Python

Introduction

Case for using Python

Introduction

The best part of programming is the triumph of seeing the machine do something useful. If you've ever spent hours renaming files or updating hundreds of spreadsheet cells, you know how tedious tasks like these can be. But what if you could have your computer do them for you?

Python was conceived by a mathematicians to accomplisg this and more in a legible and understandable fashion. The Python programming language has a wide range of syntactical constructions, standard library functions, and interactive development environment features. Fortunately, you can ignore most of that; you just need to learn enough to be useful.

The short story is this: you can do everything in Python that you can with your computer or an RPA software, but you can read your code after you do it. Without paying an expensive license. That's it—their domains largely overlap, but Python is more focused on producing readable code. For many, the enhanced readability of Python translates to better code reusability and maintainability, making Python a better choice for small companies that will not be written once and thrown away. Perl code is easy to write but can be difficult to read. Given that most software has a lifespan much longer than its initial creation, many see Python as the more effective tool. The somewhat longer story reflects the backgrounds of the designers of the two languages. Python originated with a mathematician by training, who seems to have naturally produced an orthogonal language with a high degree of uniformity and coherence.

"Life is short.

You need Python”

Bruce Eckel

Consider this: when people create a painting or a sculpture, they do so largely for themselves; the prospect of someone else changing their work later doesn't enter it. This is a critical difference between art and engineering. When people write software, they are not writing it for themselves. In fact, they are not even writing primarily for the computer. Rather, good programmers know that code is written for the next human being who must read it to maintain or reuse it.

The Python Conceptual Hierarchy
Before we get to the code, let's first establish a clear picture of how this chapter fits into the overall Python picture. From a more concrete perspective, Python programs can be decomposed into modules, statements, expressions, and objects, as follows:

1. *Programs are composed of modules.*
2. *Modules contain statements.*
3. *Statements contain expressions.*
4. *Expressions create and process objects.*

Click here to know more...

Python is an open source versatile, easy to use and fast to developprogramming languague with a vibrant community. Has all the libraries you can imagine. you can do more with less code.

- *Productivity.*
- *Speed limitations.*
- *Problems with threading.*

"Python is fast enough for our site and allows us to produce maintainable features in record times, with a minimum of developers."

YouTube