Python Summary by Diogo Viana

Diogo M. Viana

Ciências da Computação, Introdução a Ciências da Computação – Universidade do Vale de Itajaí (Univali)

diogo.viana@edu.univali.com.br

The history of the programming language Python started when it was idealized for the first time in 1989 at Centrum Wiskunde & Informatica (CWI) in the Netherlands by the mathematician and programmer Guido Van Rossum as a hobby project.

Python was created based on the ABC programming language, interfacing with the Amoeba Operating System, accompanied by the feature of exception handling. Guido, earlier in his career, helped create the ABC language; he liked most of the features but wants to create a clearer language.

The name came from Guido's favorite BBC TV show, 'Monty Python's Flying Circus' and it finally was released in 1991 with a few codes to express the concepts if compared to Java, C, and C++.

How said before Python was created to be a clear programming language, sharing some similarities to the English language, which makes it more readable compared to other programming languages that use semicolons or parentheses to complete commands or brackets to define loops, functions, and classes. Python depends only on new lines and white spaces.

Python has a very helpful library that makes the implementation of programs and algorithms easy. Many of Python's technologies were underpinned by C++, and just like C++, Python is also an object-oriented programming language.

C++ is a high-level, general-purpose programming language created by Bjarne Stroustrup in 1979 as an extension to increase Classes in the C language (the famous "C with Classes"). Over time, C++ became bigger and more modern, and it also has features that support low-level memory manipulation.

C++ has a very rigid learning curve with predefined syntaxes (other syntaxes get to be extremely abstract) and structures; it uses semicolons to complete a command and brackets to define loops, functions, and classes. On the other side, C++ is faster than Python once it is compiled, while Python is interpreted and determines the data type at run time.

The C++ functions, if they are not void functions, need a type and a value to be returned; Python does not have restrictions on the type of function or value to be returned. Both languages have a giant community to support and help beginner and novice programmers.

But in C++, the most active and interested people are those who have some experience in the field. There are a lot of other differences, but we think those are the most important to mention.

Talking about the Python job market, it’s very useful for many projects once it has a lot of libraries for mathematicians like NumPy or computer vision like OpenCV. One of those is web applications, which is a strong area because of Flask and Django frameworks.

Flask is a great choice for web applications that are not too big. It's lightweight and flexible, which makes it good. It's a small framework that has many useful features for building web applications. With Flask, developers can quickly create web applications by using things like URL mapping, templating, and extensions.

If you want to develop complex web applications quickly, Django is a great option. It's a powerful and flexible web framework that can handle big projects. Django has many useful features like ORM, URL mapping, templating, and support for different databases. These features make it easy for developers to build large-scale web applications.

Working with databases is made easier by using Object-Relational Mapping (ORM), an important feature available in both Django and Flask. ORM helps developers interact with databases, which is very important in web application development. It simplifies the process of working with databases by letting developers write Python code that is automatically converted into SQL queries.

Python is also a great choice for machine learning because it's easy to use, has many libraries, and is flexible. Two popular libraries for machine learning in Python are TensorFlow and Scikit-learn. TensorFlow is strong for creating and training deep learning models, while Scikit-learn offers many algorithms and tools for different machine learning tasks. These libraries, along with Python's popularity and active developer community, make it a useful language for building and fixing machine learning models.

https://www.geeksforgeeks.org/history-of-python/

https://www.geeksforgeeks.org/difference-between-python-and-c/

https://www.w3schools.com/python/python_intro.asp

https://www.linkedin.com/pulse/power-python-building-robust-scalable-web-applications-codegama/

https://builtin.com/machine-learning/python-machine-learning

https://www.alura.com.br/artigos/principais-frameworks-de-deep-learning-em-2021?gclid=Cj0KCQjw7aqkBhDPARIsAKGa0oJnZk6-z4ST6yV8YH0P8p7N8l-mvI7eFohy0wKFGhceVt0xfOPmtlgaAvPtEALw_wcB