¿Qué es Python? Introducción

Si has llegado hasta aquí es posible que quieras aprender Python, o tal vez simplemente tengas curiosidad por saber algo más de este lenguaje de programación tan popular. En este post te intentaremos convencer de porqué debes aprender Python, sin importar en que trabajes o cual sea tu sector.

Aprovechamos de paso a darte la bienvenida a nuestro libro, un lugar donde podrás encontrar tutoriales de Python completamente gratis desde lo más básico a conceptos avanzados, y con diferentes aplicaciones: análisis de datos, automatización de tareas, web scraping, machine learning, data science, desarrollo de videojuegos, interfaces gráficos o finanzas.

Dicho esto, empecemos. A diferencia de lo que mucha gente puede pensar, Python es un lenguaje que data de los años 1990s, y su creación se le atribuye al neerlandés Guido van Rossum[1]. Recibió su nombre por los humoristas Monty Python.

Su última versión es Python 3, y es la que te recomendamos usar ya que las anteriores ya tienen soporte oficial. En este blog usaremos esta siempre para todo.

De acuerdo con StackOverflow insights[2] en la siguiente gráfica podemos ver el número de preguntas vistas en la plataforma acerca de Pyhton. Podemos ver que Python lleva casi dos años en el podio, una auténtica burrada.

De hecho un estudio realizado hace un tiempo en el 2017[3] ya ponía a Python como uno de los favoritos en los países con mayores ingresos como Alemania, Reino Unido o Estados Unidos entre otros.

Python was the most visited tag on Stack Overflow within high-income nations

Podemos ver por lo tanto que existe una tendencia clara y seguramente el interés en Python vaya a seguir creciendo en los próximos años. Estos son los motivos por lo que ha crecido tanto y lo seguirá haciendo:

Se trata de un lenguaje fácil de aprender, con una sintaxis muy sencilla que se asemeja bastante al pseudocódigo. En otras palabras, poco código hace mucho.

Su uso no está ligado a un sector concreto. Por ejemplo el lenguaje R es útil para análisis de datos, pero no puede ser usado para desarrollo web. Python vale para todo.

Tiene una comunidad enorme, además de gran cantidad de librerías para hacer prácticamente cualquier cosa, literalmente.

Es un lenguaje multiplataforma, por lo que el mismo código es compatible en cualquier plataforma (Windows, macOS, Linux) sin hacer nada.

Por lo general se puede hacer desarrollos en Python más rápidamente que en otros lenguajes, acortando la duración de los proyectos.

Usos de Python

Como hemos dicho Python es un lenguaje muy transversal, usado en diferentes industrias y para diferentes fines. Veamos algunos de las empresas que usan Python:

YouTube usa Python[4] en la parte del servidor, unida a otros lenguajes como Java o Go.

Netflix usa Python[5] par automatizar tareas, explorar datos y labores de aprendizaje automático entre otras.

La NASA usa Python[6] en gran cantidad de programas científicos.

JPMorgan[7] ya dijo hace varios años que se esperaba de sus analistas financieros supieran Python.

Python es también usado para fines muy diversos como son los siguientes:

Desarrollo Web: Existen frameworks como Django, Pyramid, Flask o Bottle que permiten desarrollar páginas web a todos los niveles.

Ciencia y Educación: Debido a su sintaxis tan sencilla, es una herramienta perfecta para enseñar conceptos de programación a todos los niveles. En lo relativo a ciencia y cálculo numérico, existen gran cantidad de librerías como SciPy o Pandas.

Desarrollo de Interfaces Gráficos: Gran cantidad de los programas que utilizamos tienen un interfaz gráfico que facilita su uso. Python también puede ser usado para desarrollar GUIs con librerías como Kivy o pyqt.

Desarrollo Software: También es usado como soporte para desarrolladores, como para testing.

Machine Learning: En los último años ha crecido el número de implementaciones en Python de librerías de aprendizaje automático como Keras, TensorFlow, PyTorch o sklearn.

Visualización de Datos: Existen varias librerías muy usadas para mostrar datos en gráficas, como matplotlib, seaborn o plotly.

Finanzas y Trading: Gracias a librerías como QuantLib o qtpylib y a su facilidad de uso, es cada vez más usado en estos sectores.

De hecho a día de hoy prácticamente cualquier API, librería o servicio que existe en el mundo tiene una versión para Python, bien sea de manera nativa o a través de un wrapper.

Comunidad

La comunidad de Python es inmensa, con alrededor de 8.2 millones de personas en el mundo (a fecha de 2019)[8], una cifra que supera ya a los usuarios de Java.

Es importante el tamaño de la comunidad, porque cuanto más grande sea, mayor soporte se le dará al lenguaje y mayor número de personas compartirán sus problemas y se ayudarán a resolverlos.

Otra de las características de la comunidad Python son las famosas PyCon, unas convenciones anuales llevadas a cabo en gran número de países, donde los desarrolladores se reúnen para compartir ideas.

Es también importante mencionar la Python Software Foundation, una organización sin ánimo de lucro que se dedica a promover, proteger y desarrollar el lenguaje Python.

Características de Python

Como cualquier otro lenguaje, Python tiene una serie de características que lo hacen diferente al resto. Las explicamos a continuación:

Es un lenguaje interpretado, no compilado.

Usa tipado dinámico, lo que significa que una variable puede tomar valores de distinto tipo.

Es fuertemente tipado, lo que significa que el tipo no cambia de manera repentina. Para que se produzca un cambio de tipo tiene que hacer una conversión explícita.

Es multiplataforma, ya que un código escrito en macOS funciona en Windows o Linux y vice versa.

Tal vez algunos de estos conceptos puedan resultarte extraños si estás empezando en el mundo de la programación. El siguiente código pretende ilustrar algunas de las características de Python.

Algunas cosas curiosidad que en otros lenguajes no pasan. La función acepta un parámetro entrada pero no se especifica su tipo. La x almacena primero una cadena, luego un float y luego un integer. La función funcion() es llamada con un int, pero su valor se divide entre 2 y el resultado es convertido automáticamente en un float.

def funcion(entrada):

return entrada/2

x = "Hola"

x = 7.0

x = int(x)

x = funcion(x)

print(x)

print(type(x))

# 3.5

# <class 'float'>