Introduction to Python

COMP 8347

Slides prepared by Dr. Arunita Jaekel

arunita@uwindsor.ca

Python Basics

- Topics
  - Introduction
  - Data Types
  - Arithmetic/Logic Operations

Notable Features

- Many basic data types : numbers (floating point, complex, and unlimited-length long integers), strings, lists, and dictionaries.
- Supports object-oriented programming: with classes and multiple inheritance.
- Supports raising and catching exceptions: cleaner error handling.
- Strongly and dynamically typed data types: mixing incompatible types (e.g. attempting to add a string and a number) causes an exception, errors caught sooner.
- Automatic memory management: frees you from having to manually allocate and free memory in your code.

Notable Feautures

- Elegant syntax: programs easier to read.
- Easy-to-use language: ideal for prototype development.
- Large standard library: supports many common programming tasks e.g. connecting to web servers, regular expressions, file I/O.
- A bundled development environment called IDLE.
- Runs on different computers and operating systems: Windows, MacOS, many brands of Unix, OS/2, ...
- Free software: Free to download or use Python - the language is copyrighted it's available under an open source license.

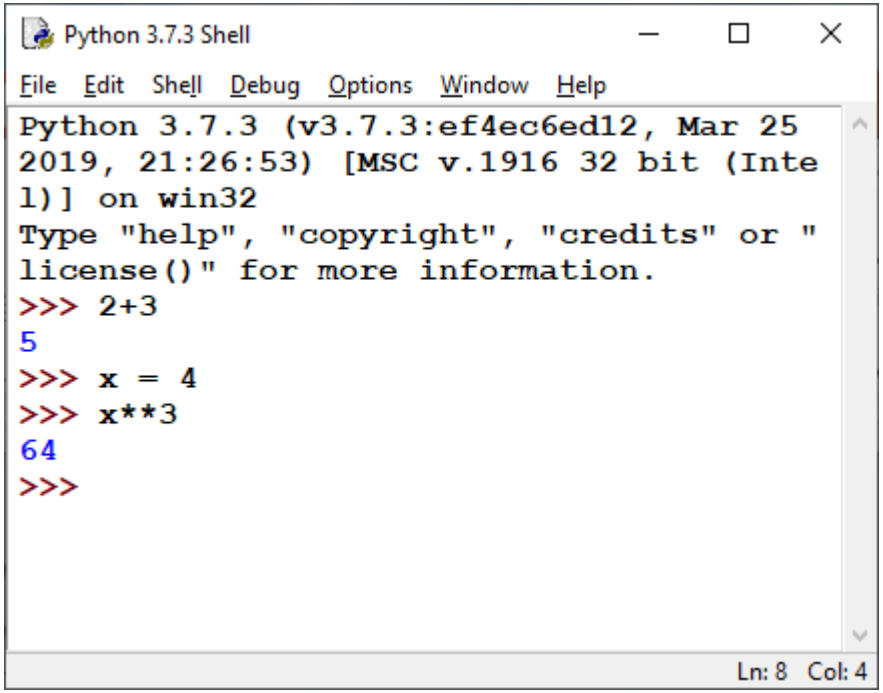
Indentation

- Python does not use brackets to structure code, instead it uses whitespaces
  - Tabs are not permitted.
    - Four spaces are required to create a new block,
  - To end a block simply move the cursor four positions left.
  - An example:

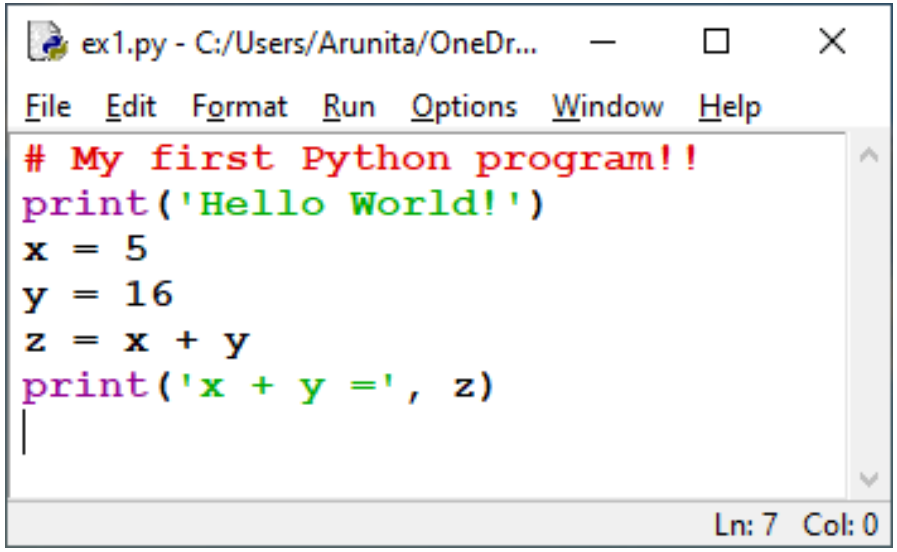
```
1. def bar(x):
2.     if x == 0:
3.         foo()
4.     else:
5.         foobar(x)
```

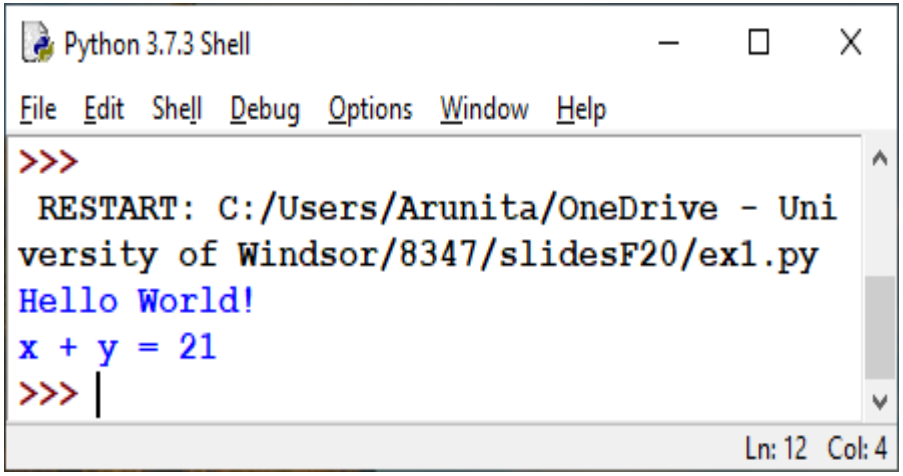
IDLE

- IDLE: Basic IDE that comes with Python
  - Should be available from Start Menu under Python program group.
  - Main "Interpreter" window.
    - Allows us to enter commands directly into Python
    - As soon as we enter in a command Python will execute it display the result.
    - '>>>' signs act as a prompt.



IDLE EXAMPLE





Numeric Data Types

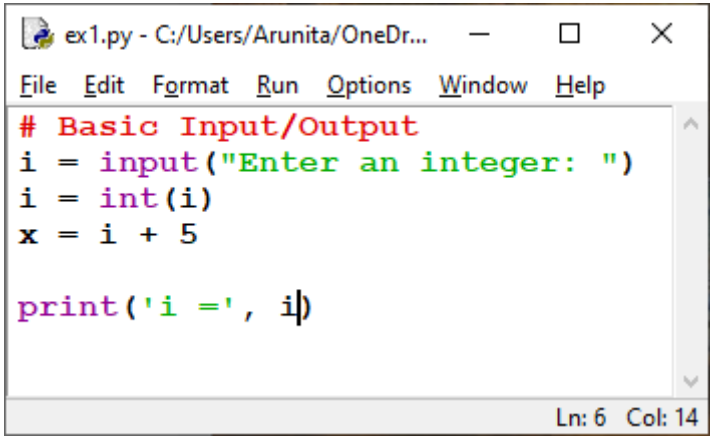
- *int*: represents positive and negative whole numbers.
  - Written without a decimal point
  - e.g. 5, 258964785663
- *float*: written with a decimal point
  - e.g. 3.0, 5.8421, 0.0, -32.5 etc

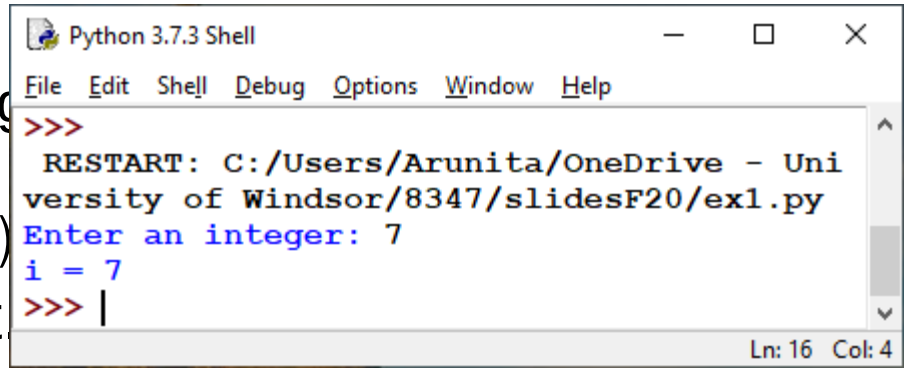
Arithmetic Operators

- Basic arithmetic operators: + (addition), - (subtraction), * (multiplication), / (division)
  - / (division) produces floating point value — 15/3 → 5.0
  - // (integer division) truncates any fractional part — 25//3 → 8
  - % (remainder) gives the remainder after integer division. — 25%3 → 1
  - Augmented assignment operators: +=, -=, *=, /=

Basic Input/Output

- Built-in input() function accepts input from user.
  - takes optional string argument to print on console
  - waits for user to type response and hit Enter
  - If no text, user just hits enter: return empty string
  - otherwise, return string containing entered text
  - Example: i = input(“Enter an integer: ”)
- Built-in print() function for output
  - Example: print(“int = ”, i)





Summary

- Python intro
- Numeric data types
- Input/Output