Chapter 1
Introduction to Computers, Programs, and Python
Part - 1

What to be covered next

- What is Python?
- The history of python
- Getting started with python

What is Python?

General Purpose Interpreted Object-Oriented

- Python is a **general purpose** programming language. That means you can use Python to write code for any programming tasks.
- Python are now used in Google search engine, in mission critical projects in NASA, in processing financial transactions at New York Stock Exchange.

What is Python?

General Purpose **Interpreted** Object-Oriented

Python is interpreted, which means that python code is translated and executed by an interpreter *one statement at a time*.

```
>>> print("Hello, world“)
Hello, world
>>> print(2+3)
5
>>> print("2+3=", 2+3)
2+3= 5
>>>
```

What is Python?

General Purpose Interpreted Object-Oriented

- Python is an object-oriented programming language.
- Object-oriented programming is a powerful tool for developing reusable software.

The History of Python

- Created by Guido van Rossum in Netherlands in 1990
- Open source
- Python 2 vs. Python 3

Questions:

1. What is ABC?
2. How long did Guido spend to create Python?
3. Is Python programming language named after the snake python?

Getting Started with Python

Step 1: Google Python → download → Python 3.9.1 → install → using IDLE (Python GUI)

https://www.python.org/

When you start Python, you will see something like:

```
Python 3.9.1 (v3.9.1:e09359112e, Jul  8 2019, 14:54:52)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

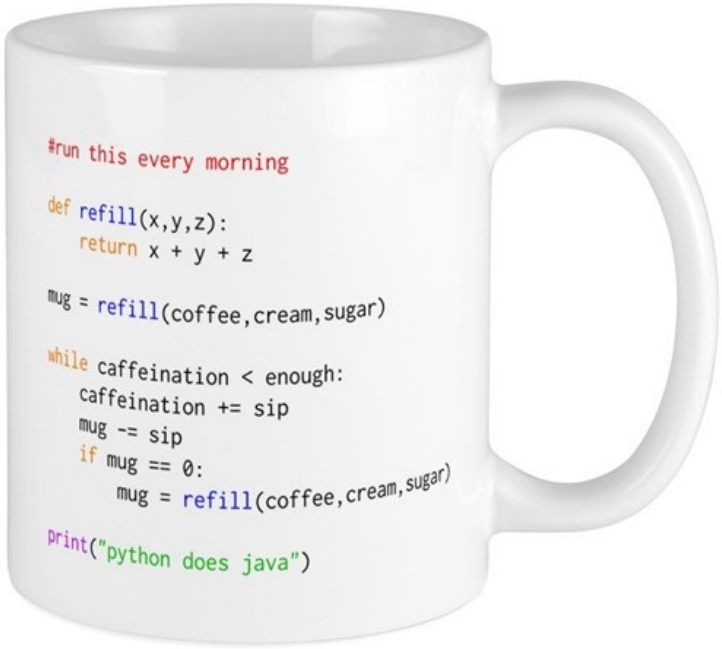
Note: The ">>>" is a Python prompt indicating that Python is ready for us to give it a command. These commands are called statements.

Step 2: Run the commands in red below one by one. Remember to press Enter key at the end of each command.

```
>>> print("Hello, world“)
Hello, world
>>> print("2+3=?")
2+3= ?
>>> 2+3
5
>>> type(2)
<class ‘int’>
>>> type(’Hello, world’)
<class ‘str’>
```

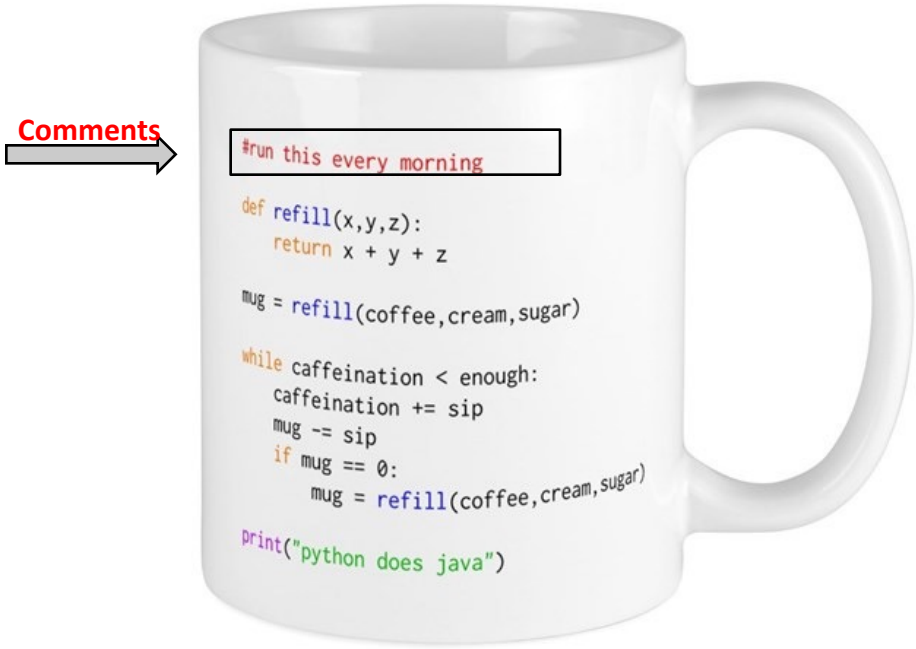
Anatomy of a Python Program

- Statements
- Comments
- Indentation



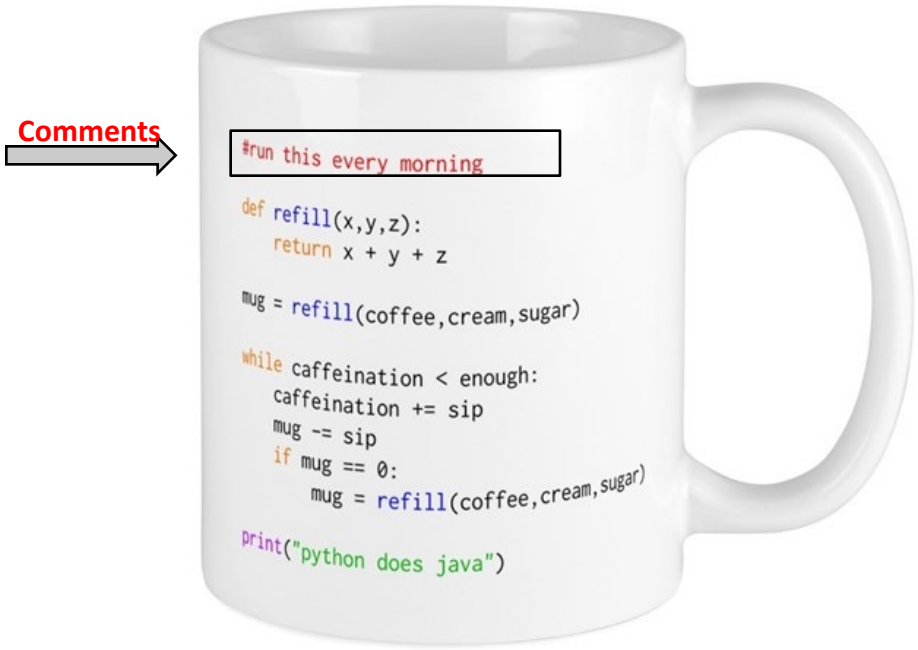
Comments

- Anything after a # is ignored by Python
- Why comment?
  - Describe what is going to happen in a sequence of code
  - Document who wrote the code or other ancillary information
  - Turn off a line of code (perhaps temporarily)



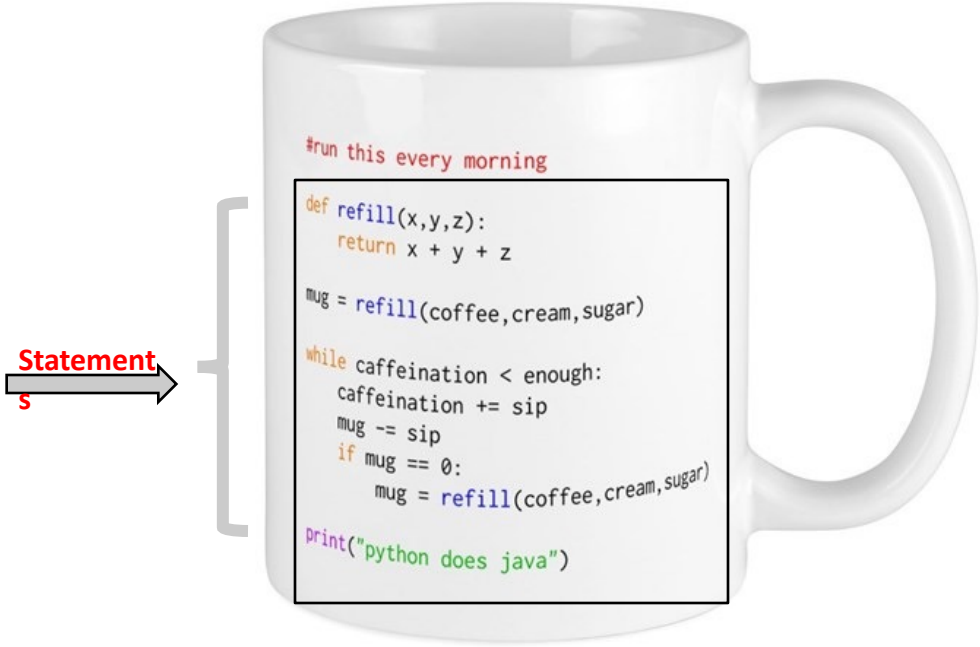
Comments

- Anything after a # is ignored by Python
- Why comment?
  - Describe what is going to happen in a sequence of code
  - Document who wrote the code or other ancillary information
  - Turn off a line of code (perhaps temporarily)



Statement

- A statement represents an action or a sequence of actions.
  - print(“Python does java") is a statement to display the greeting "Welcome to Python“.



Indentation

- The indentation matters in Python.
- Commonly statements are entered from the first column in the new line.

Indentation

Wrong indentation!
Will cause an error.

Indentation

```
# Display two messages
print("Welcome to Python")
print("Python is fun")
```

Run a Python Script

Step 3: File -> New File;

Step 4: Type the three lines below in the text editor;

Step 5: File -> Save as “test”

Step 6: Run -> run module

Please check what will be printed.

```
print("Hello, world“)
print(2+3)
print("2+3=", 2+3)
```

Two More Simple Examples

Step 7: Create a new Python program “ComputeExp” and enter the commands below.

Please check what will be printed.

```
# Display three messages
print("Welcome to Python")
print("Python is fun")
print("Problem Driven") # Display Problem Driven
```

Step 8: Create a new Python program “Welcome” and enter the commands below.

Please check what will be printed.

```
# Compute expression
print("(10.5 + 2 * 3) / (45 - 3.5) = ")
print((10.5 + 2 * 3) / (45 - 3.5)) # Display the result of the expression
```

Programming Style and Documentation

- Syntax Errors
  Can not RUN / Not executable
  - Error in code construction
- Runtime Errors
  Able to run, but quits while running
  - Causes the program to abort
- Logic Errors
  Able to run, but results not expected
  - Produces incorrect result

```
>>> print("hello)

SyntaxError: EOL while scanning string literal
>>> prrint("hello")
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    prrint("hello")
NameError: name 'prrint' is not defined
>>> print(1/0)
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    print(1/0)
ZeroDivisionError: division by zero
>>> print(5 / 9 * 35 - 32)
-12.555555555555554
>>>
```