**Batch: P1-3**

**Roll No.: 16014022098**

**Experiment / assignment / tutorial No. 1**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

**TITLE:** Basic concepts in python

**AIM:** 1) Program to find the distance between two points

2) Program to perform string operations.

**Expected OUTCOME of Experiment:** Use of input output function, arithmetic

operators in python and different operations on string.

# Resource Needed: Python IDE

**Theory:**

# How the input function works in Python:

* When input() function executes program flow will be stopped until the user has given an input.
* The text or message displayed on the output screen to ask a user to enter input value is optional i.e. the prompt, will be printed on the screen is optional.
* Whatever you enter as input, the input function converts it into a string. If you enter an integer value still input() function convert it into a string. You need to explicitly convert it into an integer in your code using [typecasting.](https://www.geeksforgeeks.org/taking-input-from-console-in-python/)

# Example:

Name=input("Enter your name") print('Hello, ' + Name)

Output:-

Enter your name Mahesh Hello, Mahesh

# Python Arithmetic Operators:

Assume variable **a** holds 10 and variable **b** holds 20, then

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| + Addition | Adds values on either side of the operator. | a + b = 30 |
| - Subtraction | Subtracts right hand operand from left hand operand. | a – b = -10 |
| \* Multiplication | Multiplies values on either side of the operator | a \* b = 200 |
| / Division | Divides left hand operand by right hand operand | b / a = 2 |
| % Modulus | Divides left hand operand by right hand operand and returns remainder | b % a = 0 |
| \*\* Exponent | Performs exponential (power) calculation on operators | a\*\*b =10 to the power 20 |
| // | Floor Division - The division of operands where the result is the quotient in which the digits after the decimal point are removed. But if one of the operands is negative, the result is floored, i.e., rounded away from zero (towards negative infinity) − | 9//2 = 4 and 9.0//2.0 = 4.0,  -11//3 = -4,  -11.0//3 = -4.0 |

**Strings:**

We can create string simply by enclosing characters in quotes. Python treats single quotes the same as double quotes. Creating strings is as simple as assigning a value to a variable.

Example:-

var1= “Hello World” var2=”Python Programming”

# String Special Operators:

Assume string variable **a** holds 'Hello' and variable **b** holds 'Python', then

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| + | Concatenation - Adds values on either side of the operator | a + b will give HelloPython |
| \* | Repetition - Creates new strings, concatenating multiple copies of the same string | a\*2 will give - HelloHello |
| [] | Slice - Gives the character from the given index | a[1] will give e |
| [ : ] | Range Slice - Gives the characters from the given range | a[1:4] will give ell |
| in | Membership - Returns true if a character exists in the given string | H in a will give 1 |
| not in | Membership - Returns true if a character does not exist in the given string | M not in a will give 1 |

# String Methods:

|  |  |
| --- | --- |
| **Function Name** | **Description** |
| [capitalize()](https://www.geeksforgeeks.org/string-capitalize-python/) | Converts the first character of the string to a capital (uppercase) letter |
| [casefold()](https://www.geeksforgeeks.org/casefold-string-python/) | Implements caseless string matching |
| [center()](https://www.geeksforgeeks.org/string-center-python/) | Pad the string with the specified character. |
| [count()](https://www.geeksforgeeks.org/python-string-count/) | Returns the number of occurrences of a substring in the string. |
| [encode()](https://www.geeksforgeeks.org/python-strings-encode-method/) | Encodes strings with the specified encoded scheme |
| [endswith()](https://www.geeksforgeeks.org/string-endswith-python/) | Returns “True” if a string ends with the given suffix |
| [expandtabs()](https://www.geeksforgeeks.org/python-expandtabs-method/) | Specifies the amount of space to be substituted with the “\t” symbol in the string |
| [find()](https://www.geeksforgeeks.org/python-string-find/) | Returns the lowest index of the substring if it is found |
| [format()](https://www.geeksforgeeks.org/python-format-function/) | Formats the string for printing it to console |

|  |  |
| --- | --- |
| **Function Name** | **Description** |
| [format\_map()](https://www.geeksforgeeks.org/python-string-format_map/) | Formats specified values in a string using a dictionary |
| [index()](https://www.geeksforgeeks.org/python-string-index-applications/) | Returns the position of the first occurrence of a substring in a string |
| [isalnum()](https://www.geeksforgeeks.org/python-string-isalnum/) | Checks whether all the characters in a given string is alphanumeric or not |
| [isalpha()](https://www.geeksforgeeks.org/python-string-isalpha-application/) | Returns “True” if all characters in the string are alphabets |
| [isdecimal()](https://www.geeksforgeeks.org/python-string-isdecimal/) | Returns true if all characters in a string are decimal |
| [isdigit()](https://www.geeksforgeeks.org/python-string-isdigit-application/) | Returns “True” if all characters in the string are digits |
| [isidentifier()](https://www.geeksforgeeks.org/python-string-isidentifier/) | Check whether a string is a valid identifier or not |
| [islower()](https://www.geeksforgeeks.org/python-string-islower-method/) | Checks if all characters in the string are lowercase |
| [isnumeric()](https://www.geeksforgeeks.org/python-string-isnumeric-application/) | Returns “True” if all characters in the string are numeric characters |
| [isprintable()](https://www.geeksforgeeks.org/isprintable-python-application/) | Returns “True” if all characters in the string are printable or the string is empty |
| [isspace()](https://www.geeksforgeeks.org/python-string-isspace-application/) | Returns “True” if all characters in the string are whitespace characters |

|  |  |
| --- | --- |
| **Function Name** | **Description** |
| [istitle()](https://www.geeksforgeeks.org/python-string-istitle/) | Returns “True” if the string is a title cased string |
| [isupper()](https://www.geeksforgeeks.org/python-string-isupper-method/) | Checks if all characters in the string are uppercase |
| [join()](https://www.geeksforgeeks.org/join-function-python/) | Returns a concatenated String |
| [ljust](https://www.geeksforgeeks.org/string-rjust-ljust-python/)() | Left aligns the string according to the width specified |
| [lower()](https://www.geeksforgeeks.org/python-string-lower/) | Converts all uppercase characters in a string into lowercase |
| [lstrip()](https://www.geeksforgeeks.org/python-string-lstrip-method/) | Returns the string with leading characters removed |
| [maketrans](https://www.geeksforgeeks.org/python-maketrans-translate-functions/)() | Returns a translation table |
| [partition()](https://www.geeksforgeeks.org/string-partition-python/) | Splits the string at the first occurrence of the separator |
| [replace()](https://www.geeksforgeeks.org/python-string-replace/) | Replaces all occurrences of a substring with another substring |
| [rfind()](https://www.geeksforgeeks.org/python-string-rfind/) | Returns the highest index of the substring |
| [rindex()](https://www.geeksforgeeks.org/string-rindex-python/) | Returns the highest index of the substring inside the string |
| [rjust()](https://www.geeksforgeeks.org/string-rjust-ljust-python/) | Right aligns the string according to the width specified |

|  |  |
| --- | --- |
| **Function Name** | **Description** |
| [rpartition()](https://www.geeksforgeeks.org/python-string-rpartition/) | Split the given string into three parts |
| [rsplit()](https://www.geeksforgeeks.org/python-string-rsplit/) | Split the string from the right by the specified separator |
| [rstrip()](https://www.geeksforgeeks.org/python-string-rstrip/) | Removes trailing characters |
| [splitlines()](https://www.geeksforgeeks.org/python-string-splitlines/) | Split the lines at line boundaries |
| [startswith()](https://www.geeksforgeeks.org/python-string-startswith/) | Returns “True” if a string starts with the given prefix |
| [strip()](https://www.geeksforgeeks.org/python-string-strip/) | Returns the string with both leading and trailing characters |
| [swapcase()](https://www.geeksforgeeks.org/python-string-swapcase/) | Converts all uppercase characters to lowercase and vice versa |
| [title()](https://www.geeksforgeeks.org/title-in-python/) | Convert string to title case |
| [translate()](https://www.geeksforgeeks.org/python-string-translate/) | Modify string according to given translation mappings |
| [upper()](https://www.geeksforgeeks.org/python-string-upper/) | Converts all lowercase characters in a string into uppercase |
| [zfill()](https://www.geeksforgeeks.org/python-string-zfill/) | Returns a copy of the string with ‘0’ characters padded to the left side of the string |

**Problem Definition:**

1. Create four variables x1, y1, x2 and y2. Assign each of them a value from user input using the input() function. Calculate distance [result] between two points having co- ordinates as (x1,y1) and (x2,y2) by using operators in python and basic built in math functions.

Finally, use print() to display “The distance between the two points is [result] ” in the output.

1. a) Create a variable and assign it the string “Python programming”
   1. Access the character “i” from the variable by index and print it
   2. Find the length of the string
   3. Print the slice “Python” from the variable
   4. Print the slice “program” from the variable
   5. Get the string “thing” from the variable
   6. Convert string into uppercase.
   7. Create another variable and assign it the string “ is interesting”. Now concatenate both the strings
   8. Apply different string methods given in table.

# Implementation details:

#Code by Yatendra Yadav

#PP\_Exp\_1, Program to find the distance between two points & Program to perform string operations.

print("To find the distance between two points") #getting thhe coordinates of points

x1,y1 =input("Enter the Coordinates of first point:").split(",")

x2,y2 =input("Enter the Coordinates of second point:").split(",")

#Printing the Distence between two points

z =((float(x2)-float(x1))\*\*2+(float(y2)-float(y1))\*\*2)\*\*0.5 print("Distance between two points is",z)

#difining string string="Python Programing"

#accessing 'i' print(string[-3])

#length of the string print(len(string))

#Print "Python"

print(string[0:6])

#prinnt "Program" print(string[7:-3])

#print "thing"

print((string[2:-13])+(string[-3:]))

#convert string to upper case print(string.upper)

#connect two string n="is intresting" print(string+n)

#check if string is upper case

print("Using 'isupper()'", string.isupper(), "as all the charecter are not uppercase")

#check if string is lowwer case

print("Using 'islower()'", string.islower(), "as all the charecter are not uppercase")

#Conver to swapcase

print("Using 'swapcase()'",string.swapcase())

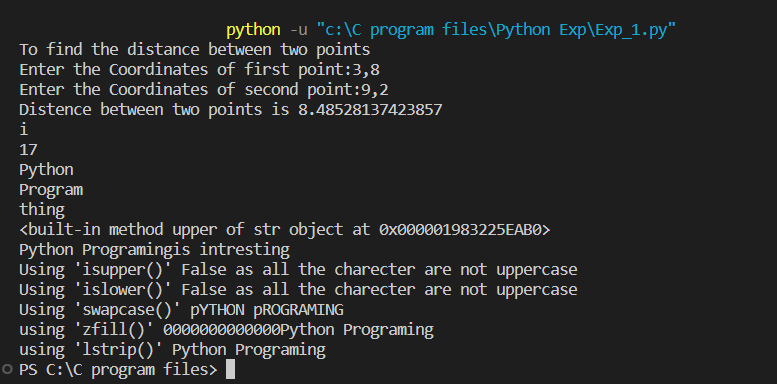
#use of zfill

print("using 'zfill()'",string.zfill(30))

#use of lstrip

print("using 'lstrip()'",string.lstrip())

**Output(s):**



# Conclusion:

**Experiment 1 is complete in doing that we understood the basics concept of Python Programing.**

# Post Lab Descriptive Questions:-

1. **How will you perform the following operations?**

# Obtain integer quotient and remainder while dividing 28 with 6.

* 1. **Obtain remainder on dividing 3.45 with 1.22**

# Obtain 4 from 3.5567.

* 1. **Print decimal equivalent of binary ‘1100001110’.**

# Obtain real part, imaginary part and conjugate of 4 + 2j.

**Implementation details of Q1:**

print("Obtain integer quotient and remainder while dividing 28 with 6.") num1=28

num2=6

print("The reminder is", num1%num2) print("The quotient is", num1//num2)

print("Obtain remainder on dividing 3.45 with 1.22") n1=3.45

n2=1.22

print("The remainder is",n1%n2)

print("Obtain 4 from 3.5567.") print(round(3.5567))

#First method to calculates the decimal equivalent print("Print decimal equivalent of binary '1100001110'.")

str=("1100001110")

n=int(str,2)

print("decimal conversion is",n)

print("Obtain real part, imaginary part and conjugate of 4 + 2j.") x = complex(4+2j)

print("The real part is:",x.real) print("The imaginary part is:",x.imag)

print ("Conjugated complex number of 4+2j is : ",x.conjugate() ) #Method 2 to calculates the decimal equivalent

# Function calculates the decimal equivalent # to given binary number

def binarytodecimal(binary): decimal, i = 0, 0

while(binary != 0): dec = binary % 10

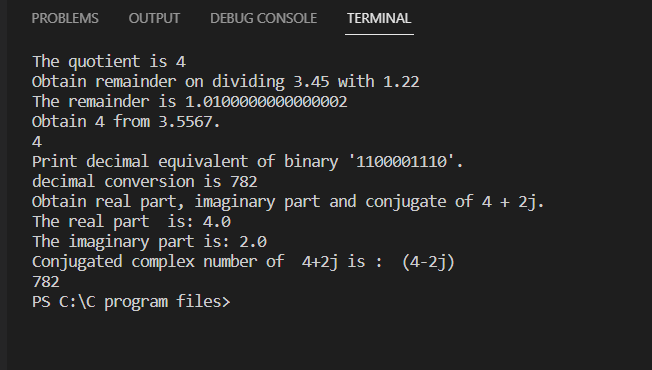
decimal = decimal + dec \*(2\*\*i) binary = binary//10

i += 1

print(decimal)

binarytodecimal(1100001110)

# Output of Q1:



1. **Which data type will you use to represent the following data values?**

# Number of days in a year Ans- integer data type

* 1. **The circumference of a circle Ans- Float data type**

# Distance between moon and earth Ans- Float data type

* 1. **Whether you will go for a trip?**

# Ans- Boolean data type

* 1. **Name of your favourite celebrity Ans- String data type**

# Books/ Journals/ Websites referred:

1. Reema Thareja, *Python Programming: Using Problem Solving Approach*, Oxford University Press, First Edition 2017, India
2. Sheetal Taneja and Naveen Kumar, *Python Programming: A modular Approach*, Pearson India, Second Edition 2018,India
3. https:/[/www](http://www.geeksforgeeks.org/python-strings/?ref=lbp).[geeksforgeeks.org/python-strings/?ref=lbp](http://www.geeksforgeeks.org/python-strings/?ref=lbp)

# Date: Signature of faculty in-charge