CMSC 124 Final Project

delCastillo, Kyle Adrian Pareja, King John

January 8, 2023

Python

Purpose (Intended use) and Motivations

Python is a high-level, dynamically-typed, interpreted programming language. It is designed to be readable, easy to learn, and simple to use. Python was originally developed as a bridge between shell-scripting (bash) and C, and today it has been extended to have multiple capabilities (Python Website).

History (Authors, Revisions, Adoption)

Originally developed by Guido Van Rossum in 1989, it was intended as a scripting language between bash-scripting and C but more easily programmable than C and more readable than shell scripts. It was named after Monty Python, a british comedy troupe.

In February 1991, Van Rossum published the source code of Python's interpreter to alt.sources, a Usenet group for open-source code. The first release (0.9.0) had features such as classes, exception handling, functions, core datatypes like list, dict, str, and so on. In January 1994, version 1.0 was released and Guido was later invited to the USA by the US National Institute for Standards and Technology to develop further releases of python.

Python 2.0 released in October 2000 with list comprehensions which was then present in programming languages like Haskell, with other additional features like unicode support and a full garbage collector. The Python Software Foundation was established in March 6 2001, taking over the development of the core Python distribution, managing intellectual rights, developer conferences, and raising funds. In December 2008, Python 3.0 was released, leading to the retirement of continuous updates for Python 2.x on 2020 as many codebases that relied on older versions of Python were slowly translated to Python 3.

Today, the latest stable version of python is 3.11.

Language Features

Python is a high-level, general-purpose programming language that has a wide range of features that make it well-suited for many types of projects. Some of the main features of Python include:

1. Easy to learn and use

   Python has a simple and easy-to-learn syntax, which makes it an ideal language for beginners. It also has a large and active community, which means that there are many resources available for learning and using Python.

2. High-level datatypes

   Python includes built-in support for many high-level data types, such as lists, dictionaries, and sets, which makes it easy to store and manipulate complex data.

3. Object-oriented programming

   Python supports object-oriented programming (OOP), which is a programming paradigm that is based on the concept of "objects", which can contain data and code that manipulates that data. In Python, you can use OOP by defining classes, which are templates for creating objects.

4. Large standard library

   Python has a large standard library, which means that there are many pre-written modules and functions that you can use in your code. This makes it easy to perform common tasks, such as connecting to a database or reading and writing files.

5. Dynamically-typed

   Python is a dynamically-typed language, which means that you don't have to specify the type of a variable when you declare it. This makes it easy to write code quickly, but it can also make it harder to find certain types of errors.

6. Interpreted

   Python is an interpreted language, which means that it is not compiled to machine code before it is executed. Instead, it is interpreted by another program at runtime. This makes it easy to run Python code on any system that has an interpreter, but it can also make the code run slower than compiled languages.

Paradigm(s)

Language Evaluation Criteria

R

Purpose (Intended use) and Motivations

History (Authors, Revisions, Adoption)

Language Features

Paradigm(s)

Language Evaluation Criteria

Feature 1

Description of feature in Language A not present in B

Advantages of having the feature

Advantages of not having the feature

How to implement similar functionality in B

Feature 2

Description of feature in Language A not present in B

Advantages of having the feature

Advantages of not having the feature

How to implement similar functionality in B