Program Execution

Table of Contents

1 Introduction

How was the pre-class content? Did you cover the Program Execution?

Students, drag the icon!

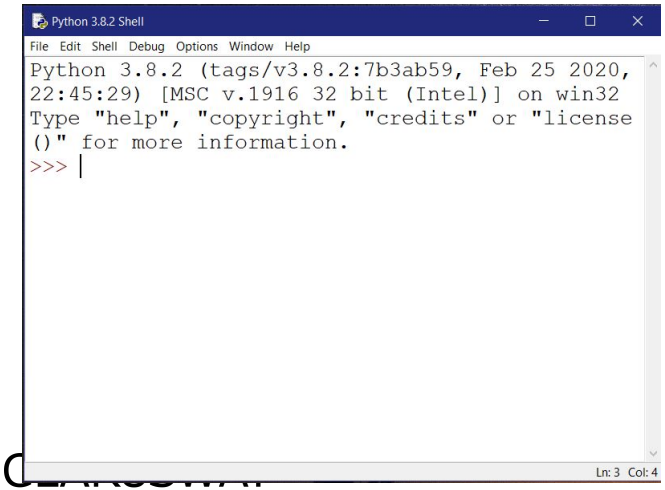
Introduction (review)

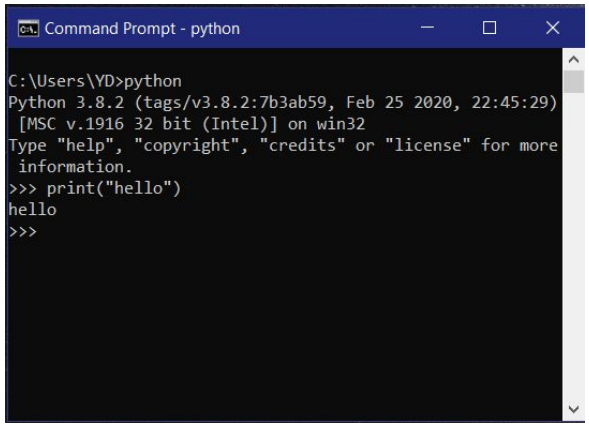
- Have you ever thought, after writing lines of Python code, how they do amazing things?
- We will try to understand how Python codes work, in other words, we will take a look at what's happening when you run the codes.
- Even if we run the simple conventional Python syntax : `print('Hello World!')`, very complex things are performed in the same way.

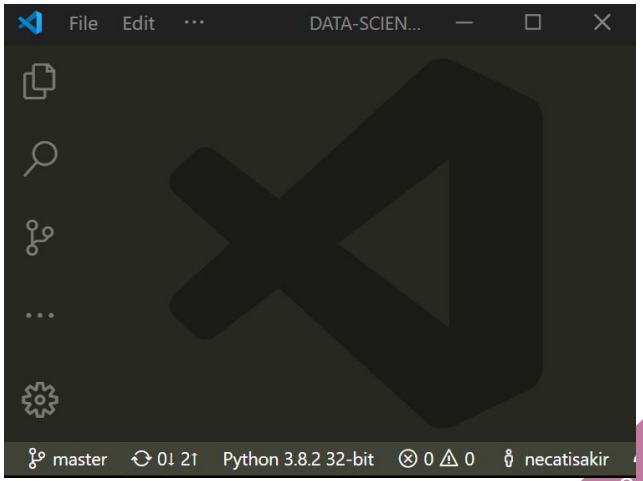
Introduction (review)

- The Python scripts (a file with **.py** extension) contain Python codes and we can run these files via several **interpreter** tools such as :

Python Shell | **OS Console** | **IDEs**







Introduction (review)

- Since the very beginning of this course, you have probably heard terms such as **interpreted** or **compiled** languages. And most likely, you learned that Python was an interpreted type of programming language. What does all this mean? Let's dive into the Pythonic world.
- The process of program execution (the program flow) basically looks like this :

2 The Interpretation

The Interpretation (review)

What is the interpretation process?

?

The Interpretation (review)

What is the interpretation process?

?

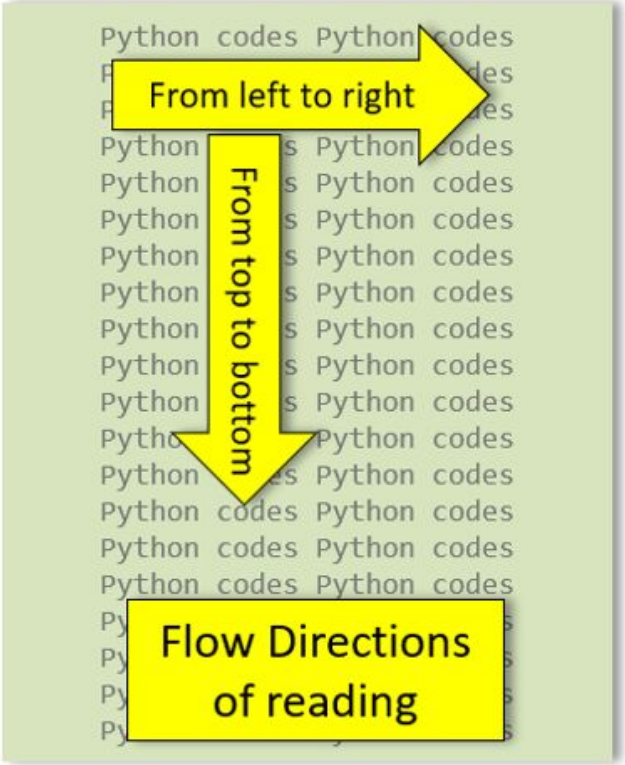
Simply, we can say that this is like **reading your codes.**

The Interpretation (review)

What is the interpretation process?

?

Simply, we can say that this is like **reading your codes.**



The Interpretation (review)

CPython

Python's **default interpreter** is a software written in the **C** programming language known as **CPython**.

- There are several other interpreters available. Let's take a look at these.

Could you remember these interpreters?

Write down their names.

Students, write your response!

The Interpretation (review)

Jython	It is an interpreter that works with a Java-based algorithm and converts Python codes into Java-compatible byte code, which will be executed later by the Java Virtual Machine.

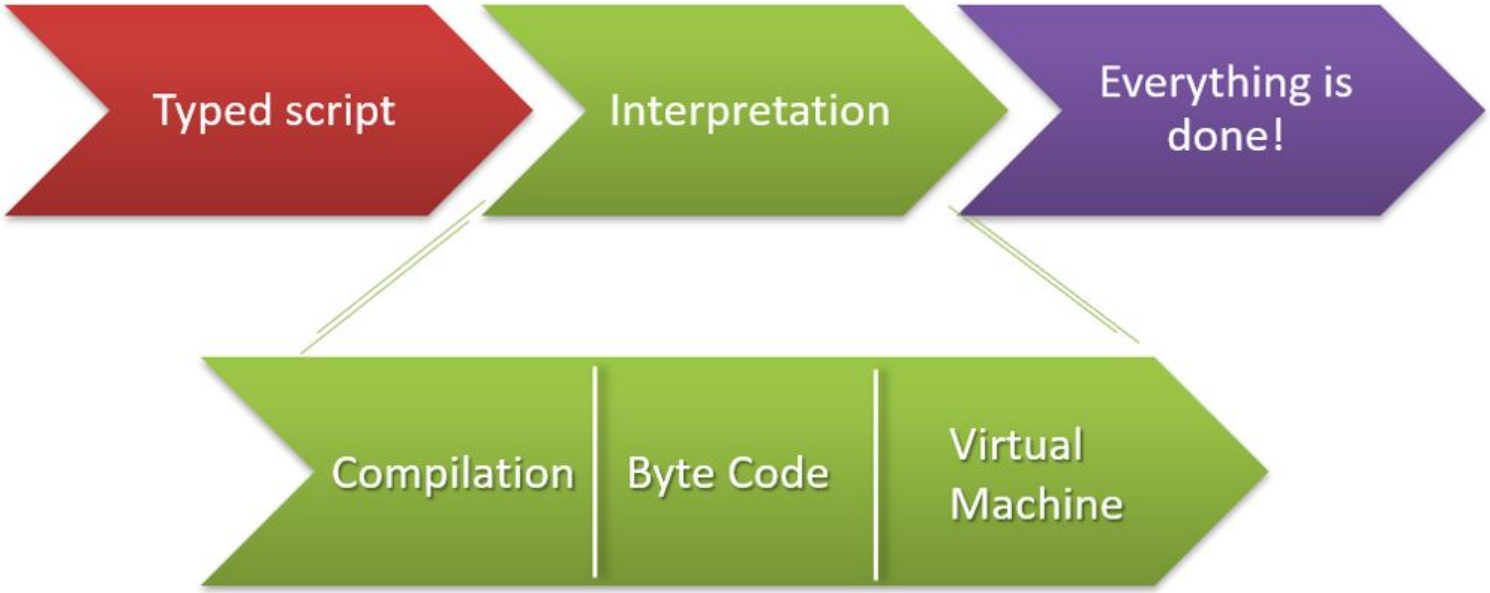
The Interpretation (review)

Jython	It is an interpreter that works with a Java-based algorithm and converts Python codes into Java-compatible byte code, which will be executed later by the Java Virtual Machine.
PyPy	It is a replacement for CPython. It is built using the RPython language that was co-developed with it. The main reason to use it instead of CPython is speed: RPython (Restricted Python) provides some restrictions to the usual Python code.

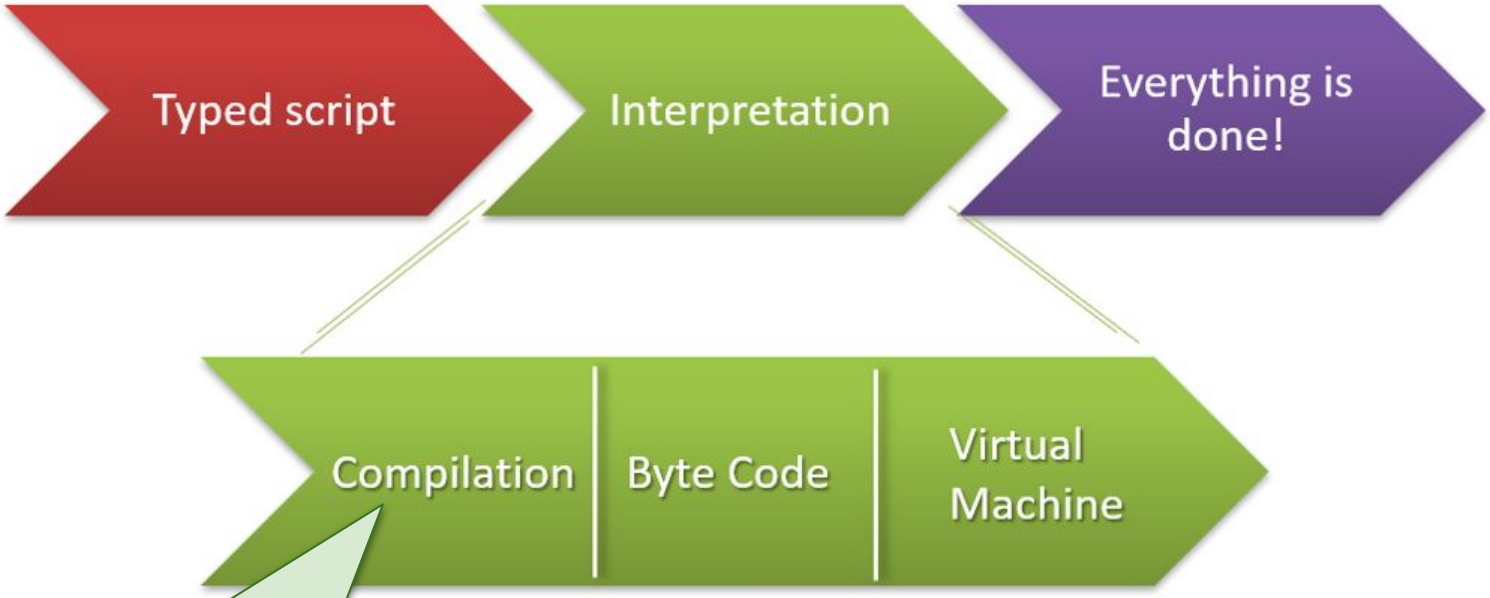
The Interpretation (review)

Jython	It is an interpreter that works with a Java-based algorithm and converts Python codes into Java-compatible byte code, which will be executed later by the Java Virtual Machine.
PyPy	It is a replacement for CPython. It is built using the RPython language that was co-developed with it. The main reason to use it instead of CPython is speed: RPython (Restricted Python) provides some restrictions to the usual Python code.
IronPython	It is an open-source implementation of the Python which is tightly integrated with the `.NET` Framework.

The Interpretation Process (re

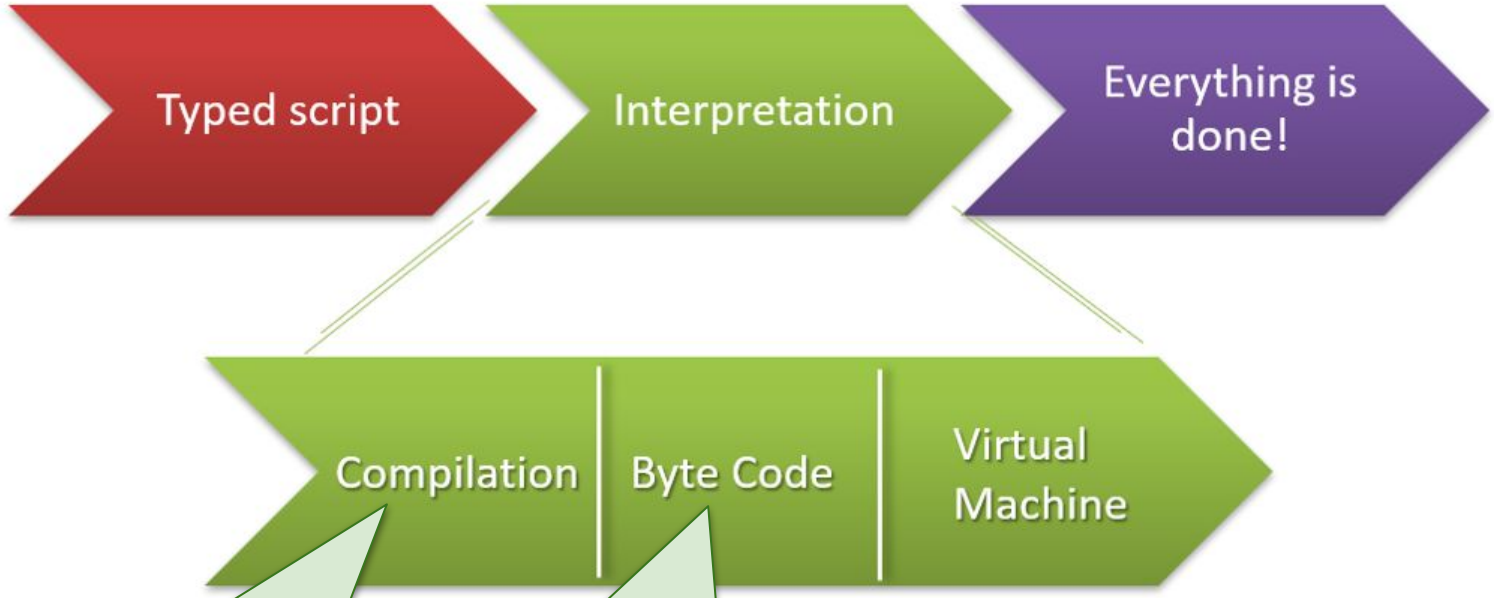


The Interpretation Process (re



The compiler translates your Python statements (source code) into **`byte-code`**.

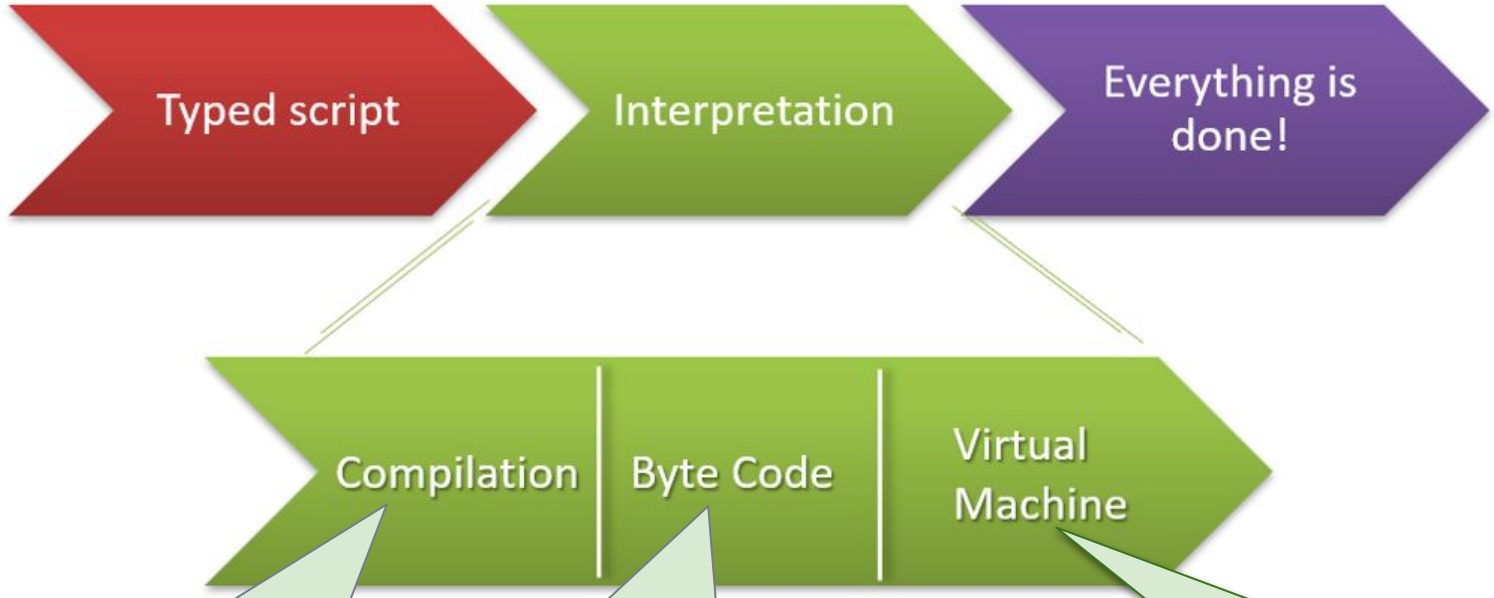
The Interpretation Process (re



The compiler translates your Python statements (source code) into **`byte-code`**.

Byte code is platform-independent, primitive level and specific version of the Python source code.

The Interpretation Process (re



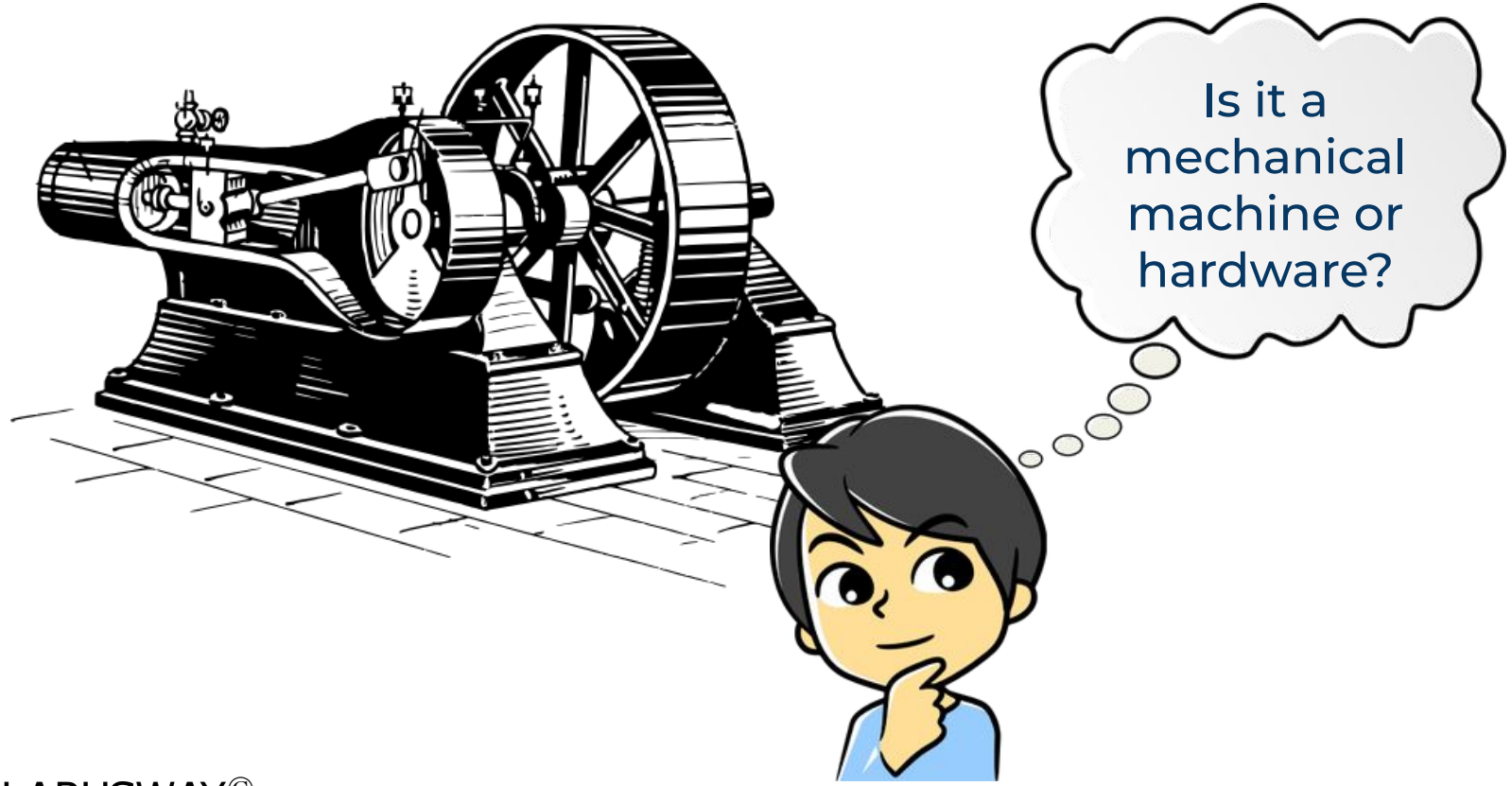
The compiler translates your Python statements (source code) into byte-code.

Byte code is platform-independent, primitive level and specific version of the Python source code.

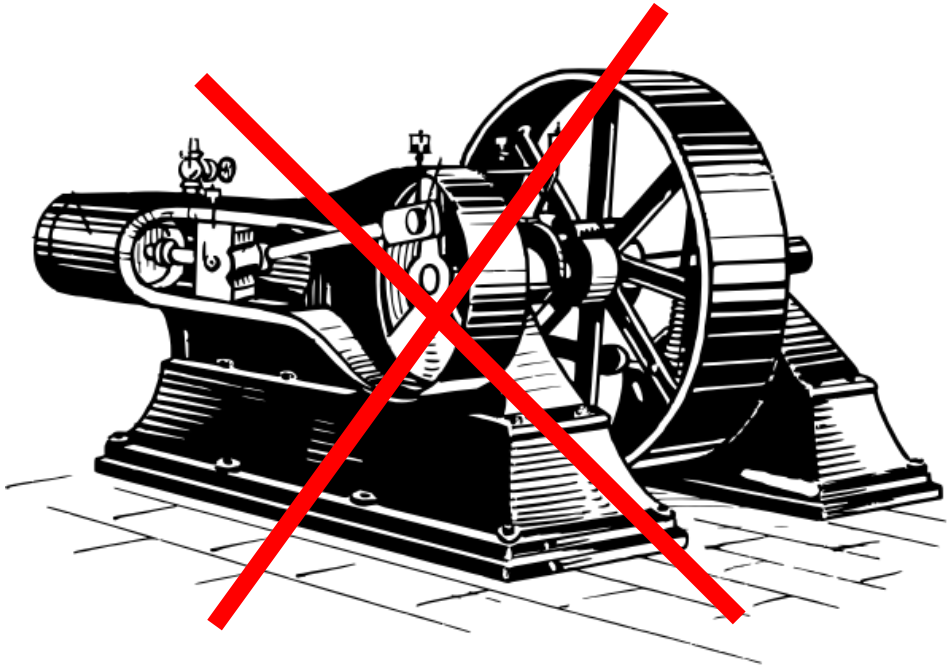
After compilation of Python statements into **byte-code**, it is now time for Python **Virtual Machine** to run.

3 Virtual Machine

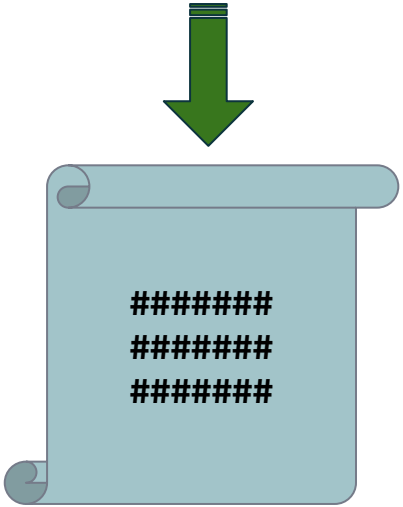
What is a Virtual Machine (review)



Virtual Machine (review)

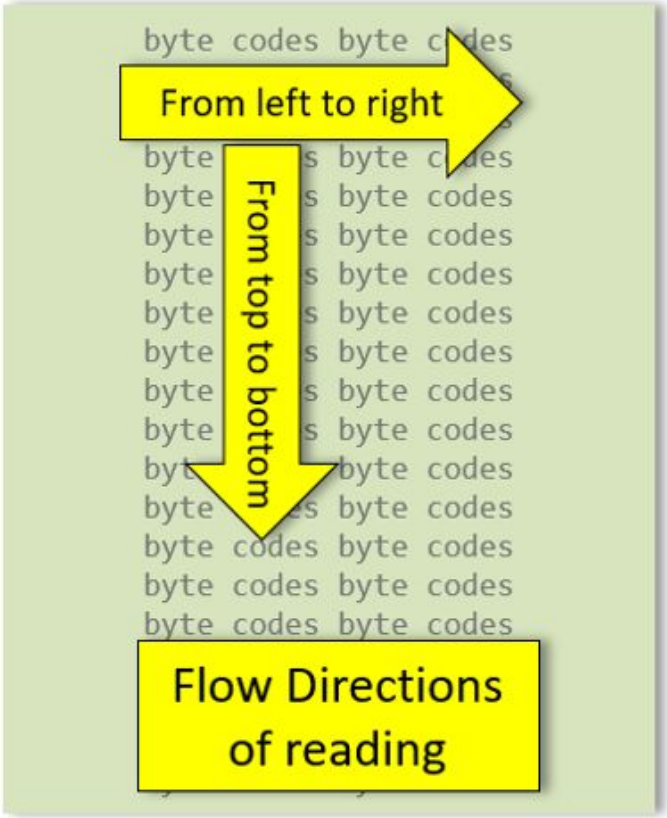


It is actually nothing but software made up of a large piece of code



Virtual Machine (review)

- Just like in compiler, Virtual Machine executes byte-codes that come into it by reading line by line from **top to bottom** and **from left to right**.
- It does not require a separate library or program installation. You do not need to know how the codes that go from the Python statements to the byte-code and from the byte-code to the Virtual Machine are created and executed.



How well did you like this lesson? Did you learn all topics?

THANKS!

Any questions?