

DATA 301

Introduction to Data Analytics

Python

Dr. Firas Moosvi
University of British Columbia Okanagan

Slides courtesy of **Dr. Ramon Lawrence**

Why learn Python?

Python is increasingly the most popular choice of programming language for data analysts because it is designed to be simple, efficient, and easy to read and write.

There are many open source software and libraries that use Python and data analysis tools built on them.

We will use Python to learn programming and explore fundamental programming concepts of commands, variables, decisions, repetition, and events.



What is Python?

Python is a general, high-level programming language designed for code readability and simplicity.

Python is available for free as open source and has a large community supporting its development and associated tools.

Python was developed by Guido van Rossum and first released in 1991. Python 2.0 was released in 2000 (latest version 2.7), and a backwards-incompatible release Python 3 was in 2008.

- Our coding style will be Python 3 but most code will also work for Python 2.
- Name does refer to Monty Python.

Python Language Characteristics

Python supports:

- dynamic typing – types can change at run-time
- multi-paradigm – supports procedural, object-oriented, functional styles
- auto-memory management and garbage collection
- extendable – small core language that is easily extendable

Python core philosophies (by Tim Peters: <https://www.python.org/dev/peps/pep-0020/>)

- Beautiful is better than ugly
- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Readability counts

Some Quotes

"If you can't write it down in English, you can't code it."

-- Peter Halpern

"If you lie to the computer, it will get you."

-- Peter Farrar

Introduction to Programming

An *algorithm* is a precise sequence of steps to produce a result. A *program* is an encoding of an algorithm in a *language* to solve a particular problem.

There are numerous languages that programmers can use to specify instructions. Each language has its different features, benefits, and usefulness.

The goal is to understand fundamental programming concepts that apply to all languages.

Python: Basic Rules

To program in Python you must follow a set of rules for specifying your commands. This set of rules is called a **syntax**.

- Just like any other language, there are rules that you must follow if you are to communicate correctly and precisely.

Important general rules of Python syntax:

- Python is **case-sensitive**.
- Python is particular on whitespace and indentation.
- The end of command is the end of line. Semi-colon is not a required terminator.
- Use four spaces for indentation whenever in a block.

```
def spam():  
    eggs = 12  
    return eggs  
print spam()
```

Comments

Comments are used by the programmer to document and explain the code. Comments are ignored by the computer. Two types:

- 1) One line comment: put “#” before the comment and any characters to the end of line are ignored by the computer.
- 2) Multiple line comment: put “""" at the start of the comment and """ at the end of the comment. The computer ignores everything between the start and end comment indicators.

Example: `# Single line comment`
`print (1) # Comment at end of line`
`""" This is a`
`multiple line`
`comment """`

Python Programming

A Python program, like a book, is read left to right and top to bottom.

Each command is on its own line.

```
# Sample Python program
name = "Joe"
print("Hello")
print("Name: "+name)
```

Flow of Execution

Start at first statement at top and proceed down executing each statement

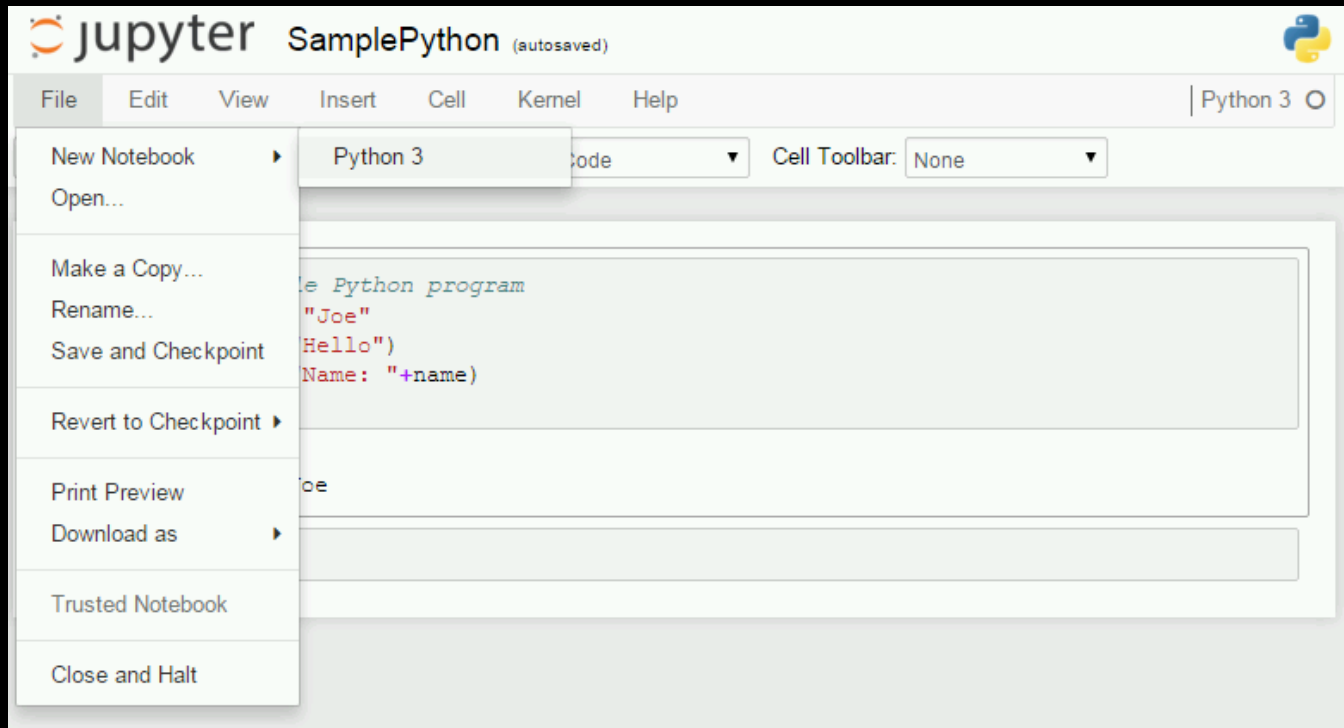


A user types in a Python program in a text editor or development environment and then runs the program.

Python Editor - jupyter

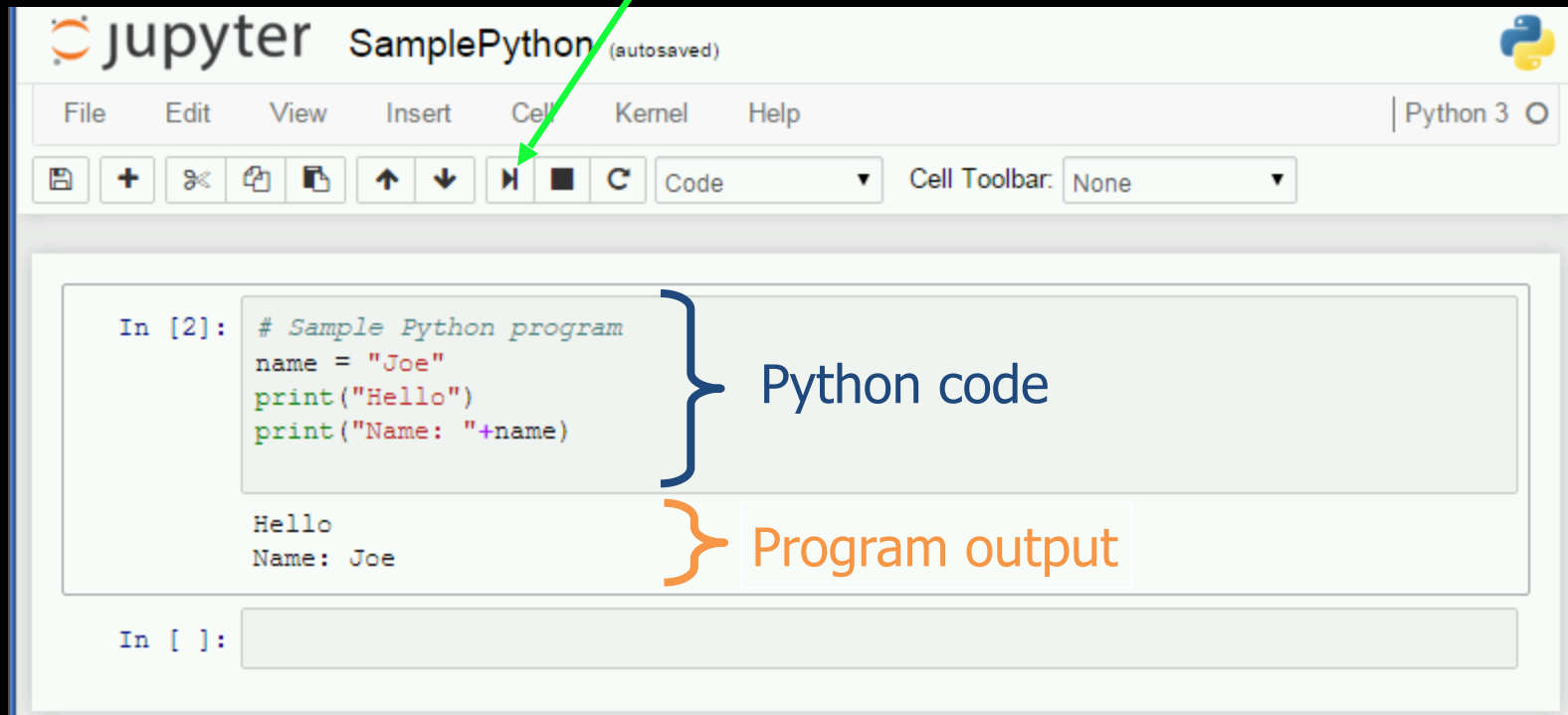
jupyter is a graphical, browser-based editor for Python.

To create a new notebook, select File, New Notebook, Python3.



Python Editor – jupyter notebook

Button to run code (shortcut is Ctrl+Enter)



The screenshot displays the Jupyter Notebook interface for a file named "SamplePython (autosaved)". The top menu bar includes "File", "Edit", "View", "Insert", "Cell", "Kernel", and "Help". The "Cell" menu is currently open, showing options like "Run", "Stop", and "Restart". A green arrow points to the "Run" button (a play icon) in the toolbar. Below the toolbar, a code cell is shown with the following Python code:

```
In [2]: # Sample Python program
name = "Joe"
print("Hello")
print("Name: "+name)
```

The output of the code is displayed below the code cell:

```
Hello
Name: Joe
```

A blue bracket on the right side of the code cell is labeled "Python code". An orange bracket on the right side of the output is labeled "Program output". Below the code cell, there is an empty input field labeled "In []:".

Python: Hello World!

Simplest program:

```
print("Hello World!")
```

The `print` function will print to the terminal (standard output) whatever data (number, string, variable) it is given.

Try it: Python Printing

Question 1: Write a Python program that prints "I am fantastic!".

Question 2: Write a Python program that prints these three lines:

```
I know that I can program in Python.
```

```
I am programming right now.
```

```
My awesome program has three lines!
```

Python Question

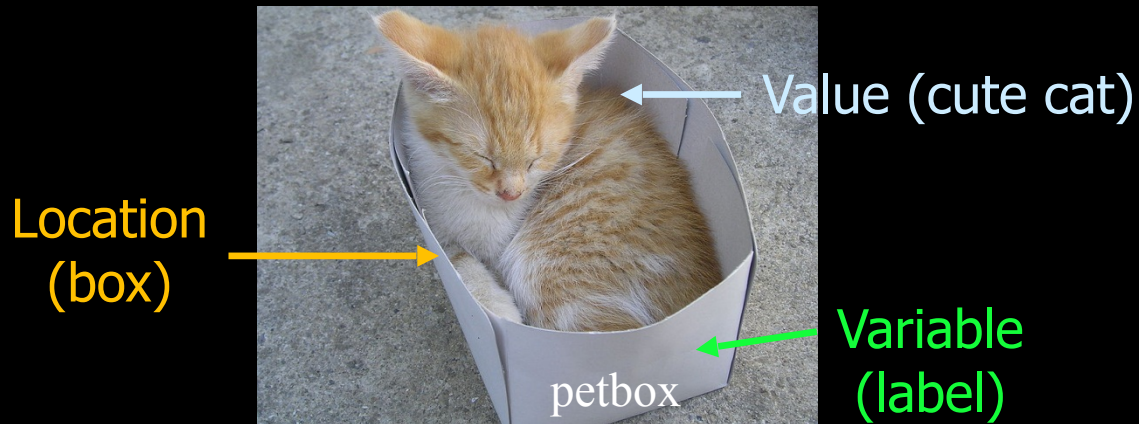
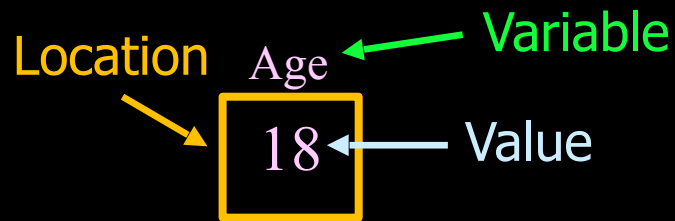
Question: How many of the following statements are **TRUE**?

- 1) Python is case-sensitive.
- 2) A command in Python must be terminated by a semi-colon.
- 3) Indentation does not matter in Python.
- 4) A single line comment starts with " " " " .
- 5) The `print` command prints to standard input.

A) 0 **B) 1** **C) 2** **D) 3** **E) 4**

Variables

A *variable* is a name that refers to a location that stores a data value.



IMPORTANT: The *value* at a location can change using initialization or assignment.

Variable Assignment

Assignment using an `=` sets the value of a variable.

Example:

```
num = 10
```

```
message = "Hello world!"
```

num

10

message

Hello world!

Python Variables

To create a variable in Python, you must only provide a name.

- A variable type is dynamic. It can store numbers, strings, or Boolean at any time.

Example:

```
val = 5
```

```
val = "Hello"
```

Boolean values can be either `True` or `False`. Note case matters.

```
isAwesome = True
```

```
isAwesome = False
```