print('Hello, Python!')

# Check the Python Version

import sys

print(sys.version)

type(12); type(12.0)

int('1') # Convert a string into an integer

int('1 or 2 people') # Convert a string into an integer with error

type(True) # Type of True

type(False) # Type of False

int(True) # Convert True to int

bool(1) # Convert 1 to Boolean

bool(0) # Convert 0 to Boolean

float(True) # Convert True to float

print(bool("1")) #convert 1 to “true”

43 + 60 + 16 + 41

50 – 60

5 \* 5

25 / 6 # Integer division operation expression

25 // 6 # Integer division operation expression

x = 43 + 60 + 16 + 41 # Store value into variable

y = x / 60 # Use another variable to store the result of the operation between variable and value

z=x+y

x = x / 60 # Overwrite variable with new value

total\_min = 43 + 42 + 57 # Total length of albums in minutes

total\_hours = total\_min / 60 # Total length of albums in hours

STRING

name = "Michael Jackson" # Assign string to variable

print(name[6]) # Print the element on index 6 in the string

print(name[-1]) # Print the last element in the string

name[0:4] # Take the slice on variable name with only index 0 to index 3

name[8:12] # Take the slice on variable name with only index 8 to index 11

name[::2] # Get every second element. The elments on index 1, 3, 5 ...

name[0:5:2] # Get every second element in the range from index 0 to index 4

statement = name + "is the best" # Concatenate two strings

3 \* "Michael Jackson"# Print the string for 3 times

print(" Michael Jackson \n is the best" ) # New line escape sequence

print(" Michael Jackson \t is the best" ) # Tab escape sequence

print(" Michael Jackson \\ is the best" ) # Include back slash in string

print(r" Michael Jackson \ is the best" ) # r will tell python that string will be display as raw string

a = "Thriller is the sixth studio album"

b = a.upper()# Convert all the characters in string to upper case

a = "Michael Jackson is the best"

b = a.replace('Michael', 'Janet') # Replace the old substring with the new target substring is the segment has been found in the string

name = "Michael Jackson"

name.find('el') # Find the substring in the string.

name.find('Jack')

print("\\\\") or print(r"\ ") #Print out a backslash

TUPLE

Tuple1=(0,1,2,3) #create the Tuple and assign it to the variable ’tuple’

Tuple1[:2] #Find the first two elements of the Tuple

Tuple1[0]="A" #Change the first element of the list to an uppercase “A”

type(A) # Print the type of the tuple you created

print(tuple1[1]) # Print the variable on 2nd index

print(type(tuple1[2])) # Print the type of value on 3rd index

tuple1[-1] # Use negative index to get the value of the last element

tuple1[-2] # Use negative index to get the value of the second last element

tuple2 = tuple1 + ("hard rock", 10) # Concatenate two tuples

tuple2[3:5] # Slice from index 3 to index 4

len(tuple2) # Get the length of tuple

sorted(Ratings) # Sort the tuple

NestedT =(1, 2, ("pop", "rock") ,(3,4),("disco",(1,2))) # Create a nest tuple

NestedT[2][1][1] # Print the second element in the second nested tuples

genres\_tuple.index("disco") # Find the first index of "disco"

LIST

L = ["Michael Jackson", 10.1,1982,"MJ",1] # Create a list

print('the same element using negative and positive indexing:\n Postive:' , L[0] , '\n Negative:' , L[-3] ) # Print the elements on 1st index

L[3:5]

L.extend(['pop', 10]) # Use extend to add elements to list

L.append(['pop', 10]) # Use append to add elements to list ['Michael Jackson', 10.2, ['pop', 10]]

A = ["disco", 10, 1.2]

A[0] = 'hard rock' # Change the element based on the index

del(A[0]) # Delete the element based on the index

'hard rock'.split()# Split the string, default is by space

'A,B,C,D'.split(',') # Split the string by comma

A = ["hard rock", 10, 1.2]

B = A

A[0] = "banana"

#B[0]: “banana” #change accordingly

B = A[:]

A[0] = "hard rock"

#B[0]: “banana” #doesn’t change accordingly

A = [1, 'a']

B = [2, 1, 'd']

A+B #Concatenate 2 lists

SET

set(['A','B','C','A','B','C'])

S.add('D') #Add the string “D” to the set S

A **&** B #intersection of set A and B

set1 = {"pop", "rock", "soul", "hard rock", "rock", "R&B", "rock", "disco"}

album\_list = []

album\_set = set(album\_list) # Convert list to set

A = set(["Thriller", "Back in Black", "AC/DC"])# Sample set

A.add("NSYNC") # Add element to set

A.remove("NSYNC") # Remove the element from set

"AC/DC" in A # Verify if the element is in the set

intersection = album\_set1 & album\_set2 # Find the intersections

album\_set1.difference(album\_set2) # Find the difference in set1 but not set2

album\_set2.difference(album\_set1) # Find the difference in set2 but not set1

album\_set1.intersection(album\_set2) # Use intersection method to find the intersection of album\_list1 and album\_list2

album\_set1.union(album\_set2) # Find the union of two sets

set(album\_set1).issuperset(album\_set2) # Check if superset

set(album\_set2).issubset(album\_set1) # Check if subset

DICTIONARY

D={'a':0,'b':1,'c':2}

D["a"] #Find the value for the key ‘a’

D.keys() #Find the keys of the dictionary D

Dict = {"key1": 1, "key2": "2", "key3": [3, 3, 3], "key4": (4, 4, 4), ('key5'): 5, (0, 1): 6}

Dict["key1"] # Access to the value by the key

Dict[(0, 1)] # Access to the value by the key

release\_year\_dict['Thriller'] # Get value by keys

release\_year\_dict.keys() # Get all the keys in dictionary

release\_year\_dict.values() # Get all the values in dictionary

release\_year\_dict['Graduation'] = '2007'# Append value with key into dictionary

del(release\_year\_dict['Graduation']) # Delete entries by key

'The Bodyguard' in release\_year\_dict # Verify the key is in the dictionary

CONDITIONS AND BRANCHING

x='a'

if(x=='a'):

print("this is a")

else:

print("this is not a")

# Elif statment example

age = 18

if age > 18:

print("you can enter" )

elif age == 18:

print("go see Pink Floyd")

else:

print("go see Meat Loaf" )

print("move on")

# “AND” Condition statement example

album\_year = 1980

if(album\_year > 1979) and (album\_year < 1990):

print ("Album year was in between 1980 and 1989")

print("")

print("Do Stuff..")

# “OR” Condition statement example

album\_year = 1990

if(album\_year < 1980) or (album\_year > 1989):

print ("Album was not made in the 1980's")

else:

print("The Album was made in the 1980's ")

# “IF NOT” Condition statement example

album\_year = 1983

if not (album\_year == '1984'):

print ("Album year is not 1984")

Example:

#If the rating is larger then eight print “this album is amazing”. If the rating is less than or equal to 8 print “this album is ok”.

rating = 8.5

if rating > 8:

print("This album is Amazing!")

else:

print("this album is ok")

#determine if an album came out before 1980 or in the years: 1991 or 1993 => TRUE

year = 1983

if year<1980 or year == 1991 or year == 1993:

print("True")

else:

print("False")

LOOPS

#Use loops to print out the elements in the list A

A=[3,4,5]

for i in A:

print(i)

#find the value x that will print out the sequence 1, 2, …, 10

x=11

y=1

while(y != x):

print(y)

y=y+1

# For loop example

dates = [1982,1980,1973]

N = len(dates)

for i in range(N):

print(dates[i])

#print out a sequence of numbers from 0 to 7

for i in range(0, 8):

print(i)

#change the elements in a list:

squares = ['red', 'yellow', 'green', 'purple', 'blue']

for i in range(0, 5):

print("Before square ", i, 'is', squares[i])

squares[i] = 'white'

print("After square ", i, 'is', squares[i])

#access the index and the elements of a list as follows

squares=['red', 'yellow', 'green', 'purple', 'blue']

for i, square in enumerate(squares):

print(i, square)

# While Loop Example

dates = [1982, 1980, 1973, 2000]

i = 0

year = dates[0]

while(year != 1973):

print(year)

i = i + 1

year = dates[i]

print("It took ", i ,"repetitions to get out of loop.")

#prints out all the element between **-5** and **5**

for i in range (-5,6):

print(i)

#Write a while loop to display the values of the Rating of an album playlist stored in the list PlayListRatings. If the score is less than 6, exit the loop.

PlayListRatings = [10, 9.5, 10, 8, 7.5, 5, 10, 10]

i = 1

Rating = PlayListRatings[0]

while(i < len(PlayListRatings) and Rating >= 6):

print(Rating)

Rating = PlayListRatings[i]

i = i + 1

#Write a while loop to copy the strings 'orange' of the list squares to the list new\_squares. Stop and exit the loop if the value on the list is not 'orange'

squares = ['orange', 'orange', 'purple', 'blue ', 'orange']

new\_squares = []

i=0

for i in squares:

while (i<len(squares) and squares[i]=='orange'):

new\_squares.append(square[i])

i=i+1

print (new\_squares)

FUNCTIONS

#Complete the function f so that it returns the product of a and b

def f(a,b):

return a\*b

#Complete the function g such that the input c is a list of integers and the output is the sum of all the elements in the list

def g(c):

return sum(c)

# First function example: Add 1 to a and store as b

def add(a):

b = a + 1

print(a, "if you add one", b)

return(b)

c=add(10) #result: 10 if you add one 11

# Define a function for multiple two numbers

def Mult(a, b):

c = a \* b

return(c)

print('This is not printed')

result = Mult(12,2)

print(result)

#If there is no return statement, the function returns None

def MJ1():

print('Michael Jackson')

return(None)

# Using if/else Statements and Loops in Functions

def type\_of\_album(artist, album, year\_released):

print(artist, album, year\_released)

if year\_released > 1980:

return "Modern"

else:

return "Oldie"

x = type\_of\_album("Michael Jackson", "Thriller", 1980)

print(x)

# Print the list using for loop in Function

def PrintList(the\_list):

for element in the\_list:

print(element)

PrintList(['1', 1, 'the man', "abc"])

# Example for setting param with default value

def isGoodRating(rating=4):

if(rating < 7):

print("this album sucks it's rating is",rating)

else:

print("this album is good its rating is",rating)

isGoodRating()

isGoodRating(10)

#global variable

artist = "Michael Jackson"

def printer(artist):

global internal\_var

internal\_var= "Whitney Houston"

print(artist,"is an artist")

printer(artist)

printer(internal\_var)

# Example of global variable and local variable with the same name

myFavouriteBand = "AC/DC"

def getBandRating(bandname):

myFavouriteBand = "Deep Purple"

if bandname == myFavouriteBand:

return 10.0

else:

return 0.0

print("AC/DC's rating is:",getBandRating("AC/DC"))

print("Deep Purple's rating is: ",getBandRating("Deep Purple"))

print("My favourite band is:",myFavouriteBand)

#Collections and Functions

def printAll(\*args): # All the arguments are 'packed' into args which can be treated like a tuple

print("No of arguments:", len(args))

for argument in args:

print(argument)

printAll('Horsefeather','Adonis','Bone') #printAll with 3 arguments

printAll('Sidecar','Long Island','Mudslide','Carriage') #printAll with 4 arguments

#The arguments can also be packed into a dictionary

def printDictionary(\*\*args):

for key in args:

print(key + " : " + args[key])

printDictionary(Country='Canada',Province='Ontario',City='Toronto')

EXCEPT

#if we divide by zero, try running the following block of code, with b as a number an exception will only be Raised if b, is zero

a = 1

try:

b = int(input("Please enter a number to divide a"))

a = a/b

print("Success a=",a)

except ZeroDivisionError:

print("The number you provided cant divide 1 because it is 0")

except ValueError:

print("You did not provide a number")

except:

print("Something went wrong")

#using “else” and “finally”

a = 1

try:

b = int(input("Please enter a number to divide a"))

a = a/b

except ZeroDivisionError:

print("The number you provided cant divide 1 because it is 0")

except ValueError:

print("You did not provide a number")

except:

print("Something went wrong")

else:

print("success a=",a)

finally:

print("Processing Complete")