multiply")
print("4.Divide")
# Take input from the user
choice = input("Enter choice(1/2/3/4):")
num1 = int(input("Enter first number: "))
```

```
num2 = int(input("Enter second number: "))
if choice == ’1’:
print(num1,"+",num2,"=", add(num1,num2))
elif choice == ’2’:
print(num1,"-",num2,"=", subtract(num1,num2))
elif choice == ’3’:
print(num1,"*",num2,"=", multiply(num1,num2))
elif choice == ’4’:
print(num1,"/",num2,"=", divide(num1,num2))
else:
print("Invalid input")
```

57 Cara untuk bilangan Fibonacci :

```
def Fibonacci(n):
   if n<0:
      print("Incorrect input")
# First Fibonacci number is 0
   elif n==1:
      return 0
# Second Fibonacci number is 1
   elif n==2:
      return 1
   else:
      return Fibonacci(n-1)+Fibonacci(n-2)
# Driver Program
   print(Fibonacci(10))
```

58 Cara untuk sorting array :

```
def insertionSort(arr):
# Traverse through 1 to len(arr)
   for i in range(1, len(arr)):
      key = arr[i]
# Move elements of arr[0..i-1], that are
# greater than key, to one position ahead
# of their current position
      j = i-1
      while j >=0 and key < arr[j] :
         arr[j+1] = arr[j]
         j -= 1
         arr[j+1] = key

# Driver code to test above
arr = [12, 11, 13, 8, 4]
insertionSort(arr)
print ("Sorted array is:")
```

```
for i in range(len(arr)):
   print ("%d" %arr[i])
```

59 Cara untuk sorting array 2:

```
test_string = "Hello World My Name Is : Joo"
# printing original string
print ("The original string is : " + test_string)
# using split()
# to count words in string
res = len(test_string.split())
# printing result
print ("The number of words in string are : " + str(res))
```

60 Cara Sederhana Menghitung Volume Balok

```
print ("PROGRAM PYTHON MENGHITUNG VOLUME BALOK")
p = float(input("Panjang = "))
l = float(input("Lebar = "))
t = float(input("Tinggi = "))

v = p*l*t

print ("Volume Balok = %0.2f" %v)
```

61 Cara Menentukan Bilangan Ganjil Genap

```
bil = int(input("Masukan Bilangan :"))

if bil % 2 == 0:
   print("%d Merupakan Bilangan Genap" % bil)
else:
   print("%d Merupakan Bilangan Ganjil" % bil)
```

62 Cara Menentukan Nilai Indeks Mahasiswa

```
print("PROGRAM PYTHON MENENTUKAN NILAI INDEKS MAHASISWA")
tugas = float(input("\nMasukkan nilai Tugas: "))
uts = float(input("Masukkan nilai UTS: "))
uas = float(input("Masukkan nilai UAS: "))

na = (0.15 * uas) + (0.35 * uts) + (0.50 * uas)
if na >= 80:
   indeks = ’A’
elif na >= 70:
   indeks = ’B’
elif na >= 55:
   indeks = ’C’
elif na >= 40:
   indeks = ’D’
else:
   indeks = ’E’

print("\nNilai Akhir = %0.2f" % na)
print("Nilai Indeks = %c" % indeks)
```

63 Cara Menentukan Bilangan Terkecil dan Terbesar

```
print ("PROGRAM PYTHON MENGHITUNG NILAI TERKECIL & TERBESAR SERTA
    NILAI RATA-RATA")
n = int(input("\nMasukan Jumlah Bilangan = "))
bil = []
tot = 0
for x in range(n):
   m=x+1
a = int(input("Bilangan ke %d = "%m))
bil.append(a)
tot += bil[x]
rata2 = tot / n

print("\nBilangan Terkecil : %d" %min(bil))
print("Bilangan Terbesar : %d" %max(bil))
print("Nilai Rata-rata : %0.2f" %rata2)
```

64 Cara menggunakan Fungsi Rekursif

```
def pangkat(x,y):
if y == 0:
   return 1
else:
   return x * pangkat(x,y-1)

x = int(input("Masukan Nilai X : "))
y = int(input("Masukan Nilai Y : "))

print("%d dipangkatkan %d = %d" % (x,y,pangkat(x,y)))
```

65 perulangan while seperti for + range

```
i = 1
```

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

66 Menjumlahkan bilangan

```
jumlah = float(bil1) + float(bil2)
```

67 perulangan while untuk list

```
    listKota = [’Jakarta’, ’Surabaya’, ’Depok’, ’Bekasi’, ’Solo’,
        ’Jogjakarta’, ’Semarang’, ’Makassar’]

    # bermain index
    i = 0
    while i < len(listKota):
        print(listKota[i])
        i += 1
```

68 perulangan while dengan inputan

```
    a = int(input(’Masukkan bilangan ganjil lebih dari 50: ’))

    while a % 2 != 1 or a <= 50:
        a = int(input(’Salah, masukkan lagi: ’))

print(’Benar’)
```

69 perulangan while dengan continue

```
    listKota = [’Jakarta’, ’Surabaya’, ’Depok’, ’Bekasi’, ’Solo’,
        ’Jogjakarta’, ’Semarang’, ’Makassar’]

    # skip jika i adalah bilangan genap
    # dan i lebih dari 0
    i = -1
    while i < len(listKota):
        i += 1
        if i % 2 == 0 and i > 0:
            print(’skip’)
            continue

    # tidak dieksekusi ketika continue dipanggil
    print(listKota[i])
```

70 perulangan while dengan break

```
listKota = [
’Jakarta’, ’Surabaya’, ’Depok’, ’Bekasi’, ’Solo’,
’Jogjakarta’, ’Semarang’, ’Makassar’
]

kotaYangDicari = input(’Masukkan nama kota yang dicari: ’)

i = 0
while i < len(listKota):
   if listKota[i].lower() == kotaYangDicari.lower():
      print(’Ketemu di index’, i)
      break

   print(’Bukan’, listKota[i])
   i += 1
```

71 Menaikkan Exceptions

```
# Raise an instance of the Exception class itself

raise Exception(’Ummm... something is wrong’)

# Raise an instance of the RuntimeError class

raise RuntimeError(’Ummm... something is wrong’)

# Raise a custom subclass of Exception that keeps the timestamp
    the exception was created

from datetime import datetime

class SuperError(Exception):

def __init__(self, message):

Exception.__init__(message)

self.when = datetime.now()

raise SuperError(’Ummm... something is wrong’)
```

72 Menelan Exception

```
import json

import yaml
```

```
def parse_file(filename):

   try:

      return json.load(open(filename))

      except json.JSONDecodeError

   return yaml.load(open(filename))
```

73 Klausul Finally

```
def fetch_some_data():

db = open_db_connection()

query(db)

close_db_Connection(db)
```

74 fungsi penjumlahan

```
def add(x, y):
return x + y
```

75 Logging

```
import logging

logger = logging.getLogger()

def f():

try:

flaky_func()

except Exception:

logger.exception()

raise
```

76 fungsi perkalian

```
def multiply(x, y):
    return x * y
```

77 fungsi pembagian

```
def divide(x, y):
    return x / y
```

78 Error Logger

```
def log_error(logger)

def decorated(f):

@functools.wraps(f)

def wrapped(*args, **kwargs):

    try:

        return f(*args, **kwargs)

    except Exception as e:

        if logger:

            logger.exception(e)

        raise

            return wrapped

    return decorated
```

79 Retrier

```
import time

import math
def retry(tries, delay=3, backoff=2):

    if backoff <= 1:

    raise ValueError("backoff must be greater than 1")
    tries = math.floor(tries)

    if tries < 0:

        raise ValueError("tries must be 0 or greater")
    if delay <= 0:

        raise ValueError("delay must be greater than 0")
def deco_retry(f):

def f_retry(*args, **kwargs):

    mtries, mdelay = tries, delay # make mutable
    rv = f(*args, **kwargs) # first attempt

    while mtries > 0:

        if rv is True: # Done on success

            return True
        mtries -= 1     # consume an attempt

        time.sleep(mdelay) # wait...

        mdelay *= backoff # make future wait longer
        rv = f(*args, **kwargs) # Try again
        return False # Ran out of tries :-(
    return f_retry # true decorator -> decorated function

    return deco_retry # @retry(arg[, ...]) -> true decorator
```

80 Cara untuk mencetak semua permutasi

```
from itertools import permutations

# Mendapatkan semua permutasi dari [1, 2, 3]
perm = permutations([1, 2, 3])

# Print semua permutasi
for i in perm:
    print(i)
```

81 Menggunakan pernyataan If else

```
myDict = {1: 1, 2: 4, 3: 9}
print("The dictionary is:", myDict)
key = 4
```

```
if key in myDict.keys():
   print(myDict[key])
else:
   print("{} not a key of dictionary".format(key))
```

82 Program Menentukan Nilai Akhir Semester

```
def fungsi_total_nilai(var_harian, var_uts, var_uas) :
   var_harian = int(var_harian) * 0.3
   var_uts = int(var_uts) * 0.3
   var_uas = int(var_uas) * 0.4

   var_total = var_harian + var_uts + var_uas
   return var_total
```

83 Mengubah string pada list ke int

```
results = list(map(int, results))
```

84 Unzipping files

```
import zipfile
with zipfile.ZipFile(path_to_zip_file, 'r') as zip_ref:
zip_ref.extractall(directory_to_extract_to)
```

85 Perbandingan string dengan case-sensitive

```
string1 = 'Hello'
string2 = 'hello'

if string1.casefold() == string2.casefold():
   print("The strings are the same (case insensitive)")
else:
   print("The strings are NOT the same (case insensitive)")
```

86 Mendefinisikan infinite number

```
import math
test = math.inf
```

87 Mencari rata-rata pada sebuah list

```
l = [15, 18, 2, 36, 12, 78, 5, 6, 9]

import statistics
statistics.mean(l)
```

88 Merename key pada dictionary

```
mydict[k_new] = mydict.pop(k_old)
```

89 Merename file

```
import os

os.rename('a.txt', 'b.kml')
```

90 Mendapatkan nama script dengan python

```
os.path.basename(__file__)
```

91 Convert JSON string ke dict

```
import json

d = json.loads(j)
print d['glossary']['title']
```

92 Mendapatkan nomor minggu

```
>>> import datetime
>>> datetime.date(2010, 6, 16).isocalendar()[1]
24
```

93 Date string ke date object

```
>>> import datetime
>>> datetime.datetime.strptime('24052010', "%d%m%Y").date()
datetime.date(2010, 5, 24)
```

94 Membuat MD5 Checksum dari file

```
import hashlib
def md5(fname):
   hash_md5 = hashlib.md5()
   with open(fname, "rb") as f:
      for chunk in iter(lambda: f.read(4096), b""):
         hash_md5.update(chunk)
   return hash_md5.hexdigest()
```

95 Disable log messages dari Requests library

```
import logging

logging.getLogger("requests").setLevel(logging.WARNING)
```

96 Convert dict ke kwargs

```
func(type='Event')
```

97 Rename banyak file dalam direktori

```
$ ls
cheese_cheese_type.bar cheese_cheese_type.foo
$ python
>>> import os
>>> for filename in os.listdir("."):
...  if filename.startswith("cheese_"):
...    os.rename(filename, filename[7:])
...
>>>
$ ls
cheese_type.bar cheese_type.foo
```

98 Convert local time string ke UTC

```
from datetime import datetime
import pytz

local = pytz.timezone("America/Los_Angeles")
naive = datetime.strptime("2001-2-3 10:11:12", "%Y-%m-%d %H:%M:%S")
local_dt = local.localize(naive, is_dst=None)
utc_dt = local_dt.astimezone(pytz.utc)
```

99 Mencari dan mereplace elemen pada list

```
>>> a=[1,2,3,1,3,2,1,1]
>>> [4 if x==1 else x for x in a]
[4, 2, 3, 4, 3, 2, 4, 4]
```

100 Mencari irisan pada list

```
>>> a = [1,2,3,4,5]
>>> b = [1,3,5,6]
>>> list(set(a) & set(b))
[1, 3, 5]
```