

Batch: P1-2

Roll Number: 16014022050

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.

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	$a ** b = 10 \text{ to the power } 20$
//	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
<u>capitalize()</u>	Converts the first character of the string to a capital (uppercase) letter
<u>casefold()</u>	Implements caseless string matching
<u>center()</u>	Pad the string with the specified character.
<u>count()</u>	Returns the number of occurrences of a substring in the string.
<u>encode()</u>	Encodes strings with the specified encoded scheme
<u>endswith()</u>	Returns “True” if a string ends with the given suffix
<u>expandtabs()</u>	Specifies the amount of space to be substituted with the “\t” symbol in the string
<u>find()</u>	Returns the lowest index of the substring if it is found
<u>format()</u>	Formats the string for printing it to console
<u>format_map()</u>	Formats specified values in a string using a dictionary

Function Name	Description
<u>index()</u>	Returns the position of the first occurrence of a substring in a string
<u>isalnum()</u>	Checks whether all the characters in a given string is alphanumeric or not
<u>isalpha()</u>	Returns “True” if all characters in the string are alphabets
<u>isdecimal()</u>	Returns true if all characters in a string are decimal
<u>isdigit()</u>	Returns “True” if all characters in the string are digits
<u>isidentifier()</u>	Check whether a string is a valid identifier or not
<u>islower()</u>	Checks if all characters in the string are lowercase
<u>isnumeric()</u>	Returns “True” if all characters in the string are numeric characters
<u>isprintable()</u>	Returns “True” if all characters in the string are printable or the string is empty
<u>isspace()</u>	Returns “True” if all characters in the string are whitespace characters

Function Name	Description
<u>istitle()</u>	Returns “True” if the string is a title cased string
<u>isupper()</u>	Checks if all characters in the string are uppercase
<u>join()</u>	Returns a concatenated String
<u>ljust()</u>	Left aligns the string according to the width specified
<u>lower()</u>	Converts all uppercase characters in a string into lowercase
<u>lstrip()</u>	Returns the string with leading characters removed
<u>maketrans()</u>	Returns a translation table
<u>partition()</u>	Splits the string at the first occurrence of the separator
<u>replace()</u>	Replaces all occurrences of a substring with another substring
<u>rfind()</u>	Returns the highest index of the substring
<u>rindex()</u>	Returns the highest index of the substring inside the string
<u>rjust()</u>	Right aligns the string according to the width specified

Function Name	Description
<u>rpartition()</u>	Split the given string into three parts
<u>rsplit()</u>	Split the string from the right by the specified separator
<u>rstrip()</u>	Removes trailing characters
<u>splitlines()</u>	Split the lines at line boundaries
<u>startswith()</u>	Returns “True” if a string starts with the given prefix
<u>strip()</u>	Returns the string with both leading and trailing characters
<u>swapcase()</u>	Converts all uppercase characters to lowercase and vice versa
<u>title()</u>	Convert string to title case
<u>translate()</u>	Modify string according to given translation mappings
<u>upper()</u>	Converts all lowercase characters in a string into uppercase
<u>zfill()</u>	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.
2. Create a variable and assign it the string “Python programming” and then,
 - a. Access the character “i” from the variable by index and print it.
 - b. Find the length of the string.
 - c. Print the slice “Python” from the variable.
 - d. Print the slice “program” from the variable.
 - e. Get the string “thing” from the variable.
 - f. Convert string into uppercase.
 - g. Create another variable and assign it the string “is interesting” and concatenate both the strings.
 - h. Apply different string methods given in table.

Implementation details:

Problem 1:

```
print("ketaki mahajan / P1-2 / 16014022050")

#defining variables for input
x1 = int(input("x1= "))
y1 = int(input("y1= "))
x2 = int(input("x2= "))
y2 = int(input("y2= "))

#formula to calculate distance
result = ((y2-y1)**2 + (x2-x1)**2)**0.5

#printing output
print("The distance between the two points is = ", result)
```

Problem 2:

```
print("ketaki mahajan / P1-2 / 16014022050")

x = "python programming"

print(x[15]) #printing i
print(len(x)) #printing length of string
print(x[:6]) #printing Python
print(x[7:14]) #printing program
```



```
a = x[2:4] #th
b = x[15:] #ing
print(a+b) #concatenating both both strings a & b

print(x.upper()) #converting string x to uppercase

y = " is interesting"
print(x+y) #concatenating both both strings x & y
```

Output(s):

Problem 1:

```
PS C:\Users\HP\OneDrive\Desktop\python\python exp> & C:/Users/HP/AppData/Local/Microsoft/WindowsApps/python3.10.exe "c:/Users/HP/OneDrive/Desktop/python/python exp/exp1_1.py"
ketaki mahajan / P1-2 / 16014022050
x1= 9
y1= 4
x2= 7
y2= 3
The distance between the two points is = 2.23606797749979
PS C:\Users\HP\OneDrive\Desktop\python\python exp> █
```

Problem 2:

```
PS C:\Users\HP\OneDrive\Desktop\python\python exp> & C:/Users/HP/AppData/Local/Microsoft/WindowsApps/python3.10.exe "c:/Users/HP/OneDrive/Desktop/python/python exp/exp1_2.py"
ketaki mahajan / P1-2 / 16014022050
i
18
python
program
thing
PYTHON PROGRAMMING
python programming is interesting
PS C:\Users\HP\OneDrive\Desktop\python\python exp> █
```

Conclusion:

To conclude, carrying out experiment 1 has resulted in learning basic concepts of python such as different types of input/output functions, operations on strings and arithmetic operators.

Using these functions and operators, a program based on finding the distance between two coordinates and a program to apply different string functions were established.

Hence, experiment 1 was carried out successfully and the objectives that the problem defined, as seen in the output, were fulfilled.

Post Lab Descriptive Questions:

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

```
quotient = 28 // 6  
remainder = 28 % 6  
print(quotient, remainder)
```

- b. Obtain remainder on dividing 3.45 with 1.22 = **3.45%1.22**
 - c. Obtain 4 from 3.5567 = **print(round(3.5567))**
 - d. Print decimal equivalent of binary '1100001110'
= **print(int('1100001110', 2))**
 - e. Obtain real part, imaginary part and conjugate of $4 + 2j$
= **print(int(4+2j.real))**
print(4+2j.imag)
print(4+2j.conjugate())
2. Which data type will you use to represent the following data values?
 - a. Number of days in a year = **int data type (integer)**
 - b. The circumference of a circle = **float data type**
 - c. Distance between moon and earth = **float data type**
 - d. Whether you will go for a trip? = **boolean data type**
 - e. Name of your favourite celebrity = **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.geeksforgeeks.org/python-strings/?ref=lbp>

Date: _____

Signature of faculty in-charge