

Python 101

Introducción

03/02/2022

Python 101

Público

Introduction

Cuando Python apareció en la escena del software a principios de la década de 1990, dio lugar a lo que ahora es una especie de conflicto clásico entre sus defensores y los de otro popular lenguaje de scripting, Perl. Personalmente, creo que el debate está cansado e injustificado hoy en día: los desarrolladores son lo suficientemente inteligentes como para sacar sus propias conclusiones. Sin embargo, este es uno de los temas más comunes que me preguntan en el camino de la formación, y subraya una de las principales razones por las que la gente elige usar Python; me parece apropiado decir unas breves palabras sobre ello aquí.

La historia corta es esta: puedes hacer todo en Python que puedes con RPA, pero puedes leer tu código después de hacerlo. Sin tener que pagar una costosa licencia. Eso es todo: sus dominios se superponen en gran medida, pero Python se centra más en la producción de código legible. Para muchos, la mayor legibilidad de Python se traduce en una mejor reutilización y mantenimiento del código, lo que hace que Python sea una mejor opción para las pequeñas empresas que no se escribirá una vez y se tirará a la basura. El código Perl es fácil de escribir, pero puede ser difícil de leer. Dado que la mayoría del software tiene una vida útil mucho más larga que su creación inicial, muchos ven a Python como la herramienta más eficaz. Python se originó con un matemático de formación, que parece haber producido naturalmente un lenguaje ortogonal con un alto grado de uniformidad y coherencia.



"Life is short.

You need Python"

Bruce Eckel

Considere lo siguiente: cuando la gente crea un cuadro o una escultura, lo hace en gran medida para sí misma; la posibilidad de que otra persona cambie su obra más adelante no entra en juego. Esta es una diferencia fundamental entre el arte y la ingeniería. Cuando la gente escribe software, no lo hace para sí misma. De hecho, ni siquiera escriben principalmente para el ordenador. Más bien, los buenos programadores saben que el código se escribe para el siguiente ser humano que debe leerlo para mantenerlo o reutilizarlo.

La jerarquía conceptual de Python Antes de llegar al código, establezcamos primero una imagen clara de cómo este capítulo encaja en el panorama general de Python. Desde una perspectiva más concreta, los programas de Python pueden descomponerse en módulos, declaraciones, expresiones y objetos, como sigue:

- 1. Los programas se componen de módulos.*
- 2. Los módulos contienen sentencias.*
- 3. Las sentencias contienen expresiones.*
- 4. Las expresiones crean y procesan objetos.*

Conclusión

Python es un lenguaje de programación de código abierto versátil, fácil de usar y rápido de desarrollar, con una comunidad muy activa. Tiene todas las bibliotecas que puedas imaginar. Puedes hacer más con menos código:

- Productividad.*
- Sin Limitaciones de escalabilidad.*
- Bajo costo.*

**"Python es lo
suficientemente rápido
para nuestro sitio producir
características
mantenibles en tiempos
récord, con un mínimo de
desarrolladores"**

- *YouTube*