INTRODUCTION TO PYTHON

- **What is a programming language?**

⇒ A programming language is a **computer language** that is used by **programmers (or developers) to communicate with computers**. It is a set of instructions written in any specific language (C, C++, Java, Python) to perform a specific task. A programming language is mainly used to **develop desktop applications, websites, and mobile applications**.

- **Python - Python** is a general purpose, dynamic, high-level, and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures. It is *easy to learn* yet powerful and versatile scripting language, which makes it attractive for Application Development. It's syntax and *dynamic typing* with its interpreted nature make it an ideal language for scripting and rapid application development. It supports *multiple programming pattern*, including object-oriented, imperative, and functional or procedural programming styles. It is not intended to work in a particular area, such as web programming. That is why it is known as *multipurpose* programming language because it can be used with web, enterprise, 3D CAD, etc. We don't need to use data types to declare variable because it is *dynamically typed* so we can write a=10 to assign an integer value in an integer variable. It makes the development and debugging *fast* because there is no compilation step included in Python development, and edit-test-debug cycle is very fast.

  Python Interpreter - The Python interpreter is a virtual machine, meaning that it is software that emulates a physical computer. This particular virtual machine is a stack machine: it manipulates several stacks to perform its operations (as contrasted with a register machine, which writes to and reads from particular memory locations).

- **How a python code runs?**

⇒ The execution of the Python program involves 2 Steps:
a) Compilation, and
b) Interpreter

Compilation -
The program is converted into **byte code.** Byte code is a fixed set of instructions that represent arithmetic, comparison, memory operations, etc. It can run on any operating system and hardware. The byte code instructions are created in the **.pyc** file. The .pyc file is not explicitly created as Python handles it internally.

Interpreter -
The next step involves converting the byte code (.pyc file) into machine code. This step is necessary as the computer can understand only machine code (binary code). Python Virtual Machine (PVM) first understands the operating system and processor in the computer and then converts it into machine code. Further, these machine code instructions are executed by processor and the results are displayed.

- **What is the latest version of python?**

⇒ The latest version of python is v3.10.0.

Python 2 vs Python 3

⇒

Comparison Parameter	**Python 2**	**Python 3**
Year of Release	Python 2 was released in the year 2000.	Python 3 was released in the year 2008.
"Print" Keyword	In Python 2, print is considered to be a statement and not a function.	In Python 3, print is considered to be a function and not a statement.
Storage of Strings	In Python 2, strings are stored as ASCII by default.	In Python 3, strings are stored as UNICODE by default.
Division of Integers	On the division of two integers, we get an integral value in Python 2. For instance, 7/2 yields 3 in Python 2.	On the division of two integers, we get a floating-point value in Python 3. For instance, 7/2 yields 3.5 in Python 3.
Exceptions	In Python 2, exceptions are enclosed in notations.	In Python 3, exceptions are enclosed in parentheses.
Variable leakage	The values of global variables do change in Python 2 if they are used inside a for-loop.	The value of variables never changes in Python 3.
Iteration	In Python 2, the xrange() function has been defined for iterations.	In Python 3, the new Range() function was introduced to perform iterations.
Ease of Syntax	Python 2 has more complicated syntax than Python 3.	Python 3 has an easier syntax compared to Python 2.
Libraries	A lot of libraries of Python 2 are not forward compatible.	A lot of libraries are created in Python 3 to be strictly used with Python 3.
Usage in today's times	Python 2 is no longer in use since 2020.	Python 3 is more popular than Python 2 and is still in use in today's times.
Backward compatibility	Python 2 codes can be ported to Python 3 with a lot of effort.	Python 3 is not backward compatible with Python 2.
Application	Python 2 was mostly used to become a DevOps Engineer. It is no longer in use after 2020.	Python 3 is used in a lot of fields like Software Engineering, Data Science, etc.

Why there are so many languages?

⇒ The main reason why there are many programming languages out there is that different problems require different tools to solve them. Each programming language has certain features and characteristics that make it suitable for specific tasks. Just like how doctor's jobs are divided into various specialties, and each solves complex problems and uses the right tools for their job, the same is true for programming.