04. Why Python? (Overview)

Sunday, November 5, 2023
8:50 AM

Python Overview for New Bee Programmer

In this lecture, I'm going to do a

- Overview of Python
- Python History
- Why choose Python for programming
- What is difference between base Python vs lots of additional libraries python.
- What you can do if Python.

Let's start with just a very brief history of Python.

HISTORY

It was created in 1990 by Guido van Rossum, as shown in picture below:

https://gvanrossum.github.io/

As Python 3 was released in 2008 with subsequent releases happening after that, such as 3.1, 3.2, and so on. When Guido van Rossum developed python it was specifically designed with a very high focus on readability of code and as an easy to use language, since start of creation of python language. Which is few of the things, that have really allowed Python to explode in popularity.

WHY CHOOSE PYTHON?

A. From the beginning of python creation. Python was designed for clear, logical code that is easy to read and easy to learn. As compare to other programming languages use braces and brackets in code may lead convoluted and hard to read. But Python makes use of something called whitespace and indentation, which will be covered later on in this course that makes its code very accessible. Where if you know any other programming language, at least you can usually read Python without knowing the real **"syntax"** *(Words that make programming come alive)* behind it. Which indicate how readable is python language.

B. Also python having lots of existing libraries and frameworks are already available, it allow users to apply Python to a wide variety of tasks. This is also another reason that help made Python popular in recent years.

C. .Another reason that developer's time is much valuable as a person, therefore Python language help really focuses on optimizing developer time, rather than a computer's computational processing time. So the whole language of Python is designed for users to really get prototyping up and running very quickly.

D. It also has fantastic documentation and lots of helpful resources online. Where you just Google Python help or check out official fantastic documentation at ***docs.python.org/3***.

WHAT DIFFERENCE BETWEEN "Base" Python VS "Additional Libraries" Python?

This course primary focuses is on **"base"** *(core)* python. Which consists of the core components of the language and writing **"scripts"** *(sequence of instructions)* and small programs.

In-addition, python is often referred to as a program that comes with quote, "Batteries included", which means even the base version of Python actually comes in with lots of additional addon **"modules"** *(extension to a main program)* that you can use for writing scripts. As example python has random module that quickly create random number, do math and etc...

So, once we learn about base Python and its base modules, later learn about outside **"libraries"** *(location where set of small program found)* and "frameworks" *(Combine of small program to help create starting base of building application)*. Which greatly help expand Python programming language capabilities. These outside libraries and frameworks consider to be 3rd party and that are not included with **"base"** Python. But you can easily download and work with your basic Python skills.

WHAT CAN YOU DO?

- *Help **Analyst** do automating simple tasks with outside library.*
  - Automatically Search and Edit file.
  - Automatically Scrape Information from website.
  - Automatically Read and Write Excel Files
  - Work with PDFs.
  - Automate Emails and TEXT messages
  - Automatically fill out forms
- *Help **Data Scientist** in Data Science and Machine learning world.*
  - Analyze large Data file using library NumPy and Pandas.
  - Create Visualization file using library Seaborn and Matplotlib.
  - Do machine learning tasks like, create and run predictive algorithms using library like scikit-learn and TensorFlow.
- *Help **Web Developer**, to create website, interactive dashboards and communicate backend server user data with website.*
  - Create website using Django framework.
  - Create interactive dashboards using library like Plotly and Dash.
  - Communicate user data (API) on website by using Flask framework.

So let's go ahead and get started with setting you up for the course.
Thanks, and I'll see you at the next lecture.