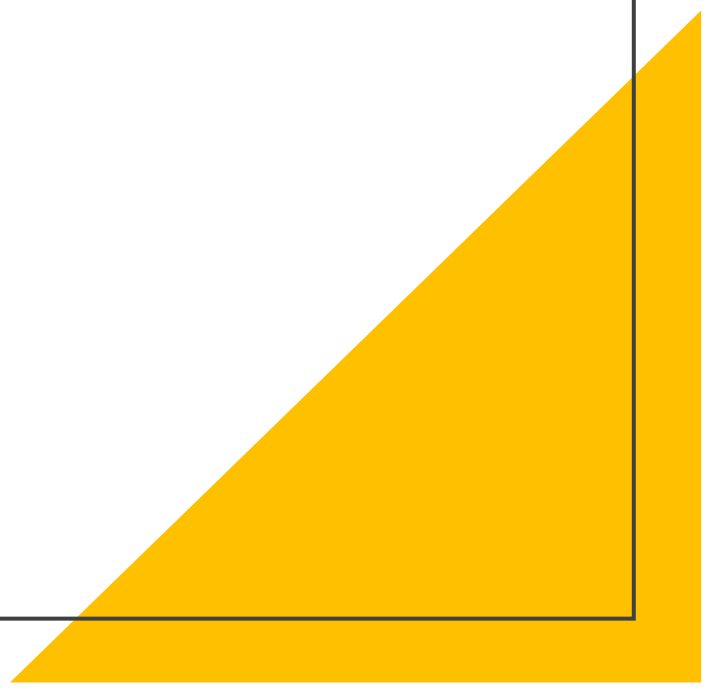


Python3

Conceptos Básicos

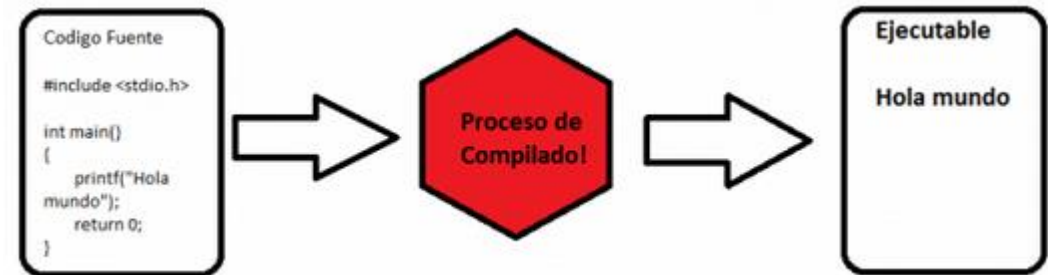
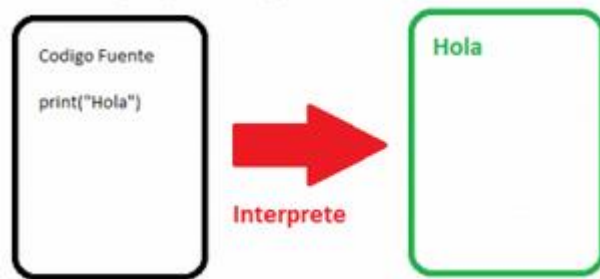


Características principales

- **lenguaje interpretado**, se compila en tiempo de ejecución
- **multiplataforma**: Windows, Linux, Unix, MacOSX, ironpython (.NET), jpython (JVM), activepython (Solaris) y otros.
- **multiparadigma**: soporta **programación orientación a objetos**, **programación estructurada**, **programación imperativa**, *programación funcional* y **programación orientada a eventos**, **programación declarativa...** entre otros. Se le conoce como la *navaja suiza* de los programadores.
- **tipado dinámico**, es decir, los objetos en tiempo de ejecución tiene un tipo, pero es **fuertemente tipado**, es decir, que un valor no cambia de repente sino que hay que hacer su conversión
- **indentación** es el formato del código, lo que lo hace muy **legible**
- **multipropósito**, desde una aplicación de gestión, al tratamiento de BigData (*PySpark*), DataScience (*Pandas*, *Numpy*, *Jupyter* o *Matplotlib*), Machine Learning, Inteligencia Artificial... quizás uno de los lenguajes con más futuro

Lenguaje - Comparativa

- Lenguaje Interpretado (Python)
- Lenguaje Compilado (C)



Breve historia de Python

- Creado a finales de los 80 por el Holandés **Guido Van Rossum**.
- Deriva de un lenguaje previo llamado **ABC**.
- El objetivo era un lenguaje de programación **de alto nivel**, con una **sintaxis muy limpia, fácil de leer y multiplataforma**.
- Con soporte para distintos estilos de programación: Imperativa, orientada a objetos y funcional.
- El nombre proviene del grupo de humor inglés **Monty Python**, no de la serpiente.



Zen de Python

El **Zen de Python** es una colección de 20 principios de software que influyen en el diseño del Lenguaje de Programación Python, de los cuales 19 fueron escritos por Tim Peters en junio de 1999.



Principios

Bello es mejor que feo.

Explícito es mejor que implícito.

Simple es mejor que complejo.

Complejo es mejor que complicado.

Plano es mejor que anidado.

Espaciado es mejor que denso.

La legibilidad es importante.

Los casos especiales no son lo suficientemente especiales como para romper las reglas.

Sin embargo la practicidad le gana a la pureza.

Los errores nunca deberían pasar silenciosamente.

A menos que se silencien explícitamente.

Frente a la ambigüedad, evitar la tentación de adivinar.

Debería haber una, y preferiblemente solo una, manera obvia de hacerlo.

A pesar de que eso no sea obvio al principio a menos que seas Holandés.

Ahora es mejor que nunca.

A pesar de que nunca es muchas veces mejor que *ahora* mismo.

Si la implementación es difícil de explicar, es una mala idea.

Si la implementación es fácil de explicar, puede que sea una buena idea.

Los espacios de nombres son una gran idea, ¡tenemos más de esos!

IMPORTANTE

— Lo que el cliente necesita

