Programming Fundamentals Lab

Lab # 04

Operators in Python

Instructor: Engr. Muhammad Usman

Email: usman.rafiq@nu.edu.pk

Course Code: CL1002

Semester Fall 2021

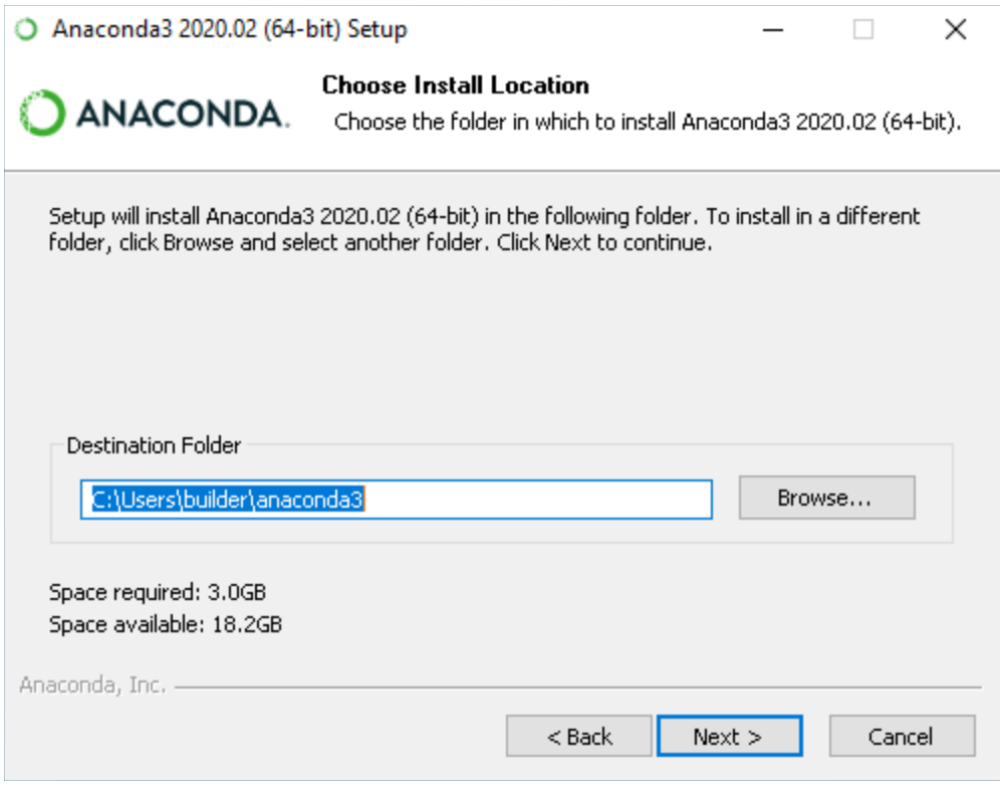
Department of Computer Science,

National University of Computer and Emerging Sciences FAST
Peshawar Campus

Contents

Installation of Anaconda

- Point your web browser at https://www.anaconda.com/download/ Download python latest version of the Windows.
- Double click the installer to launch.
- Select a destination folder to install Anaconda and click the Next button.



For more details regrading Anaconda Installation.

https://docs.anaconda.com/anaconda/install/windows/

Operators:

Operator is a symbol which is used to perform some operation. Operators are used to perform operations on variables and values.

In other words

An operator is a symbol that will perform mathematical operations on variables or on values. Operators operate on operands (values) and return a result.

Python has 7 types of operators.

1. Arithmetic Operators
2. Relational Operators
3. Assignment Operators
4. Logical Operators
5. Membership Operators
6. Identity Operators
7. Bitwise Operators

Already discussed Arithmetic, Relational, Assignment and Logical Operators in the previous Lab.

5. Membership Operators

Membership operators check whether a value is in another. Python has 2 membership operators:

1. in
2. not in

```
In [40]:
2 in [1,2,3]
Out[40]: True

In [22]:
2 in [4,5,6]
Out[22]: False

In [25]:
2 not in [1,2,3]
Out[25]: False

In [41]:
2 not in [4,5,6]
Out[41]: True
```

6. Python Identity Operators

Identity operators check whether two values are identical. Python has 2 identity operators as well:

1. is
2. is not

In [47]:
```
3 is 3
```

```
<>:1: SyntaxWarning: "is" with a literal. Did you mean "=="?
<>:1: SyntaxWarning: "is" with a literal. Did you mean "=="?
<ipython-input-47-1ee75a7b8cc8>:1: SyntaxWarning: "is" with a literal. Did you mean "=="?
  3 is 3
```

Out[47]: True

In [48]:
```
3 is 3.0
```

```
<>:1: SyntaxWarning: "is" with a literal. Did you mean "=="?
<>:1: SyntaxWarning: "is" with a literal. Did you mean "=="?
<ipython-input-48-7386e7c10e85>:1: SyntaxWarning: "is" with a literal. Did you mean "=="?
  3 is 3.0
```

Out[48]: False

In [44]:
```
2 is not 2.0
```

```
<>:1: SyntaxWarning: "is not" with a literal. Did you mean "!="?
<>:1: SyntaxWarning: "is not" with a literal. Did you mean "!="?
<ipython-input-44-0c3d7bef06bc>:1: SyntaxWarning: "is not" with a literal. Did you mean "!="?
  2 is not 2.0
```

Out[44]: True

In [30]:
```
a=3
b=a
a is b
```

Out[30]: True

7. Bitwise Operators

They operate on values bit by bit.

```
In [49]: 3&4
Out[49]: 0

In [33]: 3|4
Out[33]: 7

In [36]: 3^4
Out[36]: 7

In [37]: 6^4
Out[37]: 2
```

Function

Function is a group of related statements that perform a specific task.

Functions help break our program into smaller and modular chunks. As our program grows larger and larger, functions make it more organized and manageable.

it avoids repetition and makes code reusable

Types of function

Two types of function

1. Built-in functions
2. User define functions

Built-in Functions

The Python interpreter has a number of functions that are always available for use. These functions are called built-in functions. Few of them are listed here.

max(*arg1*, *arg2*, **args*[, *key*])

Return the largest item in an iterable or the largest of two or more arguments.

min(*arg1*, *arg2*, **args*[, *key*])
Return the smallest item in an iterable or the smallest of two or more arguments.

pow(*base*,*exp*[,*mod*])

Return *base* to the power *exp*; if *mod* is present, return *base* to the power *exp*, modulo *mod*

int([X])

Return an integer object constructed from a number or string *x*, or return `0` if no arguments are given.

float([X])

Return a floating point number constructed from a number or string *x*.

For more details regarding built-in functions check the link given below.:
https://docs.python.org/3.9/library/functions.html

User-define Function:

You can also create your own functions. These functions are called user-defined functions.

Input

There are two functions in Python that you can use to read data from the user:

- raw_input()
- input()

You can store the results from them into a variable.

Raw_Input() [not in python 3]

Input():

The input() function takes input from the user and returns it.

```
In [2]:
age = input("What is your age  ")
print("Your age is ",age)

What is your age  25
Your age is  25
```

Task:

Note: You have to submit .py file for each question e-g q1.py, q2.py etc (use the same naming scheme)

1. Write a program that takes the length, width and height of a parallelepiped from user and finds its volume. The formula to calculate the volume is given below.

   Volume = length*width*height

   Your program should work as shown below.

```
Enter the Length of parallelepiped  2
Enter the width of parallelepiped  3
Enter the height of parallelepiped  4
The Volume of parallelepiped is  24
```

2. Write a program that takes the radius of a sphere from user and finds its area. The formula to calculate the area is given below.

   Area = 4 * Pi * r^2

   Your program should work as shown below

```
Enter the radius of sphere  2
The Area of sphere is  50.26548245743669
```