

Introduction to Python

COMP 8347

Slides prepared by Dr. Arunita Jaekel

arunita@uwindsor.ca

Revised by: Dr. Saja Al-Mamoori



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 Features

- **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**.



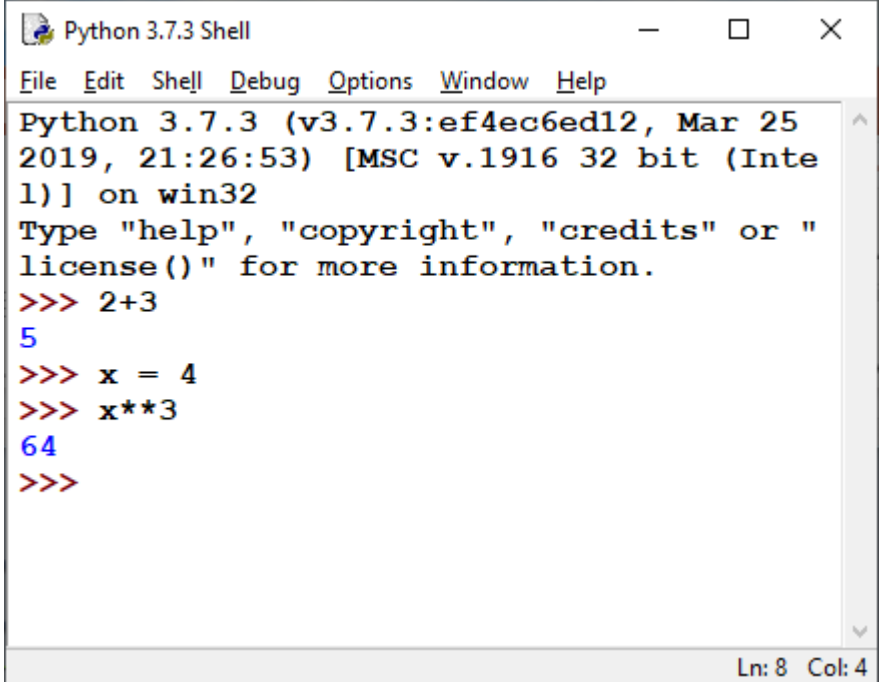
Indentation

- 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)`



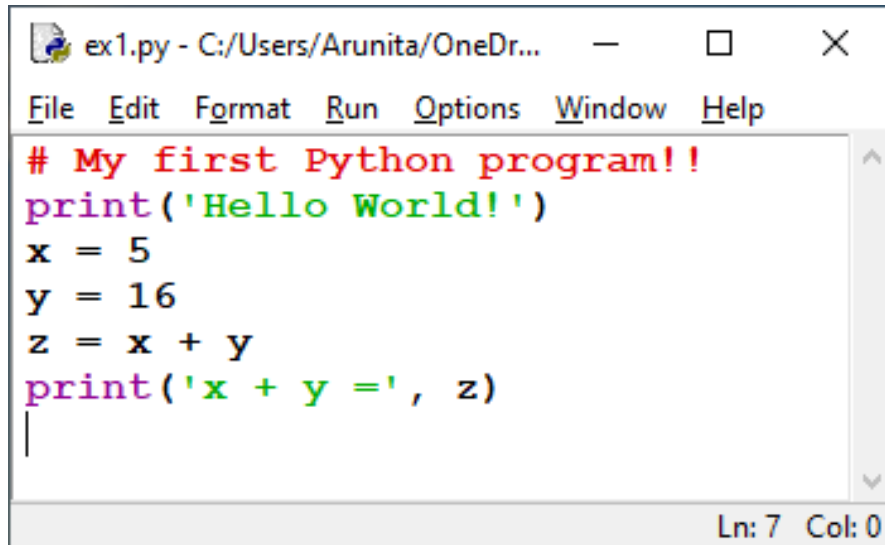
IDLE

- **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.



```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25
2019, 21:26:53) [MSC v.1916 32 bit (Inte
l)] on win32
Type "help", "copyright", "credits" or "
license()" for more information.
>>> 2+3
5
>>> x = 4
>>> x**3
64
>>>
```

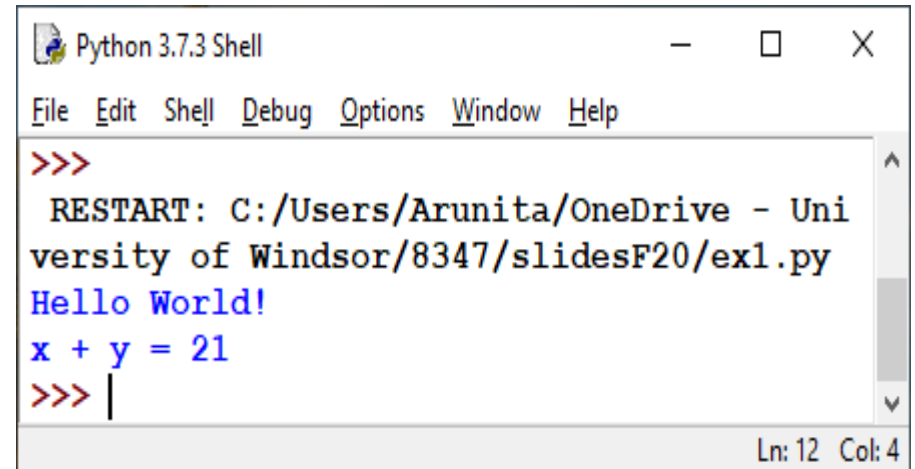
IDLE EXAMPLE



The screenshot shows the IDLE Python editor window. The title bar reads "ex1.py - C:/Users/Arunita/OneDr...". The menu bar includes "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code editor contains the following Python code:

```
# My first Python program!!  
print('Hello World!')  
x = 5  
y = 16  
z = x + y  
print('x + y =', z)  
|
```

The status bar at the bottom right indicates "Ln: 7 Col: 0".



The screenshot shows the Python 3.7.3 Shell window. The title bar reads "Python 3.7.3 Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The shell displays the output of the script execution:

```
>>>  
RESTART: C:/Users/Arunita/OneDrive - Uni  
versity of Windsor/8347/slidesF20/ex1.py  
Hello World!  
x + y = 21  
>>> |
```

The status bar at the bottom right indicates "Ln: 12 Col: 4".



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



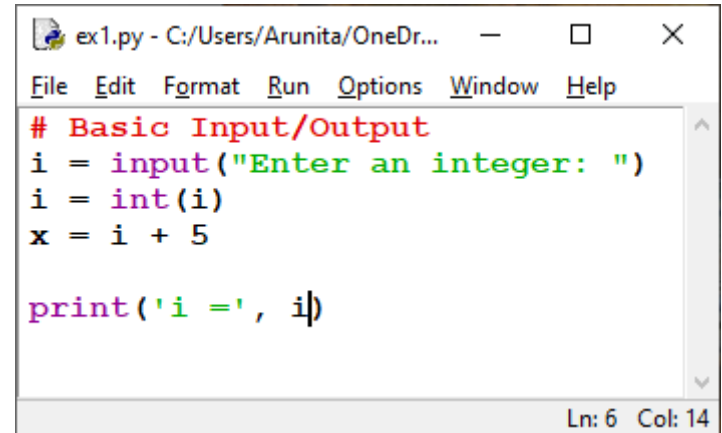
Arithmetic Operators

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



Basic Input/Output

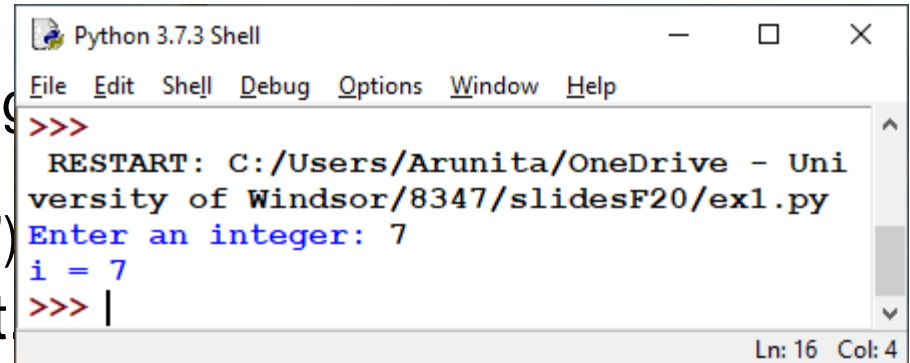
- Built-in input() function accepts input from user.
 - takes optional string argument to print on console
 - waits for user to type response and hit Enter
 - If no text, user just hits enter: return empty string
 - otherwise, return string containing entered text
 - Example: `i = input("Enter an integer: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



```
ex1.py - C:/Users/Arunita/OneDr...
File Edit Format Run Options Window Help
# Basic Input/Output
i = input("Enter an integer: ")
i = int(i)
x = i + 5

print('i =', i)
```

Ln: 6 Col: 14



```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
>>>
RESTART: C:/Users/Arunita/OneDrive - University of Windsor/8347/slidesF20/ex1.py
Enter an integer: 7
i = 7
>>> |
```

Ln: 16 Col: 4

Summary

- Python intro
- Numeric data types
- Input/Output

