

# Breve Historia del Python

---

¿De dónde viene y hacia donde va?

Luis Miguel de la Cruz Salas  
Mario Arturo Nieto Butrón

Instituto de Geofísica, UNAM

Febrero 2018

# Breve Historia del Python

- 1 **Acerca del curso**
- 2 Brevísima historia
- 3 ¿Porqué Python ?
- 4 Diferencias entre 2.x y 3.x
- 5 Hola Mundo Pythonico!
- 6 Referencias

## Temario **Curso – Taller**

- ➊ Introducción
- ➋ Ponle ambiente a tu mundo
- ➌ Pythonico es más bonito
- ➍ Python científico con baterías incluidas
- ➎ Python con clase
- ➏ Tópicos especiales
  - ➊ Introducción a las interfaces gráficas
  - ➋ Proyecto final
  - ➌ Github



<https://www.python.org/>

Versión 3.6.x

Moodle  
Luis M.  
Mario A.

<https://gmc.geofisica.unam.mx/cursos>  
[luiggi@igeofisica.unam.mx](mailto:luiggi@igeofisica.unam.mx)  
[mizar0.30@gmail.com](mailto:mizar0.30@gmail.com)

Haremos lo mínimo indispensable...

Haremos lo mínimo indispensable...  
para que Uds. hagan lo máximo posible!

Haremos lo mínimo indispensable...  
para que Uds. hagan lo máximo posible!

### Principio de Parapeto 80-20

*“La gente en la sociedad se divide naturalmente entre los «pocos de mucho» y los «muchos de poco» ”*

Vilfredo Pareto (1845-1923)  
ing., soc., eco y fil. parisino.

# Breve Historia del Python

- 1 Acerca del curso
- 2 Brevísima historia**
- 3 ¿Porqué Python ?
- 4 Diferencias entre 2.x y 3.x
- 5 Hola Mundo Pythonico!
- 6 Referencias

“Over six years ago, in December **1989**, I was looking for a ‘hobby’ programming project that would keep me occupied during the week around Christmas. My office ... would be closed, but I had a home 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 hackers. I chose Python as a working title for the project, being in a slightly irreverent mood (and a big fan of Monty Python’s Flying Circus).”

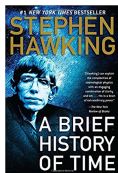
**Guido van Rossum**





“Over six years ago, in December **1989**, I was looking for a ‘hobby’ programming project that would keep me occupied during the week around Christmas. My office ... would be closed, but I had a home 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 hackers. I chose Python as a working title for the project, being in a slightly irreverent mood (and a big fan of Monty Python’s Flying Circus).”

**Guido van Rossum**



En **1988** se publicó el libro “*A brief history of time: From the Big Bang to Black Holes*”, **Stephen Hawking**  
(Breve Historia del Tiempo)

# Propuesta de Van Rossum a DARPA<sup>1</sup>

*Computer Programming for Everybody (CP4E)* <sup>2</sup>

- 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.

---

<sup>1</sup>Defense Advanced Research Projects Agency, 1999

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

# Breve Historia del Python

- 1 Acerca del curso
- 2 Brevísima historia
- 3 ¿Porqué Python ?**
- 4 Diferencias entre 2.x y 3.x
- 5 Hola Mundo Pythonico!
- 6 Referencias



# Pero:

- ¿En qué nos basamos para creer que conocemos mejor el universo?, ¿Qué sabemos acerca de él y cómo hemos llegado a saberlo?, ¿Dónde surgió?, ¿A dónde va?, ¿Tuvo un principio?, ...

*"Breve historia del tiempo", Stephen Hawking*



# Pero:

- ¿En qué nos basamos para creer que conocemos mejor el universo?, ¿Qué sabemos acerca de él y cómo hemos llegado a saberlo?, ¿Dónde surgió?, ¿A dónde va?, ¿Tuvo un principio?, ...

*"Breve historia del tiempo"*, Stephen Hawking

# Será posible algún día:

- ¿Atravesar las paredes?, ¿Construir naves que superen la velocidad de la luz?, ¿Leer la mente de otras personas?, ¿Hacerse invisible?, ¿Mover objetos con la mente?, ...

*"Física de lo imposible"*, Michio Kaku



# Pero:

- ¿En qué nos basamos para creer que conocemos mejor el universo?, ¿Qué sabemos acerca de él y cómo hemos llegado a saberlo?, ¿Dónde surgió?, ¿A dónde va?, ¿Tuvo un principio?, ...

*"Breve historia del tiempo"*, Stephen Hawking

# Será posible algún día:

- ¿Atravesar las paredes?, ¿Construir naves que superen la velocidad de la luz?, ¿Leer la mente de otras personas?, ¿Hacerse invisible?, ¿Mover objetos con la mente?, ...

*"Física de lo imposible"*, Michio Kaku

- **¿Podremos tener una computadora en cada escritorio? (70s)**
- **¿Qué pasaría si cada usuario puede programar su propia computadora? (80s)**
- **Surgimiento del Open Source (80s  $\rightarrow$   $\infty$ )**
- **Nuevas herramientas y lenguajes de programación (90s)**

# ¿Porqué Python?

- Es un lenguaje de propósito general que ofrece distintos tipos de paradigmas de programación.
- Es intuitivo, fácil de aprender, portable y abierto.
- Existen cientos(miles) de bibliotecas (baterías incluidas)
- El segundo lenguaje más popular en [GitHub Octoverse 2017](#).
- Empresas de alto prestigio lo utilizan para programar todo tipo de aplicaciones y servicios.



## ¿Porqué Python?

- Es un lenguaje de propósito general que ofrece distintos tipos de paradigmas de programación.
- Es intuitivo, fácil de aprender, portable y abierto.
- Existen cientos(miles) de bibliotecas (baterías incluidas)
- El segundo lenguaje más popular en [GitHub Octoverse 2017](#).
- Empresas de alto prestigio lo utilizan para programar todo tipo de aplicaciones y servicios.

	Javascript	<b>Python</b>	Java	Ruby	PHP	C / C++	Matlab
Simple	×	✓	×	✓	×	×	✓
Eficiente	×	✓	×	?	?	✓	×
Barato	✓	✓	✓	✓	✓	✓	×
Genérico	×	✓	✓	×	×	✓	✓
Portable	✓	✓	✓	✓	×	✓	?

## ¿Porqué Python?

- Es un lenguaje de propósito general que ofrece distintos tipos de paradigmas de programación.
- Es intuitivo, fácil de aprender, portable y abierto.
- Existen cientos(miles) de bibliotecas (baterías incluidas)
- El segundo lenguaje más popular en [GitHub Octoverse 2017](#).
- Empresas de alto prestigio lo utilizan para programar todo tipo de aplicaciones y servicios.

	Javascript	<b>Python</b>	Java	Ruby	PHP	C / C++	Matlab
Simple	×	✓	×	✓	×	×	✓
Eficiente	×	✓	×	?	?	✓	×
Barato	✓	✓	✓	✓	✓	✓	×
Genérico	×	✓	✓	×	×	✓	✓
Portable	✓	✓	✓	✓	×	✓	?

**¡Es Bueno, Bonito y Barato!**

# Breve Historia del Python

- 1 Acerca del curso
- 2 Brevísima historia
- 3 ¿Porqué Python ?
- 4 Diferencias entre 2.x y 3.x**
- 5 Hola Mundo Pythonico!
- 6 Referencias

# Diferencias entre Python 2.x y Python 3.x

## Python 2.x

- Herencia
- 2.7 liberado en 2010
- No habrá versiones nuevas.

## Python 3.x

- Presente y el futuro.
- 3.0 liberado en 2008
- Desarrollo activo → 3.6.4

- ¿Qué versión debería usar?

# Diferencias entre Python 2.x y Python 3.x

## Python 2.x

- Herencia
- 2.7 liberado en 2010
- No habrá versiones nuevas.

## Python 3.x

- Presente y el futuro.
- 3.0 liberado en 2008
- Desarrollo activo → 3.6.4

- ¿Qué versión debería usar?
  - ▶ Si vas a iniciar un proyecto (incluyendo el aprendizaje del lenguaje), usa Python 3.x.

# Diferencias entre Python 2.x y Python 3.x

## Python 2.x

- Herencia
- 2.7 liberado en 2010
- No habrá versiones nuevas.

## Python 3.x

- Presente y el futuro.
- 3.0 liberado en 2008
- Desarrollo activo → 3.6.4

- ¿Qué versión debería usar?
  - ▶ Si vas a iniciar un proyecto (incluyendo el aprendizaje del lenguaje), usa Python 3.x.
  - ▶ Si puedes hacer todo lo que deseas con Python 3.x, entonces úsalo.

# Diferencias entre Python 2.x y Python 3.x

## Python 2.x

- Herencia
- 2.7 liberado en 2010
- No habrá versiones nuevas.

## Python 3.x

- Presente y el futuro.
- 3.0 liberado en 2008
- Desarrollo activo → 3.6.4

- ¿Qué versión debería usar?
  - ▶ Si vas a iniciar un proyecto (incluyendo el aprendizaje del lenguaje), usa Python 3.x.
  - ▶ Si puedes hacer todo lo que desees con Python 3.x, entonces úsalo.
  - ▶ Si hay alguna biblioteca en Python 2.x que no tenga soporte en 3.x, entonces:

# Diferencias entre Python 2.x y Python 3.x

## Python 2.x

- Herencia
- 2.7 liberado en 2010
- No habrá versiones nuevas.

## Python 3.x

- Presente y el futuro.
- 3.0 liberado en 2008
- Desarrollo activo → 3.6.4

- ¿Qué versión debería usar?

- ▶ Si vas a iniciar un proyecto (incluyendo el aprendizaje del lenguaje), usa Python 3.x.
- ▶ Si puedes hacer todo lo que deseas con Python 3.x, entonces úsalo.
- ▶ Si hay alguna biblioteca en Python 2.x que no tenga soporte en 3.x, entonces:
  - ★ Intenta portar esa dependencia a Python 3.x ☺



# Diferencias entre Python 2.x y Python 3.x

## Python 2.x

- Herencia
- 2.7 liberado en 2010
- No habrá versiones nuevas.

## Python 3.x

- Presente y el futuro.
- 3.0 liberado en 2008
- Desarrollo activo → 3.6.4

### • ¿Qué versión debería usar?

- ▶ Si vas a iniciar un proyecto (incluyendo el aprendizaje del lenguaje), usa Python 3.x.
- ▶ Si puedes hacer todo lo que deseas con Python 3.x, entonces úsalo.
- ▶ Si hay alguna biblioteca en Python 2.x que no tenga soporte en 3.x, entonces:
  - ★ Intenta portar esa dependencia a Python 3.x ☺
  - ★ Si no es posible, entonces usa Python 2.x ☹

# Diferencias entre Python 2.x y Python 3.x

## Python 2.x

- Herencia
- 2.7 liberado en 2010
- No habrá versiones nuevas.

## Python 3.x

- Presente y el futuro.
- 3.0 liberado en 2008
- Desarrollo activo → 3.6.4

- ¿Qué versión debería usar?
  - ▶ Si vas a iniciar un proyecto (incluyendo el aprendizaje del lenguaje), usa Python 3.x.
  - ▶ Si puedes hacer todo lo que deseas con Python 3.x, entonces úsalo.
  - ▶ Si hay alguna biblioteca en Python 2.x que no tenga soporte en 3.x, entonces:
    - ★ Intenta portar esa dependencia a Python 3.x ☺
    - ★ Si no es posible, entonces usa Python 2.x ☹
- Más información: <https://wiki.python.org/moin/Python2orPython3>

# Breve Historia del Python

- 1 Acerca del curso
- 2 Brevísima historia
- 3 ¿Porqué Python ?
- 4 Diferencias entre 2.x y 3.x
- 5 Hola Mundo Pythonico!**
- 6 Referencias

## Python interpreter

```
[Wed Jan 03] luiggi@ichXulub > python
Python 3.6.0 |Anaconda custom (64-bit)| (default, Dec 23 2016, 13:19:00)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 2+2
4
>>> 1/3
0.3333333333333333
>>> 1//3
0
>>> print('Hola Mundo Pythonico!')
Hola Mundo Pythonico!
>>>
```

## IPython Interactive Computing (ipython.org)

```
[Wed Jan 03] luiggi@ichXulub > ipython
Python 3.6.0 |Anaconda custom (64-bit)| (default, Dec 23 2016, 13:19:00)
Type "copyright", "credits" or "license" for more information.
```

```
IPython 5.1.0 -- An enhanced Interactive Python.
```

```
?          -> Introduction and overview of IPython's features.
```

```
%quickref -> Quick reference.
```

```
help       -> Python's own help system.
```

```
object?    -> Details about 'object', use 'object??' for extra details.
```

```
In [1]: 2**3
```

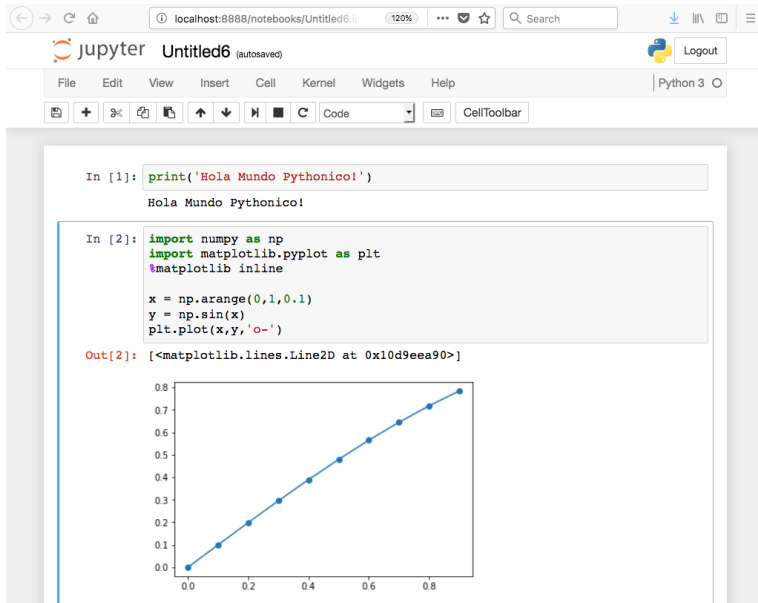
```
Out[1]: 8
```

```
In [2]: print('Hola Mundo Pythonico!')
```

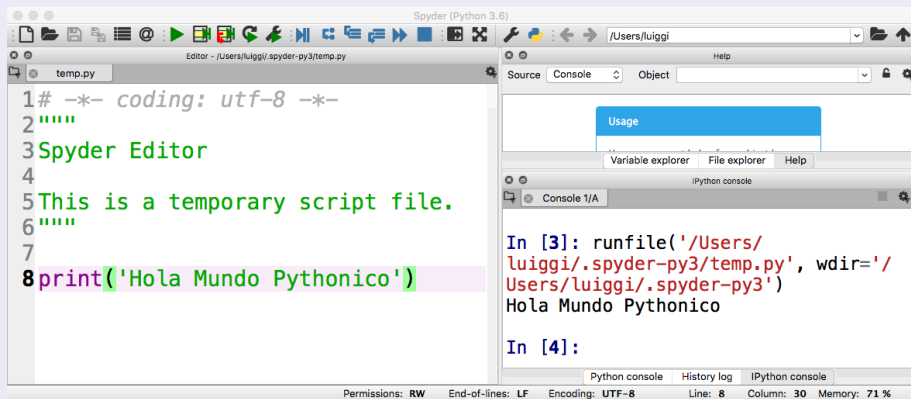
```
Hola Mundo Pythonico!
```

```
In [3]:
```

# jupyter notebook (jupyter.org)



# Spyder: Scientific PYTHON Development Environment



<https://pythonhosted.org/spyder/>

# Breve Historia del Python

- 1 Acerca del curso
- 2 Brevísima historia
- 3 ¿Porqué Python ?
- 4 Diferencias entre 2.x y 3.x
- 5 Hola Mundo Pythonico!
- 6 Referencias**





Anaconda: The Most Popular Python Data Science Platform.  
<https://www.anaconda.com/>



Python Software Foundation.  
<https://www.python.org/>



Python Documentation.  
<https://docs.python.org/>



The Hitchhiker's Guide to Python  
<http://docs.python-guide.org/>



Programa como un Pythonista: Python Idiomático  
<http://mundogeek.net/traducciones/python-idiomatico/>



Python 3 Basics Tutorial  
<https://www.gitbook.com/book/krother/python-3-basics-tutorial/details>



Interactive Python Tutorial  
<http://www.learnpython.org/>



Spyder is the Scientific PYthon Development EnviRonment:  
<https://pythonhosted.org/spyder/>



IPython Interactive Computing

<http://ipython.org/>



IPython Documentation

<https://ipython.readthedocs.io/>



Project Jupyter

<https://jupyter.org/>



*Python para todos*

Raúl González Duque

<http://mundogeek.net/tutorial-python/>.



*A primer on scientific programming with Python*

Hans Petter Langtangen (<http://hplgit.github.io/homepage/index.html>)

Springer-Verlag Berlin Heidelberg, 2012.



IPython: A System for Interactive Scientific Computing

Fernando Pérez & Brian E. Granger

*Computing in Science and Engineering*, Vol. 7, No. 3, May/Jun, 2007.



Scientific Python

Varios Autores

*Computing in Science and Engineering*, Vol. 13, No. 2, March/April, 2011.