# 2

# PYTHON

Python es un lenguaje interpretado, de alto nivel y enfocado principalmente a la legibilidad y facilidad de aprendizaje y uso.

Python es un lenguaje orientado a objetos, aunque soporta otros paradigmas como la programación funcional y, por supuesto, la programación imperativa (multiparadigma).

Python es un lenguaje multiplataforma, lo que significa que puede usarse en multitud de sistemas distintos. Funciona en ordenadores con sistemas operativos Linux, BSD, Apple, Windows y muchos otros, pero también hay versiones para otros dispositivos, como terminales telefónicos inteligentes, etc.

Naturalmente, Python dispone, por medio del uso de bibliotecas, de herramientas para aprovechar las posibilidades concretas que le brinda cada plataforma, pero también es posible escribir programas evitando el uso de esas bibliotecas específicas, de modo que esos programas funcionen indistintamente en cualquier ordenador.

Python es software libre, y se distribuye bajo la licencia “Python Software Foundation License”. Esto, entre otras cosas, significa que se distribuye gratuitamente y no necesita del pago de licencias o royalties para su uso, ya sea privado o comercial.

Guido van Rossum es el creador y BDLF (*Benevolent Dictator for Life*, Benevolente dictador de por vida) de Python; y la filosofía que quiso darle es que el código debe ser limpio y legible, evitando “atajos” o construcciones que dificulten la comprensión del programa. *Python es simple sin ser limitado*.

Estas características hacen que Python sea un lenguaje ideal para aprender e iniciarse en la programación.

A pesar de que Python es una muy buena elección para aprender a programar no es un lenguaje diseñado para aprender a programar. Python es un lenguaje completo perfectamente funcional, muy potente, y viene acompañado por una serie de paquetes que facilitan funciones para el trabajo con casi cualquier cosa. A este respecto, se dice que Python viene con “pilas incluidas”.

### 2.1. SOFTWARE LIBRE

Ya hemos mencionado un par de veces que Python es software libre.

Aunque a menudo se interpreta equivocadamente como "software gratuito", el software libre es una filosofía que afirma que el software debe ser accesible a los usuarios.

A grandes rasgos, significa que **los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software**

En particular, los defensores del software libre afirman que hay una serie de derechos o "libertades del software" que todo usuario de un programa debería tener.

Según las propias palabras de la Free Software Foundation (*https://www.gnu.org/philosophy/free-sw.html*) , las cuatro libertades del software libre son:

- La libertad de ejecutar el programa como se desee, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y cambiarlo para que haga lo que se desee (libertad 1). El acceso al código fuente es una condición necesaria para ello.
- La libertad de redistribuir copias para ayudar a otros (libertad 2).
- La libertad de distribuir copias de sus versiones modificadas a terceros (libertad 3). Esto le permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones. El acceso al código fuente es una condición necesaria para ello.

A lo largo de los últimos tiempos, la presencia del software libre se ha extendido por todo el mundo, hasta tal punto que el software libre está detrás de la mayor parte de los grandes logros de la informática, desde la propia Internet hasta el sistema operativo Linux pasando por el propio Python.

### 2.2. EL ZEN DE PYTHON

El zen de Python (*http://www.python.org/dev/peps/pep-0020/*) viene a ser la carta de principios del lenguaje, su filosofía. Describe una serie de reglas que deberían seguirse tanto en el desarrollo del propio lenguaje, como en los programas que se hagan con él.

```
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

### 2.3. PYTHON 2 VERSUS PYTHON 3

Actualmente, existen dos versiones de Python conviviendo.

Por un lado, está la serie de "Python 2", que se encuentra en su versión 2.7.18 (abril 2020), y simultáneamente, existe "Python 3", que se encuentra en su versión 3.13.2 (octubre 2024).

Ambas versiones son muy similares, pero lo suficientemente diferentes para que un script escrito en Python 2 no sea compatible con Python 3, y viceversa.

Actualmente, **Python 3 es, con mucha diferencia, el más popular**. Python 2 llegó al **fin de su soporte en 2020**, lo que significa que ya no recibe actualizaciones ni parches de seguridad.

**Razones por las que Python 3 es más popular:**

1. **Python 2 ya no recibe mantenimiento**: No hay más actualizaciones ni parches de seguridad.
2. **Compatibilidad con nuevas tecnologías**: Herramientas modernas de IA, ciencia de datos y desarrollo web solo funcionan en Python 3.
3. **Mayor rendimiento y optimización**: Python 3 es más eficiente en la gestión de memoria y ejecución de código.
4. **Mejoras en la sintaxis**: Manejo más limpio de cadenas de texto (Unicode por defecto), mejor tipado y mejores estructuras de datos.

5. **Comunidad y documentación**: Los recursos y cursos ya están enfocados en Python 3.

Hoy en día, **Python 2 solo se usa en sistemas heredados** que no han sido migrados aún, pero en el desarrollo moderno, **Python 3 es el estándar absoluto**.

### 2.4. ¿QUÉ HEMOS VISTO EN ESTE TEMA?

Las características de Python y sus variantes. La filosofía que tiene detrás y el modo de enfocar la programación implícito en este lenguaje.

#### 2.4.1. Tareas sugeridas

No sería mala idea consultar la página de Wikipedia dedicada a Python, para empezar a conocer algo de este lenguaje de programación:

- ¿De dónde viene el nombre de Python?
- ...