

Comenzando con Python

Luis Miguel de la Cruz Salas

GMMC, Recursos Naturales
Proyecto PAPIME PE101019

Octubre 2019



Acerca de los cursos

- [Temarios](#)
- [Python 3.7](#)
- luiggi@igeofisica.unam.mx
- Filosofía del curso:
 - “Haremos lo mínimo indispensable, para Uds. hagan lo máximo posible.”





Brevísima historia de Python

"Over six years ago, in December 1989, I was looking for a programming project that would keep me occupied around Christmas. My office ... would be closed, but I had a computer, and not much else on my hands. I decided to write an interpreter for the new scripting language I had been thinking about lately: a descendant of ABC that would appeal to Unix/C programmers. I chose Python as a working title for the project, being in a slightly rebellious mood (and a big fan of Monty Python's Flying Circus)."



[Guido](#)



[Monty Python](#)



Propuesta de Van Rossum a DARPA¹ *Computer Programming for Everybody (CP4E)*²

- Python debería ser fácil, intuitivo y tan potente como sus principales competidores.
- El proyecto sería de Código Abierto para que cualquiera pudiera colaborar.
- El código escrito en Python sería tan comprensible como cualquier texto en inglés.
- Python debería ser apto para las actividades diarias permitiendo la construcción de prototipos en poco tiempo.

¹ Defense Advanced Research Projects Agency, 1999

² <https://www.python.org/doc/essays/cp4e/>



Python hoy en día

- El tercer lenguaje más popular en [GitHub Octoverse](#).

	Javascript	Java	Python	C/C++	Matlab	Fortran
Simple	✗	✗	💖	✗	👍	✗
Eficiente	✗	✗	💖	👍	👍💰	💖
Barato	👍	👍	💖	👍	✗	👍
Genérico	✗	👍	💖	👍	👍💰	✗
Portable	👍	👍	💖	✗	👍💰💣	✗

¡Python es Bueno, Bonito y Barato!



Diferencias entre Python 2 y Python 3

Python 2

- Sigue usándose por herencia.
- La versión 2.7 fue la última, liberada en 2010.

Python 3

- Es el presente y el futuro.
- La versión 3.0 fue la primera, liberada en 2008.

¿Qué versión debería usar?

- Proyecto nuevo → 3.7
- Proyecto que depende de una biblioteca en Python 2.7 sin soporte en Python 3.7:
 - Intenta portar esa dependencia a la versión 3.7 (y ser felices)
 - Solo en caso de que no sea posible usarás la versión 2.7

Más información: <https://wiki.python.org/moin/Python2orPython3>



Instalación

Distribuciones:

- Oficial: <https://www.python.org/downloads/>
- Anaconda: <https://www.anaconda.com/>
- Enthought Canopy: <https://www.enthought.com/>
- Otras: <https://wiki.python.org/moin/PythonDistributions>

Entornos de desarrollo:

- | | | | |
|-----------|-----------|-------------------|-----------|
| • Python | • Jupyter | • PyDev (Eclipse) | • Vim |
| • iPython | • Spyder | • Visual Studio | • Emacs |
| | • PyCharm | • Sublime | • Notepad |



Pensando como *Pythonista* : las bases



¿Qué es Python realmente?

- Un **lenguaje** es un recurso que hace posible la comunicación.
- Un **lenguaje de programación** es aquella estructura que, con una cierta base sintáctica y semántica, permite implementar algoritmos para ejecutarse en una computadora.
- Un **lenguaje de programación de alto nivel** contiene elementos del lenguaje humano y permite una comunicación simple con una computadora.
- Una **interfaz** es la conexión funcional entre dos sistemas que proporciona una comunicación de distintos niveles permitiendo el intercambio de información.





¿Qué es Python realmente?



El Turco, Wolfgang von Kempelen (1769).

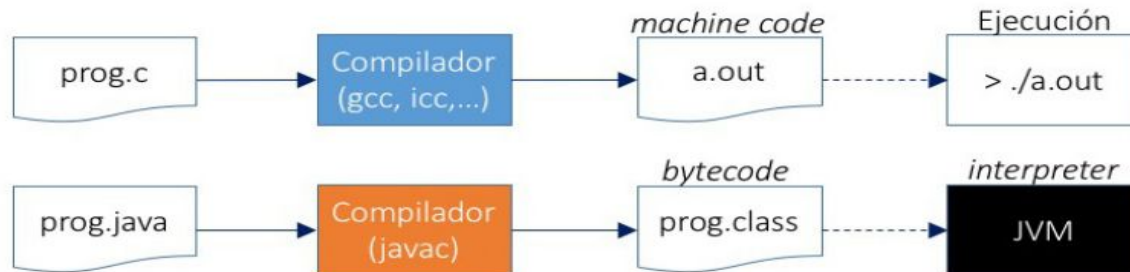
Sentado frente al hombre que todos temían, en 1809 se produjo una batalla que resultaría inaudita para la época. Mucho antes de que el hombre soñara con conceptos como la IA y de que las máquinas amenazaran con cambiar la vida del ser humano, un autómatas se enfrentó al mismísimo Napoléon Bonaparte en una partida de ajedrez. Y la máquina ganó ... ([historia completa](#)).

Python es una interfaz¹ entre el ser humano y una computadora.

¹ Existen muchas implementaciones de esta interfaz



¿Interpretado o Compilado?

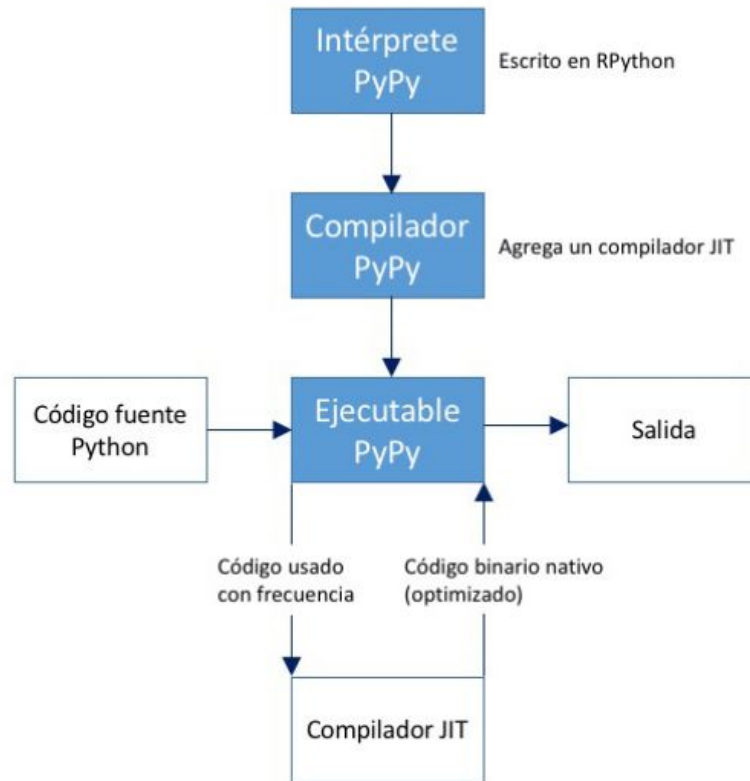
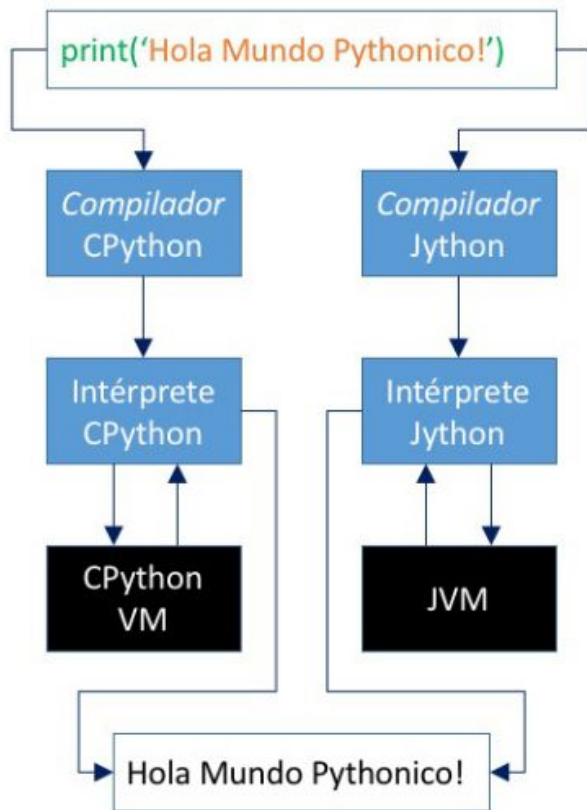


Implementación	Máquina Virtual	Compatibilidad
CPython	CPython VM	C
JPython	JVM	Java
IronPython	CLR	C#
Brython	Motor Javascript	Javascript
RubyPython	Ruby VM	Ruby
PyPy	Compilado	RPython

Especificación:
[The Python Language Reference](#)



¿Intepretado o Compilado?



Observación

- Suele decirse que Python es:
 - 1) interpretado, 2) tipado dinámico, 3) genérico, 4) poco eficiente.
- En realidad estas cuatro características son una propiedad de la implementación, no de la interfaz Python.



¿Variables o etiquetas?

Lenguaje C

```
int a = 1;
```



```
a = 2;
```



```
int b = a;
```



Python

```
a = 1
```



```
a = 2
```



```
b = a
```



1^{¿?}



“No solo no estoy aprendiendo nada, sino que estoy olvidando lo que ya sabía”

Milhouse Van Houten, The Simpsons.

Resumen



Características principales

- En Python todo es un objeto.
- El término etiqueta es usado en vez de variable.
- Existe el tipado dinámico:
 - el tipo de un objeto se evalúa durante la ejecución.
- Existe la generación espontánea:
 - los objetos se crean, se transforman y se destruyen.
- Los objetos viven en espacios de nombres.





Propiedades de un objeto

- Una identidad única (`id()`)
- Un tipo (`type()`).
- Un estado interno.
- Uno o varios nombres (etiquetas).
- Un comportamiento.



El Zen de Python

- **import this**
- [pep20 by example](#)
- [A Brief Analysis of "The Zen of Python"](#)
- [Code Style](#)

