



CICLO 1

[FORMACIÓN POR CICLOS]

INGLÉS



Lectura: *What is Python?*

What are the benefits of using Python?

Python is a high-level, interpreted, general-purpose programming language. Being a general-purpose language, **it** can be used to build almost any type of application with the right tools/libraries. Additionally, python supports objects, modules, threads, exception-handling, and automatic memory management which help in modelling real-world problems and building applications to solve these problems.

Benefits of using Python

Python is a general-purpose programming language that has a simple, easy-to-learn syntax that emphasizes readability and therefore reduces the cost of program maintenance. Moreover, the language is capable of scripting, is completely open-source, and supports third-party packages encouraging modularity and code reuse.

Its high-level data structures, combined with dynamic typing and dynamic binding, attract a huge community of developers for Rapid Application Development and deployment.

What is a dynamically typed language?

Before we understand a dynamically typed language, we should learn about what typing is. Typing refers to type-checking in programming languages. In a strongly-typed language, such as Python, "1" + 2 will result in a type error since these languages don't allow for "type-coercion" (implicit conversion of data types). On the other hand, a weakly-typed language, such as Javascript, will simply output "12" as result.

Type-checking can be done at two stages:

- ° Static - Data Types are checked before execution.
- ° Dynamic - Data Types are checked during execution.

Python is an interpreted language, executes each statement line by line and thus type-checking is done on the fly, during execution. Hence, Python is a Dynamically Typed Language.

What is an interpreted language?

An Interpreted language executes **its** statements line by line. Languages such as Python, Javascript, R, PHP, and Ruby are prime examples of Interpreted languages. Programs written in an interpreted language runs directly from the source code, with no intermediary compilation step.

What is PEP 8 and why is it important?

PEP stands for Python Enhancement Proposal. A PEP is an official design document providing information to the Python community or describing a new feature for Python or **its** processes. PEP 8 is especially important since **it** documents the style guidelines for Python Code. Apparently contributing to the Python open-source community requires you to follow these style guidelines sincerely and strictly.

What is Scope in Python?

Every object in Python functions within a scope. A scope is a block of code where an object in Python remains relevant. Namespaces uniquely identify all the objects inside a program. However, these namespaces also have a scope defined for **them** where you could use **their** objects without any prefix. A few examples of scope created during code execution in Python are as follows:

- ° A local scope refers to the local objects available in the current function.
- ° A global scope refers to the objects available throughout the code execution since their inception.
- ° A module-level scope refers to the global objects of the current module accessible in the program.

An outermost scope refers to all the built-in names callable in the program. The objects in this scope are searched last to find the name referenced.

TechVidvan (2021, June 27). Retrieved February 4, 2022, from:
<https://techvidvan.com/tutorials/python-programmin>