# PYTHON BASIC PRACTICE ASSIGNMENTS

1 **Python Program to Print Hello world!**

A simple program that displays “Hello, World!”. It's often used to illustrate the syntax of the language.

```
# This program prints Hello, world!

print('Hello, world!')
```

2**Python Program to Add Two Numbers**

In this program, you will learn to add two numbers and display it using print() function.

```
# This program adds two numbers

num1 = 1.5

num2 = 6.3

# Add two numbers

sum = num1 + num2

# Display the sum

print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))
```

3 **Add Two Numbers With User Input**

```
# Store input numbers
num1 = input('Enter first number: ')
num2 = input('Enter second number: ')


# Add two numbers
```

```
sum = float(num1) + float(num2)
# Display the sum
print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))
```

4 **Program to Find the Square Root**

```
# Python Program to calculate the square root

# Note: change this value for a different result

num = 8

# To take the input from the user

num = float(input('Enter a number: '))

num_sqrt = num ** 0.5

print('The square root of %0.3f is %0.3f'%(num ,num_sqrt))
```

5 **Program to Calculate the Area of a Triangle**

```
# Python Program to find the area of triangle
a = 5
b = 6
c = 7
a = float(input('Enter first side: '))
b = float(input('Enter second side: '))
```

```
c = float(input('Enter third side: '))

# calculate the semi-perimeter
s = (a + b + c) / 2

# calculate the area
area = (s*(s-a)*(s-b)*(s-c)) ** 0.5
print('The area of the triangle is %0.2f' %area)
```

6 # **Python program to swap two variables**

```
x = 5
y = 10
# To take inputs from the user
x = input('Enter value of x: ')
y = input('Enter value of y: ')
# create a temporary variable and swap the values
temp = x
x = y
y = temp
print('The value of x after swapping: {}'.format(x))
print('The value of y after swapping: {}'.format(y))
```

7 # **Program to generate a random number between 0 and 9**

```
# importing the random module
import random
print(random.randint(0,9))
```

8 # **Taking kilometers input from the user**

```
kilometers = float(input("Enter value in kilometers: "))
# conversion factor
conv_fac = 0.621371
```

```
# calculate miles
miles = kilometers * conv_fac
print('%0.2f kilometers is equal to %0.2f miles' %(kilometers,miles))
```

9 # **Python Program to convert temperature in celsius to fahrenheit**

```
# change this value for a different result
celsius = 37.5
# calculate fahrenheit
fahrenheit = (celsius * 1.8) + 32
print('%0.1f degree Celsius is equal to %0.1f degree Fahrenheit' %(celsius,fahrenheit))
```

10 **# Python Program to convert temperature in celsius to fahrenheit**

```
# change this value for a different result
celsius = 37.5

# calculate fahrenheit
fahrenheit = (9/5)*celsius+32
print('%0.1f degree Celsius is equal to %0.1f degree Fahrenheit' %(celsius,fahrenheit))
```

11 **Program to check whether a number entered by the user is positive, negative or zero.**

```
num = float(input("Enter a number: "))
if num > 0:
  print("Positive number")
elif num == 0:
  print("Zero")
else:
  print("Negative number")
```

12 Program with **Using Nested if**

```
num = float(input("Enter a number: "))
if num >= 0:
   if num == 0:
      print("Zero")
   else:
      print("Positive number")
else:
   print("Negative number")
```

13 # **Python program to check if the input number is odd or even.**

```
# A number is even if division by 2 gives a remainder of 0.
# If the remainder is 1, it is an odd number.

num = int(input("Enter a number: "))
if (num % 2) == 0:
   print("{0} is Even".format(num))
else:
   print("{0} is Odd".format(num))
```

14 # **Python program to check if year is a leap year or not**

```
year = 2000

# To get year (integer input) from the user
# year = int(input("Enter a year: "))

# divided by 100 means century year (ending with 00)
# century year divided by 400 is leap year
```

```
if (year % 400 == 0) and (year % 100 == 0):
    print("{0} is a leap year".format(year))

# not divided by 100 means not a century year
# year divided by 4 is a leap year
elif (year % 4 ==0) and (year % 100 != 0):
    print("{0} is a leap year".format(year))

# if not divided by both 400 (century year) and 4 (not century year)
# year is not leap year
else:
    print("{0} is not a leap year".format(year))
```

15 **# Python program to find the largest number among the three input numbers**

```
# change the values of num1, num2 and num3
# for a different result
num1 = 10
num2 = 14
num3 = 12

# uncomment following lines to take three numbers from user
#num1 = float(input("Enter first number: "))
#num2 = float(input("Enter second number: "))
#num3 = float(input("Enter third number: "))

if (num1 >= num2) and (num1 >= num3):
   largest = num1
elif (num2 >= num1) and (num2 >= num3):
   largest = num2
```

```
else:
  largest = num3

print("The largest number is", largest)
```

16 **# Program to check if a number is prime or not**

```
num = 29

# To take input from the user
#num = int(input("Enter a number: "))

# define a flag variable
flag = False

# prime numbers are greater than 1
if num > 1:
   # check for factors
   for i in range(2, num):
      if (num % i) == 0:
         # if factor is found, set flag to True
         flag = True
         # break out of loop
         break

# check if flag is True
if flag:
   print(num, "is not a prime number")
else:
   print(num, "is a prime number")
```

17 # **Program to check if a number is prime or not**

```
num = 407

# To take input from the user
#num = int(input("Enter a number: "))

# prime numbers are greater than 1
if num > 1:
   # check for factors
   for i in range(2,num):
      if (num % i) == 0:
         print(num,"is not a prime number")
         print(i,"times",num//i,"is",num)
         break
   else:
      print(num,"is a prime number")

# if input number is less than
# or equal to 1, it is not prime
else:
   print(num,"is not a prime number")
```

18 # **Python program to display all the prime numbers within an interval**

```
lower = 900
upper = 1000
```

```
print("Prime numbers between", lower, "and", upper, "are:")

for num in range(lower, upper + 1):
  # all prime numbers are greater than 1
  if num > 1:
     for i in range(2, num):
        if (num % i) == 0:
           break
     else:
        print(num)
```

19 **Factorial - The factorial of a number is the product of all the integers from 1 to that number.**

**For example, the factorial of 6 is `1*2*3*4*5*6 = 720`. Factorial is not defined for negative numbers, and the factorial of zero is one, `0! = 1`**

```
# Python program to find the factorial of a number provided by the user.

# change the value for a different result
num = 7

# To take input from the user
#num = int(input("Enter a number: "))

factorial = 1

# check if the number is negative, positive or zero
if num < 0:
```

```
  print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
  print("The factorial of 0 is 1")
else:
  for i in range(1,num + 1):
     factorial = factorial*i
  print("The factorial of",num,"is",factorial)
```

20 # **Python program to find the factorial of a number provided by the user**

**# using recursion**

```
def factorial(x):
   """This is a recursive function
   to find the factorial of an integer"""

   if x == 1:
      return 1
   else:
      # recursive call to the function
      return (x * factorial(x-1))


# change the value for a different result
num = 7

# to take input from the user
# num = int(input("Enter a number: "))

# call the factorial function
result = factorial(num)
print("The factorial of", num, "is", result)
```

21 # **Multiplication table (from 1 to 10) in Python**

```
num = 12

# To take input from the user
# num = int(input("Display multiplication table of? "))

# Iterate 10 times from i = 1 to 10
for i in range(1, 11):
   print(num, 'x', i, '=', num*i)
```

**22A Fibonacci sequence is the integer sequence of 0, 1, 1, 2, 3, 5, 8....**

**The first two terms are 0 and 1. All other terms are obtained by adding the preceding two terms. This means to say the nth term is the sum of (n-1)th and (n-2)th term.**

**# Program to display the Fibonacci sequence up to n-th term**

```
nterms = int(input("How many terms? "))

# first two terms
n1, n2 = 0, 1
count = 0

# check if the number of terms is valid
if nterms <= 0:
   print("Please enter a positive integer")
# if there is only one term, return n1
elif nterms == 1:
   print("Fibonacci sequence upto",nterms,":")
   print(n1)
```

```
# generate fibonacci sequence
else:
  print("Fibonacci sequence:")
  while count < nterms:
     print(n1)
     nth = n1 + n2
     # update values
     n1 = n2
     n2 = nth
     count += 1
```

23 **A positive integer is called an Armstrong number of order n if**

**In case of an Armstrong number of 3 digits, the sum of cubes of each digit is equal to the number itself. For example:**

```
153 = 1*1*1 + 5*5*5 + 3*3*3  // 153 is an Armstrong number.
```

**# Python program to check if the number is an Armstrong number or not**

```
# take input from the user
num = int(input("Enter a number: "))

# initialize sum
sum = 0

# find the sum of the cube of each digit
temp = num
while temp > 0:
  digit = temp % 10
  sum += digit ** 3
```

```
  temp //= 10

# display the result
if num == sum:
  print(num,"is an Armstrong number")
else:
  print(num,"is not an Armstrong number")
```

24 **armstrong number**

```
num = 1634

# Changed num variable to string,
# and calculated the length (number of digits)
order = len(str(num))

# initialize sum
sum = 0

# find the sum of the cube of each digit
temp = num
while temp > 0:
  digit = temp % 10
  sum += digit ** order
  temp //= 10

# display the result
if num == sum:
  print(num,"is an Armstrong number")
else:
  print(num,"is not an Armstrong number")
```

25 **find all the Armstrong numbers present in between two intervals in Python.**

```
# Program to check Armstrong numbers in a certain interval

lower = 100
upper = 2000

for num in range(lower, upper + 1):

   # order of number
   order = len(str(num))

   # initialize sum
   sum = 0

   temp = num
   while temp > 0:
      digit = temp % 10
      sum += digit ** order
      temp //= 10

   if num == sum:
      print(num)
```

26

**# Sum of natural numbers up to num**

```
num = 16

if num < 0:
   print("Enter a positive number")
```

```
else:
  sum = 0
  # use while loop to iterate until zero
  while(num > 0):
     sum += num
     num -= 1
  print("The sum is", sum)
```

27 **ASCII stands for American Standard Code for Information Interchange.**

**It is a numeric value given to different characters and symbols, for computers to store and manipulate. For example, the ASCII value of the letter `'A'` is 65.**

**# Program to find the ASCII value of the given character**

```
c = 'p'
print("The ASCII value of '" + c + "' is", ord(c))
```

```
28   >>> chr(65)
'A'
>>> chr(120)
'x'
>>> chr(ord('S') + 1)
'T'
```

29

**The highest common factor (H.C.F) or greatest common divisor (G.C.D) of two numbers is the largest positive integer that perfectly divides the two given numbers. For example, the H.C.F of 12 and 14 is 2.**

**# Python program to find H.C.F of two numbers**

```
# define a function
def compute_hcf(x, y):

# choose the smaller number
   if x > y:
      smaller = y
   else:
      smaller = x
   for i in range(1, smaller+1):
      if((x % i == 0) and (y % i == 0)):
         hcf = i
   return hcf

num1 = 54
num2 = 24

print("The H.C.F. is", compute_hcf(num1, num2))
```

30 **The least common multiple (L.C.M.) of two numbers is the smallest positive integer that is perfectly divisible by the two given numbers.**

**For example, the L.C.M. of 12 and 14 is 84.**

```
# Python Program to find the L.C.M. of two input number

def compute_lcm(x, y):

  # choose the greater number
  if x > y:
```

```
    greater = x
  else:
    greater = y

  while(True):
    if((greater % x == 0) and (greater % y == 0)):
      lcm = greater
      break
    greater += 1

  return lcm

num1 = 54
num2 = 24

print("The L.C.M. is", compute_lcm(num1, num2))
```

**31 # Python Program to find the factors of a number**

```
# This function computes the factor of the argument passed
def print_factors(x):
  print("The factors of",x,"are:")
  for i in range(1, x + 1):
    if x % i == 0:
      print(i)

num = 320

print_factors(num)
```

32 **# Program make a simple calculator**

```
# 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")

while True:
   # take input from the user
   choice = input("Enter choice(1/2/3/4): ")

   # check if choice is one of the four options
```

```
if choice in ('1', '2', '3', '4'):
    num1 = float(input("Enter first number: "))
    num2 = float(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))

    # check if user wants another calculation
    # break the while loop if answer is no
    next_calculation = input("Let's do next calculation? (yes/no): ")
    if next_calculation == "no":
      break

else:
    print("Invalid Input")
```

33 **A palindrome is a string that is the same read forward or backward.**

**For example, `"dad"` is the same in forward or reverse direction. Another example is "aibohphobia", which literally means, an irritable fear of palindromes.**

**# Program to check if a string is palindrome or not**

```
my_str = 'aIbohPhoBiA'

# make it suitable for caseless comparison
my_str = my_str.casefold()

# reverse the string
rev_str = reversed(my_str)

# check if the string is equal to its reverse
if list(my_str) == list(rev_str):
   print("The string is a palindrome.")
else:
   print("The string is not a palindrome.")
```

34 **Sometimes, we may wish to break a sentence into a list of words.**

**In such cases, we may first want to clean up the string and remove all the punctuation marks. Here is an example of how it is done.**

```
# define punctuation
punctuations = '''!()-[]{};:'"\,<>./?@#$%^&*_~'''

my_str = "Hello!!!, he said ---and went."

# To take input from the user
# my_str = input("Enter a string: ")
```

```
# remove punctuation from the string
no_punct = ""
for char in my_str:
   if char not in punctuations:
       no_punct = no_punct + char

# display the unpunctuated string
print(no_punct)
```

35 **# Program to sort alphabetically the words form a string provided by the user**

```
my_str = "Hello this Is an Example With cased letters"

# To take input from the user
#my_str = input("Enter a string: ")

# breakdown the string into a list of words
words = [word.lower() for word in my_str.split()]

# sort the list
words.sort()

# display the sorted words

print("The sorted words are:")
for word in words:
   print(word)
```

36 **# Program to perform different set operations like in mathematics**

```
# define three sets
E = {0, 2, 4, 6, 8};
N = {1, 2, 3, 4, 5};

# set union
print("Union of E and N is",E | N)

# set intersection
print("Intersection of E and N is",E & N)

# set difference
print("Difference of E and N is",E - N)

# set symmetric difference
print("Symmetric difference of E and N is",E ^ N)
```

**37 # Program to count the number of each vowels**

```
# string of vowels
vowels = 'aeiou'

ip_str = 'Hello, have you tried our tutorial section yet?'

# make it suitable for caseless comparisions
ip_str = ip_str.casefold()

# make a dictionary with each vowel a key and value 0
count = {}.fromkeys(vowels,0)

# count the vowels
for char in ip_str:
```

```
    if char in count:
        count[char] += 1

print(count)
```

**38 Programs to print triangles using *, numbers and characters**

```
rows = int(input("Enter number of rows: "))

for i in range(rows):
    for j in range(i+1):
        print("* ", end="")
    print("\n")
```

**39 Make pyramid**

```
rows = int(input("Enter number of rows: "))

for i in range(rows):
    for j in range(i+1):
        print(j+1, end=" ")
    print("\n")
```

40 **Program to print half pyramid using alphabets**

```
rows = int(input("Enter number of rows: "))

ascii_value = 65

for i in range(rows):
    for j in range(i+1):
```

```
        alphabet = chr(ascii_value)
        print(alphabet, end=" ")

    ascii_value += 1
    print("\n")
```

41 **Programs to print inverted half pyramid using * and numbers**

**inverted half pyramid using ***

```
rows = int(input("Enter number of rows: "))

for i in range(rows, 0, -1):
    for j in range(0, i):
        print("* ", end=" ")

    print("\n")
```

42 **Inverted half pyramid using numbers**

```
rows = int(input("Enter number of rows: "))

for i in range(rows, 0, -1):
    for j in range(1, i+1):
        print(j, end=" ")

    print("\n")
```

43 **Program to print full pyramid using ***

```
rows = int(input("Enter number of rows: "))

k = 0

for i in range(1, rows+1):
    for space in range(1, (rows-i)+1):
        print(end="  ")

    while k!=(2*i-1):
        print("* ", end="")
        k += 1

    k = 0
    print()
```

**44 Full Pyramid of Numbers**

```
rows = int(input("Enter number of rows: "))

k = 0
count=0
count1=0

for i in range(1, rows+1):
    for space in range(1, (rows-i)+1):
        print("  ", end="")
        count+=1
```

```
    while k!=((2*i)-1):
        if count<=rows-1:
            print(i+k, end=" ")
            count+=1
        else:
            count1+=1
            print(i+k-(2*count1), end=" ")
        k += 1

    count1 = count = k = 0
    print()
```

45 **Inverted full pyramid of ***

```
rows = int(input("Enter number of rows: "))

for i in range(rows, 1, -1):
    for space in range(0, rows-i):
        print("  ", end="")
    for j in range(i, 2*i-1):
        print("* ", end="")
    for j in range(1, i-1):
        print("* ", end="")
    print()
```

46 **you will learn to merge two dictionaries into one in Python programming.**

**Using the | Operator**

```
dict_1 = {1: 'a', 2: 'b'}
```

```
dict_2 = {2: 'c', 4: 'd'}

print(dict_1 | dict_2)
```

47 **Using the ** Operator**

```
dict_1 = {1: 'a', 2: 'b'}
dict_2 = {2: 'c', 4: 'd'}

print({**dict_1, **dict_2})
```

48 **Using copy() and update()**

```
 dict_1 = {1: 'a', 2: 'b'}
dict_2 = {2: 'c', 4: 'd'}

dict_3 = dict_2.copy()
dict_3.update(dict_1)

print(dict_3)
```

49 **you will learn to concatenate two lists in Python.**

```
list_1 = [1, 'a']
list_2 = [3, 4, 5]

list_joined = list_1 + list_2
print(list_joined)
```

50 **using extend**

```
list_1 = [1, 'a']
```

```
list_2 = [1, 2, 3]


list_2.extend(list_1)
print(list_2)
```

**51 you will learn to check if a string is a number (float).**

```
def isfloat(num):
    try:
        float(num)
        return True
    except ValueError:
        return False

print(isfloat('s12'))
print(isfloat('1.123'))
```

52 **you will learn to count the occurrence of an item in a list.**

```
freq = ['a', 1, 'a', 4, 3, 2, 'a'].count('a')
print(freq)
```

53 **learn to get the full path of the current working directory**

```
import pathlib
# path of the given file
print(pathlib.Path("my_file.txt").parent.absolute())

# current working directory
print(pathlib.Path().absolute())
```

54

```
import os
```

```
# path of the given file
print(os.path.dirname(os.path.abspath("my_file.txt")))

# current working directory
print(os.path.abspath(os.getcwd()))
```

55

**you will learn to read a file line by line into a list**

Let the content of the file `data_file.txt` be

```
honda 1948
mercedes 1926
ford 1903
```

```
with open("data_file.txt") as f:
    content_list = f.readlines()

# print the list
print(content_list)

# remove new line characters
content_list = [x.strip() for x in content_list]
print(content_list)
```