Daffodil International School
Value, Culture and Innovation

Input() function in python

Python provides us the facility to take input from user using input() function. This function prompts the users to input a value. For example,

name = input(“Enter your name: “)

print(name)

Output will be printed the name as you input.

name = input()

print(name)

Enter the name in the output screen and it will show the output of the given input.

number = input (“Enter a number: “)

print(number)

String variable stored here as number.

type() function will show the type of input given by the user.

Examples:

```
name = input(“Enter your name: “)
print(type(name))
number = input (“Enter  a number: “)
print(type(number))
number = int(input (“Enter  a number: “))
print(type(number))
```

```
Rahman
<class 'str'>
25
<class ‘str’>
25
<class ‘int’>
```

```
number = 5
print(type(number)
number = 5.5
print(type(number)
```

```
<class 'int'>
<class 'float'>
```

```
x = int(input("Enter 1st number: "))
y = int(input("Enter 2nd number: "))
z = x+y
print(z)
```

Find the output of the program using python IDE

and write the output here.

Important notes:

1) We use input() function to get user input.
2) Input() function always takes input in the string form
3) To convert string to an integer, we use int() function
4) To convert string to a float, we use float() function
5) We can only convert numeric strings to numbers.

Input a character:

```
ch = input("Enter a character: ")
print(ch)
```

The output can be character or string.

To input only one character the code should be:

```
ch = input("Enter a character: ")[0]
print(ch)
```

Evaluate Function (eval):

```
result = eval(input("Enter an expression: "))
print(result)
```

Input: 5+6-1

Output: 10

Concatenation example

name = “John”

print(“Hello, my name is” +name)

Practice python program using Input() function:

Program-1: To find out the cost one dozen of bananas.

```
P = int(input(“Enter the prince of one banana: “))
P = 12*P
Print (“Price of one dozen banana is: “, P)
```

Find the output of the program using python IDE

and write the output here.

Program-2: Converting hours to minutes.

```
H = int(input(“Enter the time in hours: “))
M = H*60
print (“Time in minutes is “, M)
```

Find the output of the program using python IDE

and write the output here.

Program-3: Find out the average age of two students.

```
A1 = int (input(“Enter the age of 1st student: “))
A2 = int (input(“Enter the age of 2nd student: “))
avg =  A1+A2/2
print (“The average age is: “, avg)
```

Find the output of the program using python IDE

and write the output here.

Program-4: Input the length and breadth and find out the perimeter.

```
L = int(input("Enter the length: "))

B = int(input("Enter the breadth: "))

P = 2 * (L+B)

print("The perimeter is ", P)
```

Find the output of the program using python IDE

and write the output here.

Python Challenges using input() of Python: (Home work)

1) Ananya went to buy a big cake and wanted to divide it in five friends. Input the weight of the cake in W. Calculate and print how much will each friend get.
2) Subarna Express had N number of coaches where the capacity of each coach was 80 passengers. Input the number of coaches. Calculate and print the total number of passengers travelling in a journey.
3) Accept the height and base of triangle and display it's area. [Area = 0.5 x height x base]
4) Input the temperature in Celsius and convert it to Farenheit. [F=1.8xC+32]
5) Input a number in N. Divide it by 10 and store the reminder in R. Print the result. [Use operator %]

Python Comments

A Python comment is **a line of text in a program that is not executed by the interpreter**. Comments are used during debugging to identify issues and to explain code. Comments start with a hash character (#).

```
# Program to take the user's input

name = input ("Enter your name: ")    # take input of the name

print("your name is:, name) # print the name variable
```

Python Multiline Comments:

Python doesn't have any other symbols for multiline comments except #. We can assign # sign for multi lines. But multiline strings are available in python.

```
test = ‘’’ I am
        multiline string ‘’’
print(test)
```

Why comments are so useful in python? Discussion.

Important note:

1) Comments are hints that we can add to our program to make the code easier to understand.
2) In python, comments start with the # symbol.
3) Comments can also be used for debugging code.
4) Python doesn't have multiline comments. However we can use multiline strings to act like multiline comments.

Operators in Python

Operators are the symbol that is used to perform operations on values variables.

name = “R.Rodrick”

Here equal to (=) sign is an operator.

Common arithmetic operators:

Addition	+
Subtraction	-
Multiplication	*
Exponent	**
Division	/
Floor Division	//
Remainder	%

Examples:

```
x =5
result = x+10
print(result)
```

output: 15

```
x = 5
quotient = x//2
remainder = x%2
print("Quotient is:", quotient)
print("Reminder is:", remainder)
```

Output: Quotient is: 2

Remainder is: 1

```
number = 34*5-9/3
number = (34*5)-(9/3)
print(number)
```

(+) operator can be used for concatenation.

```
str1 = "Hi"
str2 = "John"
print(str1+str2)
```

Assignment operators:

Assignment operators are used in Python to assign values to variables.

`a = 5` is a simple assignment operator that assigns the value 5 on the right to the variable `a` on the left.
There are various compound operators in Python like `a += 5` that adds to the variable and later assigns the same. It is equivalent to `a = a + 5`.

Examples:

x,y = 5,6

equivalent to

x = 5

y = 6

More assignment operators' examples:

x+=10 # x = x+5

x- = 10 # x = x-5

Program-1: Suppose you are a school student, and you need to pay taka 5300 tuition fee for the next month. The school is giving you a discount of 10% on the early payment of your tuition fee. Since it's a good offer, you decided to move on early payment. Can you find out how much money you have to pay?

Solution:

fee = 5300 or fee = float(input("Enter your fee: "))

discount_percent = 10

discount_amount = (discount_percent)/100*fee

discounted_fee = fee -discount_amount

print ("Fee after discount:", discounted_fee)

Program-2: Can you create a program to convert distant in kilometers to miles?

Hints: 1 mile = 0.621371 km

Solution:

kilometers = float(input("Enter kilometers to convert: "))

miles = kilometers * 0.621371

print ("The conversion of Kilometres to miles is: ", miles)

Important points to remember:

1) Equals operators (=) assigns the value in the right to the variable in the left.
2) Arithmetic operators perform basic arithmetic operations: such as addition, subtraction, division, multiplication etc.
3) If we use the + operator with string, it concatenates two strings.
4) To make out code more readable , we can use the parenthesis, for example: 34*(15-5)

Python Booleans:

In programming you often need to know if an expression is True or False.

You can evaluate any expression in Python, and get one of two answers, True or False.

When you compare two values, the expression is evaluated and Python returns the Boolean answer:

```
print(10 > 9)
print(10 == 9)
print(10 < 9)
```

x = "Hello"

y = 15

print(bool(x))

print(bool(y))

result1 = True

result2 = False

print(result1)

print(result2)

Now using comparison operators:

number = 5

print(number<10)

output: True

Python Comparison Operators:

Comparison operators are used to compare two values:

Operator	Name	Example
==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Examples:

Ex-1:

```
x = 5

y = 3

print(x == y)
```

Output: False

Ex-2:

```
x = 5

y = 3

print(x != y)
```

Output: True

Ex-3:

```
x = 5

y = 3

print(x > y)
```

Output: True

returns True because 5 is greater than 3

Ex-4:

```
x = 5

y = 3

print(x < y)
```

Output: False

returns False because 5 is not less than 3

Ex-5:

```
x = 5

y = 3

print(x >= y)
```

Output: True

returns True because five is greater, or equal, to 3

Python Logical Operators:

Logical operators are used to combine conditional statements:

and	True if both operands/statements are true
or	True if either of the operands/statements are true
not	True if the operands is false.

Examples:

Ex-1:

```
age = 22

gpa = 3.8

result = age>=18 and gpa>3.6

print(result)
```

output: True

Here both conditions are true.

Ex-2:

```
age = 16

gpa = 3.8

result = age>=18 or gpa<3.6

print(result)
```

output: False

Ex-3:

x = 5

print(x > 3 and x < 10)

Output: Ture

returns True because 5 is greater than 3 AND 5 is less than 10

Ex-4:

x = 5

print(x > 3 or x < 4)

Output: True

returns True because one of the conditions are true (5 is greater than 3, but 5 is not less than 4)

Ex-5:

x = 5

print(not(x > 3 and x < 10))

Output: False

returns False because not is used to reverse the result

Python challenge:

What is the output of the following program:

```
language = “Python”
print (“1.”, language == “Python”)
age= 18
print(“2.”, age>=18)
print(“3.”, age>18)
print(“4.”, age>=18 and language == “Java”)
```

Find the output of the program using python IDE

and write the output here.

Python Identity Operators:

Identity operators are used to compare the objects, not if they are equal, but if they are actually the same object, with the same memory location.

Examples:

```
x = ["apple", "banana"]
y = ["apple", "banana"]
z = x
print(x is z)
# returns True because z is the same object as x
print(x is y)
# returns False because x is not the same object as y, even if they have the same content
print(x == y)
# to demonstrate the difference betweeen "is" and "==": this comparison returns True because x is equal to y
```

Example: 2

```
x = ["apple", "banana"]
y = ["apple", "banana"]
z = x
print(x is not z)
# returns False because z is the same object as x
print(x is not y)
# returns True because x is not the same object as y, even if they have the same content
print(x != y)

# to demonstrate the difference betweeen "is not" and "!=": this comparison returns False because x is equal to y
```

Python Membership Operators:

Membership operators are used to test if a sequence is presented in an object.

Example: 1

```
x = ["apple", "banana"]
print("banana" in x)
# returns True because a sequence with the value "banana" is in the list
```

Example: 2

```
x = ["apple", "banana"]
print("pineapple" not in x)
# returns True because a sequence with the value "pineapple" is not in the list
```

Important Points to remember:

1) The Boolean represents one of two values: either true or false.
2) The comparison operators are used to compare two values.
3) If the comparison is right, the result is true. If not, the result is false.
4) The logical operators are used on Booleans.
5) There are three logical operators: and, or and not.