**PYTHON**

***Author: Wahidur Rahman***

* High level programming language.
* Interpreted
* Object oriented

<https://docs.python.org/3.13/tutorial/index.html>

to print something 🡪 print()

print (123)

string 🡪 print ("Wahidur Rahman")

**Backslash characters and comments**

Comment:

Single line comment 🡪#abc

Multiple line comment 🡪

*'''  
abc  
def  
ghi  
'''*

Backslash character/Escape sequence:

Print new line : \n

Tab distance : \t

To print double cotation(“”)🡪print ("\"Wahidur Rahman\"")

\”=”

\’=’

**Variables and Data Types**

name = ("Wahidur Rahman")  
age = 18  
print ("Our new student name is " + name)  
print(name + " is " , age , " years old.")  
print("At the age of ", age , "he learned python.")  
print(name + " lives in canad")

String+integer variable print in same line🡪string,int,string

print(name + " is " , age , " years old.")

**Basic Numerical Operations**

**Let,** a= 20, b = 3

+ 🡪 print(a + b) 🡪23

- 🡪 print(a - b) 🡪17

/ 🡪 print(a / b) 🡪6.666666666666667

\* 🡪 print(a \* b) 🡪60

% 🡪 print(a % b) 🡪2

//(Floor) 🡪 print(a // b) 🡪 6

\*\*(Exponentiation) 🡪 print(a // b) 🡪 8000 (it’s is power 2n)

**Getting User Input**

To take input 🡪 num=input()

Print with direction about input🡪 name = input("Enter your name: ")

**Type Casting**

Select type getting input 🡪 num1 = int(input("Enter first number: "))

Convert 🡪sum = num1 + int(num2)

**Math related Library functions**

Library 🡪 from math import \*

from math import \*  
print(max(10, 20))  
print(min(10, 20))  
print(abs(-4))  
print(pow(2, 3))  
print(sqrt(25))  
print(round(3.8))  
print(floor(3.8))  
print(ceil(3.8))

**Formatted String | Type function**

Type function 🡪 type(num)

num = 3.8  
print(type(num))

format string 🡪 output = 2 + 5 = 7

num1 = 2  
num2 = 5  
print(f"{num1} + {num2} = {num1 + num2}")

print("Wahidur Rahman", end=" ")  
print("123456789")

output: Wahidur Rahman 123456789

**if, else statement**

if 3 > 4:  
 print(3)  
else:  
 print(4)

**elif statement**

if 🡪 elif 🡪 else

n =5  
3):  
 print(3)  
elif(n>4):  
 print(4)  
else:if(n>  
 print(5)

**Ternary Operator**

num1 = 2  
num2 = 4  
print(num1 if num1<num2 else num2)

**Logical operator**

if(num1<num2 and num2< 5):  
 print(num1)  
elif(num1<num2 or num2>2):  
 print(num2)

**List**

List is basically array

list = ["c", "c++", "java", "python", "c#", "c++", "java", "python"]  
print(list)  
print(list[0])

print from fixed index 🡪 print(list[2:])

print from last 🡪 print(list[-1])

in operator 🡪 to search something in the list 🡪 output : true/false

print("python" in list)

not in 🡪 to check if the item is not exist in list

* Exist 🡪 false
* Not exist 🡪 true

print("nodejs" not in list)

output 🡪 true

length of list 🡪 print(len(list))

.append() function 🡪 to add something on list and the item will add last in the list

Insert 🡪 name\_of\_list(index number,“item”)🡪 insert something on fixed position

list.insert(2, "OS")

.remove() 🡪 to remove object 🡪 list.remove("java")

.sort()🡪 sort something 🡪 in string it will first sort capital letter then small letter

list.sort()

.reverse()🡪 to make it reverse 🡪 list.reverse()

.pop()🡪 to delete last item 🡪 list.pop()

.clear()🡪 clear all object of list 🡪item.clear()

.copy()🡪 to copy one list to another list🡪 list2 = list.copy()

.index(“item\_name”)🡪 to find position of item🡪 print(list2.index("java"))

.count(“item”)🡪to count number of item🡪 print(list2.count("python"))

**Range function**

range(st, end)🡪 num = list(range(5, 10))

range(st, end, gap)🡪 num = list(range(5, 20, 2))

**for Loop**

num = {10, 30, 50, 5, 3}  
sum = 0  
for x in num:  
 print(x)  
 sum = sum + x  
print("sum =",sum)

**pattern printing:**

n = int(input("Enter row number:"))  
for i in range(1, n + 1):  
 print((n - i) \* " ", i \* "\*")

**Guessing Game**

import random  
guessnumber = int(input("Guess your number(1-5):"))  
randomnumber = random.randint(1, 5)  
if guessnumber == randomnumber:  
 print("you won!")  
else:  
 print("you lost!", "random number was", randomnumber)

**List as input from user**

n = input("Enter a text of numbers:")  
list = n.split()  
sum = 0  
for num in list:  
 sum = sum + int(num)  
print(sum)

**Matrix**

2D list

matrix = [  
 [1, 2, 3],  
 [4, 5, 6],  
]  
print(matrix)

**Dictionaries**

studentId = {  
 "101" : "Wahidur Rahman",  
 "102" : "nafiz",  
 "103" : "Nola saikat"  
}  
print(studentId["102"])  
print(studentId.get("106", "Not a valid key"))

**Tuples**

Items can’t be changed

students = (  
 "Wahidur Rahman", "Nola saikat", "nafiz"  
)  
print(students)

2D

student = (  
 ("Wahidur Rahman", 21, 2.0),  
 ("Nola saikat", 21, 3.50),  
 ("nafiz", 21, 4.00),  
)  
print(student)

print(student[1:])

**Set**

* Unordered collection of item
* Can’t be store duplicate value

A set can be created using

1. Curly braces
2. num = {1,3,4,5,5}  
   print(num)
3. Set function

num2 = set([4,5,6,7])  
print(num2)

Add and remove something from set

num2 = set([4,5,6,7])  
print(num2)  
num2.add(8)  
num2.remove(5)  
print(num2)

Search element in set 🡪print(5 in num2)

Operation

Union”|” 🡪 all element both set 🡪print(num1 | num2)

Intersection “&” 🡪 common element 🡪print(num1 & num2)

Difference“-”🡪delete common values from 1st set🡪

print(num1 - num2)

**Stack And Queue**

Stack🡪 push(), pop() 🡪 stackName.append(\_) = push()

books= []  
books.append("Learn C")  
books.append("Learn C++")  
books.append("Learn Java")  
print(books)  
  
books.pop()  
print(books)  
print("Now the top book is-", books[-1])  
  
books.pop()  
print("Now the top book is-", books[-1])  
  
books.pop()  
if not books:  
 print("Nothing to see here.")

Queue

from collections import deque  
bank = deque(["Nola saikat", "BBB", "BBC"])  
bank.popleft()  
print(bank)  
  
if not bank:  
 print("Nothing to see here.")

**Function**

def function\_name (parameter1, parameter2…..)

return type

def add (a, b):  
 return a + b  
  
x = int(input())  
y = int(input())  
print(add(x, y))

**xxargs and xxxargs**

xxargs 🡪 works like tuples

def student(\*details):  
 print (details)  
  
student(101, "Nola")  
student(101, "Nola", 3.75)

xxxargs 🡪 works like dictionary

def student(\*\*details):  
 print (details)  
 print(details["name"])  
  
student(ID=101, name="Nola", CGPA=3.75)

**Lambda Functions**

* A function with no name (Anonymous Function).
* Not powerful as named function.
* It can work with single expression / single line of code.

Lambda parameter : expression

print((lambda a,b : a\*a + 2\*a\*b + b\*b) (2,3))

**map and filter function**

map 🡪 map(function,list)

def square(a):  
 return a\*a  
num = [1,2,3,4,5,6,7,8,9]  
result = list(map(square, num))  
print(result)

filter

result = list(filter(lambda x: x%2==0, num))  
print(result)

**List Comprehensions**

Syntax = **[**Expression for item in list**]**

result = [x\*x for x in num]  
print(result)

result =[x for x in num if x%2==0]  
print(result)

**Zip Function**

roll = [104, 105, 106, 107, 108, 109, 110]  
name = ["nola", "nafiz", "BBB", "BBC", "monkey", "dart irfan", "rivu"]  
print(list(zip(roll, name)))

**Recursion**

* Recursive call
* Base case

def fact (n):  
 if n == 1:  
 return 1  
 else:  
 return n \* fact(n - 1)  
print(fact(5))

**Reading a file**

file = open("student.txt", "r")  
  
print(file.readable())  
text = file.read()  
print(text)  
size = len(text)  
print(size)  
file.close()

Convert it list

file = open("student.txt", "r") #list  
text = file.readlines()  
print(text)  
file.close()

Print one fixed index line from file

file = open("student.txt", "r") #index print  
text = file.readlines()[0]  
print(text)  
file.close()

Using for loop to print full file one by one

file = open("student.txt", "r")  
for line in file:  
 print(line)  
file.close()

**writing in a file**

file = open("student.txt", "a")  
file.write("\nNoman - Topper")  
file.close()

by using ‘w’ it will create auto file and write but all information of previous will be deleted.

**Exception Handling (part-1)**

Errors

To handle exception

try:  
 list = [10, 0, 30]  
 result = list[0] / list[1]  
 print(result)  
 print("Done")  
except ZeroDivisionError:  
 print("Division by zero is not possible")

This will print finally, even exception handing is not working

try:  
 list = [10, 0, 30]  
 result = list[0] / list[3]  
 print(result)  
 print("Done")  
except ZeroDivisionError:  
 print("Division by zero is not possible")  
finally:  
 print("successful")

Another Version:

def voter(age):  
 if age < 18:  
 raise ValueError("Invalid age")  
 return "You are allowed to vote"  
  
try:  
   
 print(voter(17))  
except ValueError as e:  
 print(e)

**Swapping**

a = 10  
b = 20  
a, b = b, a  
print("a =", a)  
print("b =", b)