
Security and Networking with Python (CSE 2157)

MINOR ASSIGNMENT-1: BASIC ELEMENTS OF PYTHON PROGRAMMING

Introduction to Python Programming.

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – One can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – One can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features listed below-

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Visit the official page for python <https://www.python.org/downloads/> on the Windows operating system.

-
1. Evaluate the following expressions:
(x<y) or (not(z==y) and (z<x))
 - a. x =0, y=6, z=10
 - b. x=1, y=1, z=1
 2. Evaluate the following expressions involving arithmetic operators:
 - a. -7*20+8/16*2+54
 - b. 7**2//9%3
 - c. (7-4*2)*10-25*8//5
 - d. 5%10+10-25*8//5
 - e. 'hello'*2-5
 3. Evaluate the following expressions involving relational and logical operators:
 - a. 'hi' > 'hello' and 'bye' < 'Bye'
 - b. 'hi' > 'hello' or 'bye' < 'Bye'
 - c. 7 > 8 or 5 < 6 and 'I am fine' > 'I am not fine'
 - d. 10 !=9 and 29 >= 29
 - e. 10 !=9 and 29 >= 29 and 'hi' > 'hello' or 'bye' < 'Bye' and 7 <= 2.5
 4. Evaluate the following expressions involving arithmetic, relational and logical operators:
 - a. 5 % 10 + 10 < 50 and 29 >= 29
 - b. 7 ** 2 <= 5 // 9 % 3 or 'bye' < 'Bye'
 - c. 5 % 10 < 8 and -25 > 1 * 8 // 5
 - d. 7 ** 2 // 4 + 5 > 8 or 5 != 6
 - e. 7/4 < 6 and 'I am fine' > 'I am not fine'
 - f. 10 + 6 * 2 ** 2 != 9//4-3 and 29 >= 29/9
 - g. 'hello' * 5 > 'hello' or 'bye' < 'Bye'
 5. Evaluate the following expressions involving bitwise operators:
 - a. 15 & 22
 - b. 15 |22
 - c. -15 & 22
 - d. -15 |22
 - e. ~ 15
 - f. ~ 22
 - g. ~ -20
 - h. 15^22
 - i. 8 << 3
 - j. 40 >> 3
 6. Differentiate between the following operators with the help of examples:
 - a. = and ==
 - b. / and %
 - c. / and //
 - d. * and **

-
7. What output will be displayed when the following commands are executed in Python shell in sequence:
- ```
>>> a = 6
>>> a == 6
>>> a < 5.9
>>> a > 5.9
```
  - ```
>>> b = 7
>>> b / 6
>>> b // 6
>>> b / 4
>>> b % 4
>>> b % 7
>>> b * 2
>>> b ** 2
```
8. Construct logical expressions for representing the following conditions:
- marks scored should be greater than 300 and less than 400.
 - Whether the value of grade is an uppercase letter.
 - The post is engineer and experience is more than four years.
9. Write Python statements for the following equations:
- $\text{root1} = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$
 - $\text{result} = \frac{2xy - 9y}{2xy^3} - \frac{4yx^2}{2y}$
 - $\text{result} = 2 \cos_2^{-1}(x + y) \cos_2^{-1}(x - y) + e^x - 1 - \frac{x}{4} + \tan x - \log(v)$
10. How does the effect of the following two statements differ?
- $x += x + 10$
 - $x = x + 10$
11. Write a program that asks the user to enter the width and length of a room. Once these values have been read, your program should compute and display the area of the room. The length and the width will be entered as floating-point numbers. Include units in your prompt and output message; either feet or meters, depending on which unit you are more comfortable working with.
12. An online retailer sells two products: widgets and gizmos. Each widget weighs 75 grams. Each gizmo weighs 112 grams. Write a program that reads the number of widgets and the number of gizmos from the user. Then your program should compute and display the total weight of the parts.
13. Write a program that determines how quickly an object is travelling when it hits the ground. The user will enter the height from which the object is dropped in meters (m). Because the object is dropped its initial speed is 0 m/s. Assume that the acceleration due to gravity is 9.8 m/s^2 . You can use the formula $v_f = \sqrt{v_i^2 + 2ad}$ to compute the final speed, v_f , when the initial speed, v_i , acceleration, a , and distance, d , are known.
14. Write a program that reads a four-digit integer from the user and displays the sum of its digits. For example, if the user enters 3141 then your program should display $3 + 1 + 4 + 1 = 9$.
15. Write a program that reads three integers from the user and displays them in sorted order (from smallest to largest). Use the min and max functions to find the smallest and largest values. The middle

value can be found by computing the sum of all three values, and then subtracting the minimum value and the maximum value.

16. Create a program that reads duration from the user as a number of days, hours, minutes, and seconds. Compute and display the total number of seconds represented by this duration.