

| **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 that 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 strings 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 |
| [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 |
| [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 |
| [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 coordinates 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.

2) a) Create a variable and assign it the string “Python programming”

b) Access the character “i” from the variable by index and print it

c) Find the length of the string

d) Print the slice “Python” from the variable

e) Print the slice “program” from the variable

f) Get the string “thing” from the variable

g) Convert string into uppercase

h) Create another variable and assign it the string “is interesting”. Now concatenate both the strings

i) Apply different string methods given in the table

**Implementation details:**

**(1)**

#Assuming (x1,y1) and (x2,y2) to be the coordinates lying in the same plane.

#Taking the coordinates of the first point as input from the user

print("Enter the coordinates of the first point(x1,y1)")

x1=int(input("Enter x1= "))

y1=int(input("Enter y1= "))

print("Enter the coordinates of the second point(x2,y2)")

x2=int(input("Enter x2= "))

y2=int(input("Enter y2= "))

#Formula to calculate the distance between the 2 points

d=((x2-x1)\*\*2+(y2-y1)\*\*2)\*\*(1/2)

print("The distance between the two points is (in units): ",d

**(2)**

**#(a)Creating a variable 'a' and assigning the string to it**

**a="Python programming"**

**#(b)Using slice string operator to access the character “i” from the variable by its index number**

**print("The 15th character in the string is: ",a[15])**

**#(c)Using string function len() to find the length of the string**

**print("The length of the string is: ",len(a))**

**#(d)Using range slice string operator to print the slice “Python” from the variable**

**print(a[0:6])**

**#(e)Using range slice string operator to print the slice “program” from the variable**

**print(a[7:14])**

**#(f)Using a combination of range slice string operator and concatenation string operator to get the slice “thing” from the variable**

**print(a[2:4]+a[15:18])**

**#(g)Using string function upper() to convert all the characters in the string into uppercase**

**print("Converting the string into uppercase: ",a.upper())**

**#(h)Creating another variable 'b' and assigning another string to it**

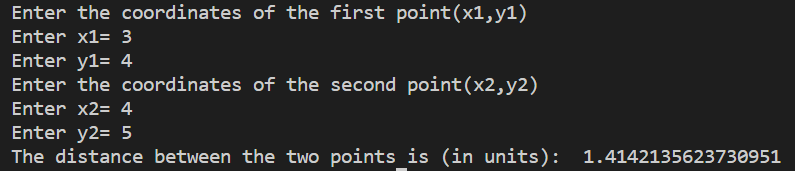
**b=" is interesting."**

**#Concatenating both the strings**

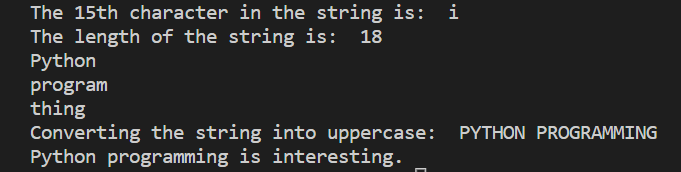
**print(a+b)**

**Output(s):**

**(1)**

****

**(2)**

****

**Conclusion:**

After the successful execution of the assignment, we learnt about basic variables,

mathematical functions, strings and almost all string functions in Python.

**Post Lab Descriptive Questions:-**

1. **How will you perform the following operations?**
   1. **Obtain integer quotient and remainder while dividing 28 with 6.**

**Ans-** We can obtain an integer quotient and remainder while dividing 28 with 6 using the floor division operator (28 // 6)

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

**Ans-** We can obtain the remainder on dividing 3.45 with 1.22 using the modulus operator (3.45 % 1.22)

* 1. **Obtain 4 from 3.5567.**

**Ans-** We can obtain 4 from 3.5567 by using the built-in round() function (round(3.5567))

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

**Ans-** We can use the built-in int() function to convert the binary string to its decimal equivalent, and then use the built-in print() function to output the result. In the int() function the second argument would be 2 to specify that the input string is in base 2 (ie its binary). The function would then return the decimal equivalent of the binary string which would then be printed as the output.

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

**Ans-** To obtain real part, imaginary part and conjugate of 4 + 2j, we can use the built-in functions ‘real’, ‘imag’ and ‘conjugate’ from the ‘cmath’ module. The ‘cmath’ module needs to be imported before using these built-in functions.

1. **Which data type will you use to represent the following data values?**
   1. **Number of days in a year**

**Ans-** Integer

* 1. **The circumference of a circle**

**Ans-** Float

* 1. **Distance between moon and earth**

**Ans-** Float

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

**Ans-** String

* 1. **Name of your favourite celebrity**

**Ans-** String

**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.geeksforgeeks.org/python-strings/?ref=lbp

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_ Signature of faculty in-charge**