

Módulo 1 – Fundamentos de Programación Python

# Generalidades del lenguaje Python

Ciencia de Datos

# Contenido

1. ¿Por qué Python?
2. Zen de Python.
3. Características del lenguaje Python.
4. Crecimiento de Python.
5. Webs de utilidad.
6. Entorno de trabajo.



# ¿Por qué Python?

Python es un lenguaje **libre, sencillo, legible** y con una curva de aprendizaje exponencial. El zen de Python es la **sencillez** y **entendibilidad**.



**Java**

**C++**

**Visual Basic**



**Python**



# Zen de Python

```
1 import this
2
```

The Zen of Python, by Tim Peters

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!

# Zen de Python (traducción)

El Zen de Python, por Tim Peters

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.

Disperso es mejor que denso.

La legibilidad cuenta.

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

Aunque la practicidad le gana a la pureza.

Los errores nunca deberían pasar en silencio.

A menos que se silencien explícitamente.

Ante la ambigüedad, rechaza la tentación de adivinar.

Debería haber una –y preferiblemente solo una– forma obvia de hacerlo.

Aunque esa forma puede no ser obvia al principio a menos que seas holandés.

Ahora es mejor que nunca.

Aunque nunca es a menudo mejor que \*ahora\* mismo.

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

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

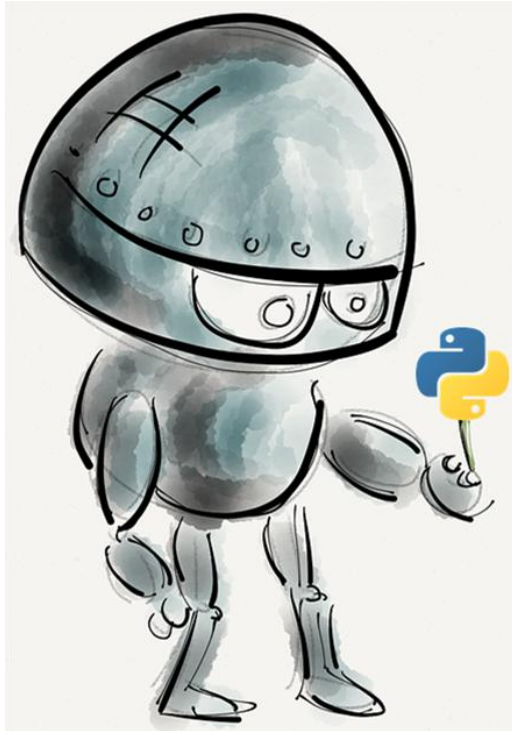
Los espacios de nombres son una gran idea – ¡hagamos más de esos!

# Características del Lenguaje Python

Python es un lenguaje de programación de **alto nivel**, de **propósito general**, **multiparadigma**, **interpretado**, de **tipado dinámico**, **multiplataforma**, cuya filosofía apunta a la **legibilidad de su código**.



# Múltiples propósitos



Python puede ser aplicado para **automatización** de tareas, **machine learning**, **big data**, **computación científica**, **desarrollo web**, **aplicaciones GUI**, **programación IoT**, entre otras.

# Multiparadigma

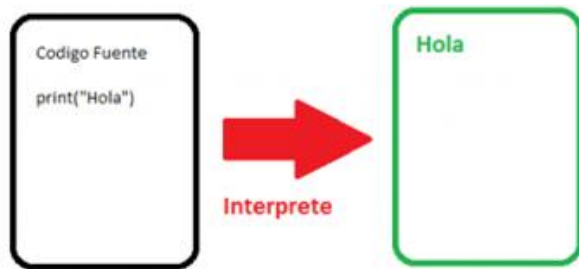


Python no obliga a utilizar un paradigma particular, permite varios estilos: programación **procedural**, programación **orientada a objetos**, programación **imperativa**, programación **funcional**.



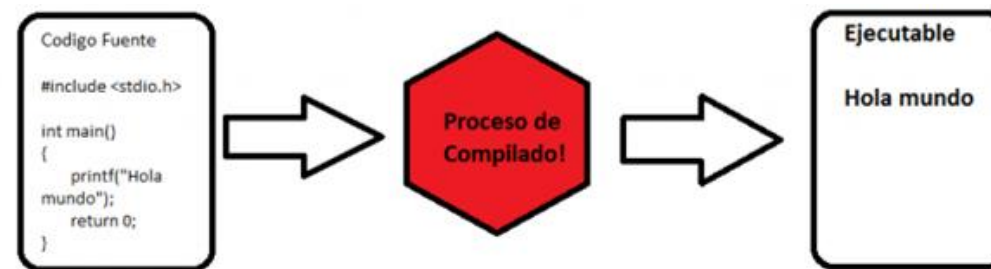
# Lenguaje Interpretado

## Lenguaje Interpretado



El código es traducido mediante un **intérprete** a medida que es necesario. Ej: “**Python**”, “Ruby”, “Javascript”, etc.

## Lenguaje Compilado



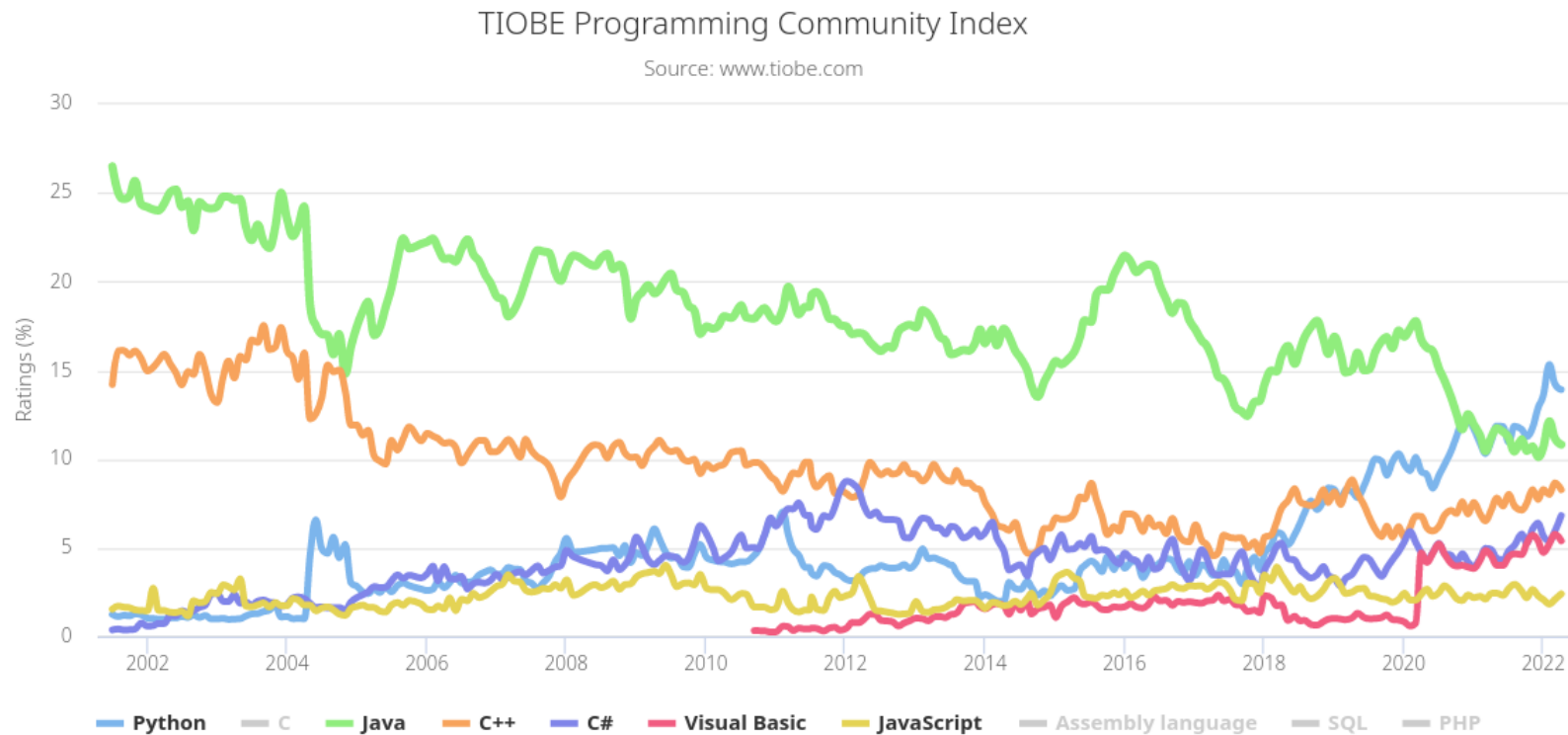
El código es traducido por completo de una sola vez mediante un proceso llamado **compilación** para ser ejecutado por un sistema predeterminado. Ej: “C”, “C++”, Java, etc.

# Una gran comunidad

Es un lenguaje de **código abierto**, con una curva de **aprendizaje alta**, soportado por una **gran comunidad de desarrolladores** alrededor del mundo, contribuyendo en el desarrollo de **librerías para múltiples propósitos**, contestando preguntas en los **foros** especializados, con muchos instructores construyendo maravillosos **cursos**.



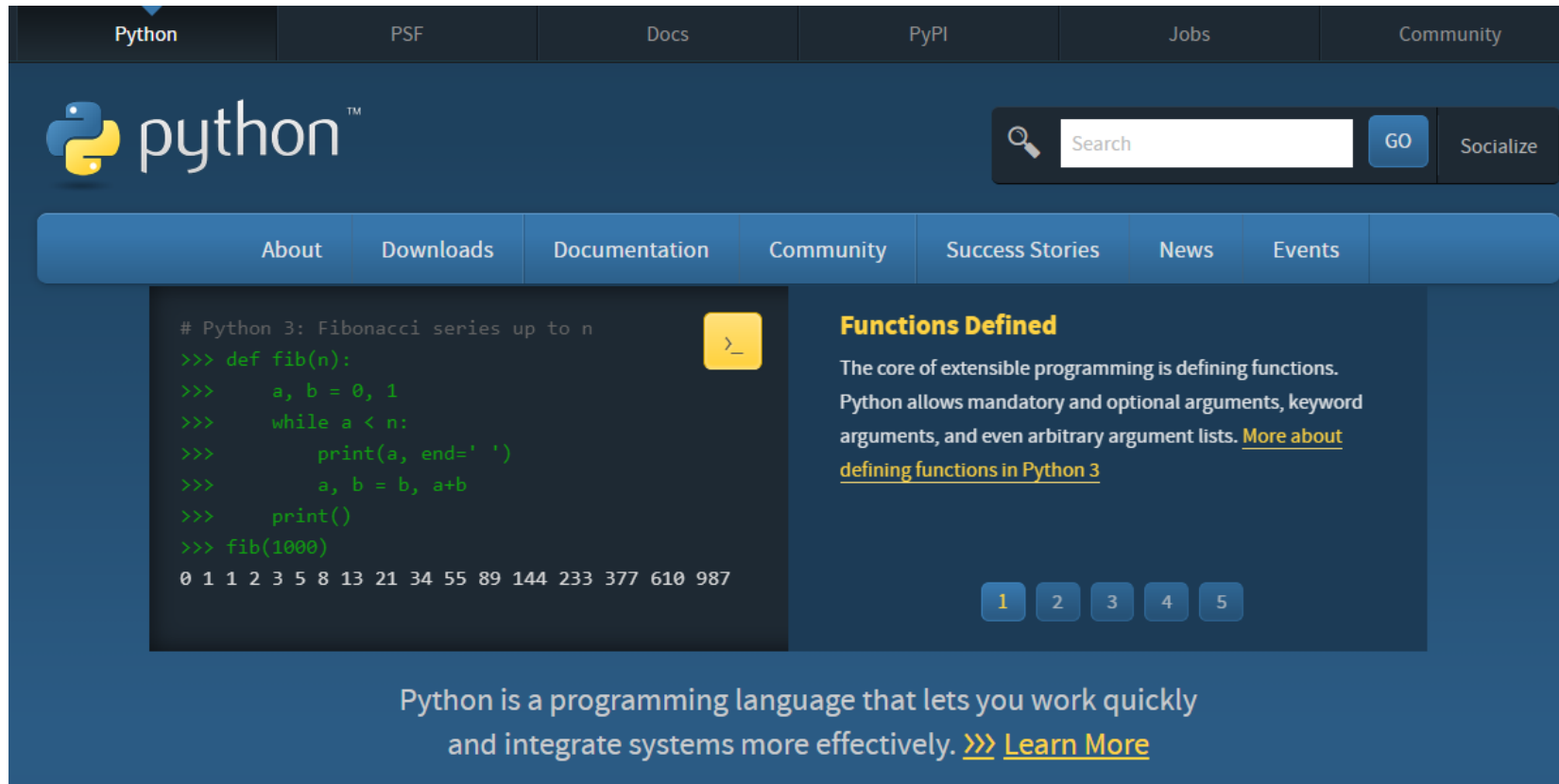
# Crecimiento de Python



Webs de utilidad



# Sitio Oficial



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

# Documentación Oficial

Python » English 3.9.7 Documentation »

**Download**

Download these documents

**Docs by version**

- Python 3.11 (in development)
- Python 3.10 (pre-release)
- Python 3.9 (stable)
- Python 3.8 (security-fixes)
- Python 3.7 (security-fixes)
- Python 3.6 (security-fixes)
- Python 3.5 (EOL)
- Python 2.7 (EOL)
- All versions

**Other resources**

- PEP Index
- Beginner's Guide
- Book List
- Audio/Visual Talks
- Python Developer's Guide

## Python 3.9.7 documentation

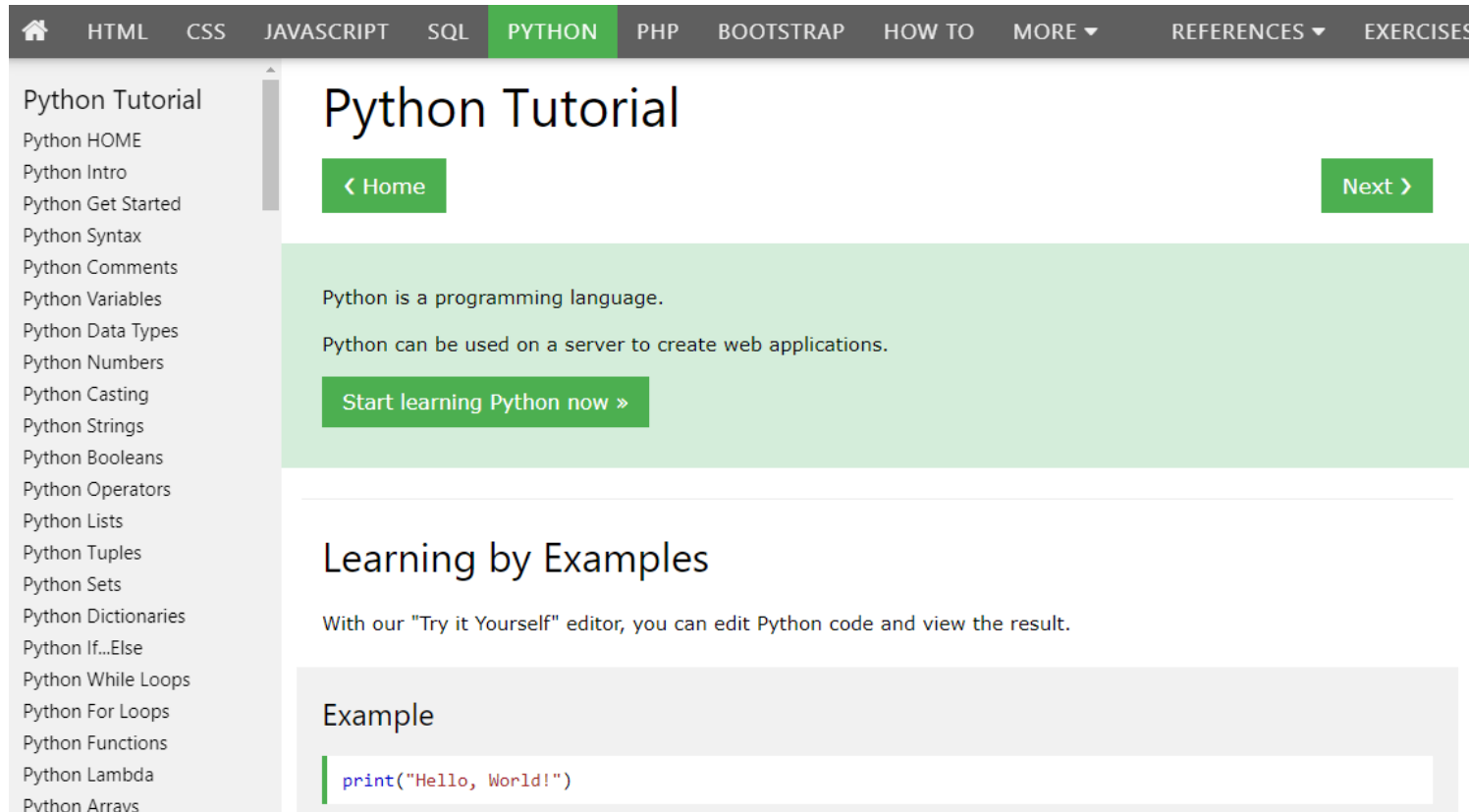
Welcome! This is the official documentation for Python 3.9.7.

**Parts of the documentation:**

- [What's new in Python 3.9?](#)  
*or all "What's new" documents since 2.0*
- [Tutorial](#)  
*start here*
- [Library Reference](#)  
*keep this under your pillow*
- [Language Reference](#)  
*describes syntax and language elements*
- [Python Setup and Usage](#)  
*how to use Python on different platforms*
- [Python HOWTOs](#)  
*in-depth documents on specific topics*
- [Installing Python Modules](#)  
*installing from the Python Package Index & other sources*
- [Distributing Python Modules](#)  
*publishing modules for installation by others*
- [Extending and Embedding](#)  
*tutorial for C/C++ programmers*
- [Python/C API](#)  
*reference for C/C++ programmers*
- [FAQs](#)  
*frequently asked questions (with answers!)*

<https://docs.python.org/3/>

# W3C Schools



The screenshot shows the W3C Schools Python Tutorial page. At the top is a navigation bar with links for HTML, CSS, JAVASCRIPT, SQL, PYTHON (highlighted), PHP, BOOTSTRAP, HOW TO, MORE, REFERENCES, and EXERCISES. On the left is a sidebar menu titled 'Python Tutorial' with a list of topics including Python HOME, Python Intro, Python Get Started, Python Syntax, Python Comments, Python Variables, Python Data Types, Python Numbers, Python Casting, Python Strings, Python Booleans, Python Operators, Python Lists, Python Tuples, Python Sets, Python Dictionaries, Python If...Else, Python While Loops, Python For Loops, Python Functions, Python Lambda, and Python Arrays. The main content area has the title 'Python Tutorial' and navigation buttons '< Home' and 'Next >'. Below the title is a green box containing the text 'Python is a programming language.' and 'Python can be used on a server to create web applications.', with a button 'Start learning Python now >'. Further down is a section titled 'Learning by Examples' with the text 'With our "Try it Yourself" editor, you can edit Python code and view the result.' Below this is an 'Example' section with a code editor showing the code `print("Hello, World!")`.

Python Tutorial

Python HOME  
Python Intro  
Python Get Started  
Python Syntax  
Python Comments  
Python Variables  
Python Data Types  
Python Numbers  
Python Casting  
Python Strings  
Python Booleans  
Python Operators  
Python Lists  
Python Tuples  
Python Sets  
Python Dictionaries  
Python If...Else  
Python While Loops  
Python For Loops  
Python Functions  
Python Lambda  
Python Arrays

## Python Tutorial

< Home

Next >

Python is a programming language.

Python can be used on a server to create web applications.

Start learning Python now >

### Learning by Examples

With our "Try it Yourself" editor, you can edit Python code and view the result.

#### Example

```
print("Hello, World!")
```

<https://www.w3schools.com/python/>

# Tutorial Learn Python

## Español

Bienvenidos a los tutoriales de [learnpython.org](https://www.learnpython.org) sobre Python.

Seas un programador experimentado o no, este sitio web es creado para cualquiera que desea aprender el lenguaje de programación Python.

La idea de crear la versión en español es tener a disposición este sitio a más interesado sin que el idioma sea una barrera.

Como primer paso es traducir al español todos los tutoriales que existen en la versión en inglés creando nuevos tutoriales adicionando el texto "(es)" el final del nombre para indicar que es la versión en español, toda ayuda es bienvenida!

Ahora sólo da click en el capítulo que quieras iniciar y sigue las instrucciones o [platica](#) con otros estudiantes de Python. Buena suerte!

Si quieres unirse al grupo de [Facebook](#) bienvenido!

## Indice

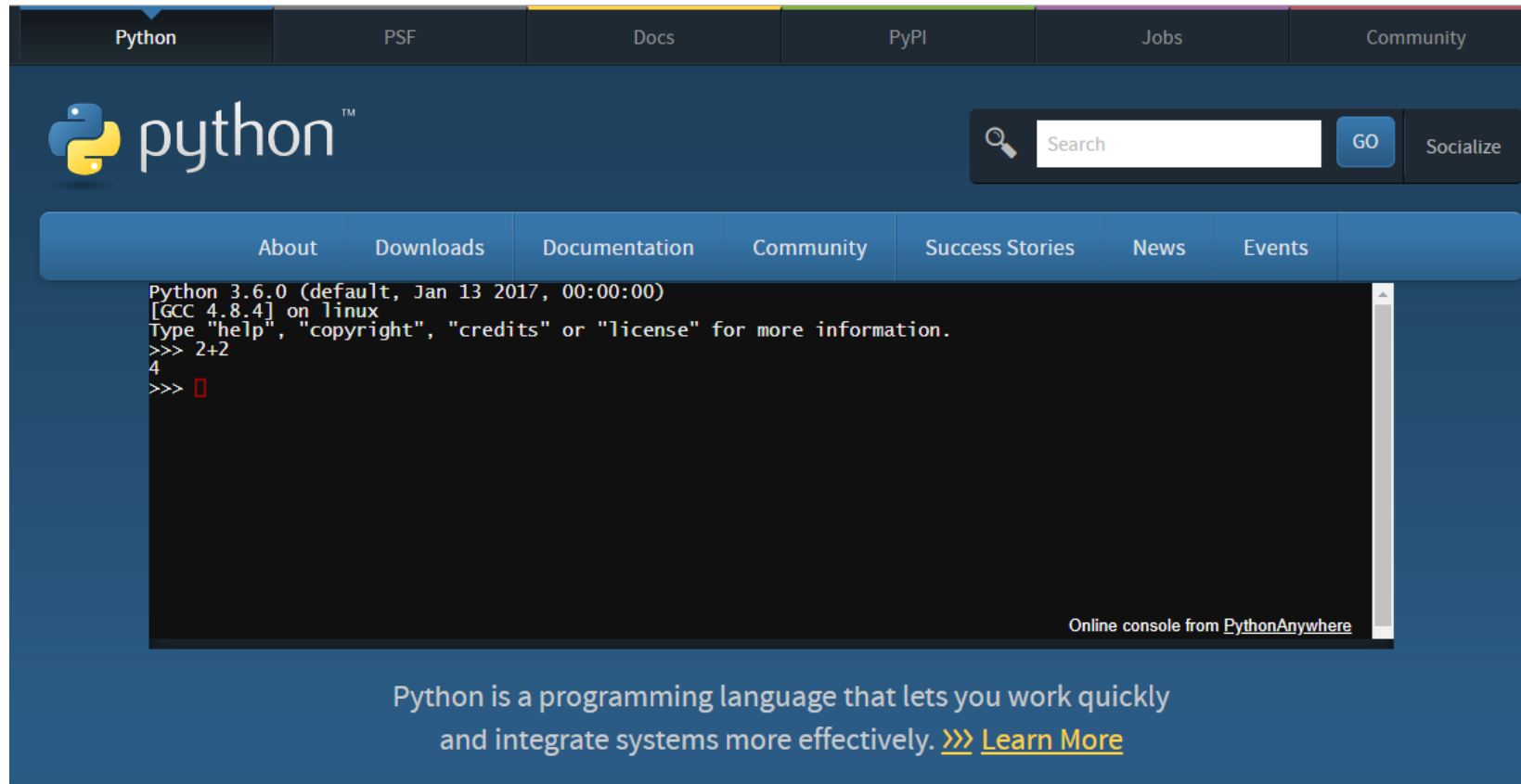
### Aprende las bases

- [Hola Mundo](#)
- [Variables y Tipos de datos](#)
- [Listas](#)
- [Operadores basicos](#)
- [Formato de texto](#)
- [Operaciones basicas con texto](#)

<https://www.learnpython.org/es/>




# Consola Online



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

# Python donde sea!

Send feedbackForumsHelpBlogPricing & signupLog in

 pythonanywhere

## Host, run, and code Python in the cloud!

Get started for free. Our basic plan gives you access to machines with a [full Python environment](#) already installed. You can develop and host your website or any other code directly from your browser without having to install software or manage your own server.

Need more power? Upgraded plans start at \$5/month.

Start running Python online in less than a minute! »

Watch our one-minute video »

Not convinced? [Read what our users are saying!](#)

pythonanywhereDashboardConsoles

All done! Your web app is now set up. Details below.

example.pythonanywhere.com

www.mydomain.com

www.myotherdomain.com

Add a new web app

Configuration for [www.myotherdomain.com](#)

Reload:

Reload [www.myotherdomain.com](#)

DNS setup:

How to point your domain at your website.

CNAME: [webapp-4.pythonanywhere.com](#)


Traffic:

How busy is your site?


Hits per month

Hits per day

<https://www.pythonanywhere.com/>

 KIBERNUM


# Jupyter iPython Notebooks

 [Install](#) [About Us](#) [Community](#) [Documentation](#) [NBViewer](#) [Widgets](#) [Blog](#)

## Try Jupyter


You can try Jupyter out right now, without installing anything. Select an example below and you will get a temporary Jupyter server just for you, running on [mybinder.org](#). If you like it, you can [install Jupyter](#) yourself.

### Try Jupyter with Python




A tutorial introducing basic features of Jupyter notebooks and the IPython kernel.

### Try JupyterLab



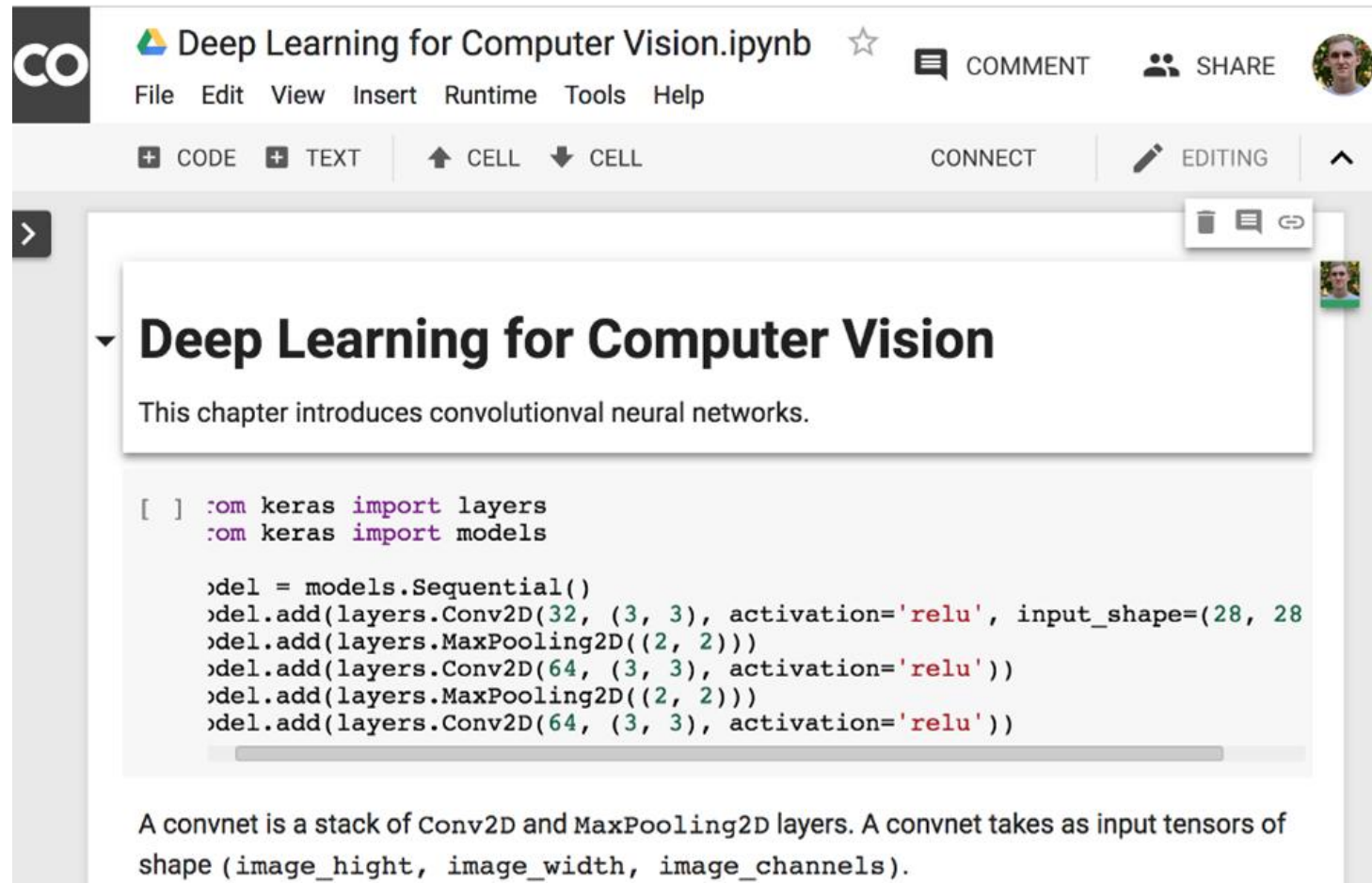
JupyterLab is the new interface for Jupyter notebooks and is ready for testing. Give it a try!

### Try Jupyter with Julia



A basic example of using Jupyter with Julia.

<https://jupyter.org/try>



The screenshot shows a Google Colab notebook interface. At the top, the title bar reads 'Deep Learning for Computer Vision.ipynb' with a star icon, a 'COMMENT' button, a 'SHARE' button, and a user profile picture. Below the title bar is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. A secondary toolbar contains '+ CODE', '+ TEXT', '↑ CELL', '↓ CELL', 'CONNECT', 'EDITING', and an upward arrow. The notebook content area features a large heading 'Deep Learning for Computer Vision' with a subtext 'This chapter introduces convolutional neural networks.' Below this is a code cell containing Python code for a Keras sequential model. The code defines a model with three layers: a Conv2D layer with 32 filters, a MaxPooling2D layer, and another Conv2D layer with 64 filters, all using 'relu' activation. A scrollbar is visible at the bottom of the code cell. Below the code cell, a paragraph explains that a convnet is a stack of Conv2D and MaxPooling2D layers and that it takes input tensors of shape (image\_height, image\_width, image\_channels).

**Deep Learning for Computer Vision**

This chapter introduces convolutional neural networks.

```
[ ] from keras import layers
    from keras import models

    model = models.Sequential()
    model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28)))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(64, (3, 3), activation='relu'))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(64, (3, 3), activation='relu'))
```

A convnet is a stack of Conv2D and MaxPooling2D layers. A convnet takes as input tensors of shape (image\_height, image\_width, image\_channels).

<https://colab.research.google.com/>



Entorno de trabajo



Products ▼

Pricing

Solutions ▼

Resources ▼

Blog

Company ▼

Get Started

# Data science technology for human sensemaking.

A movement that brings together millions of data science practitioners,  
data-driven enterprises, and the open source community.

Get Started

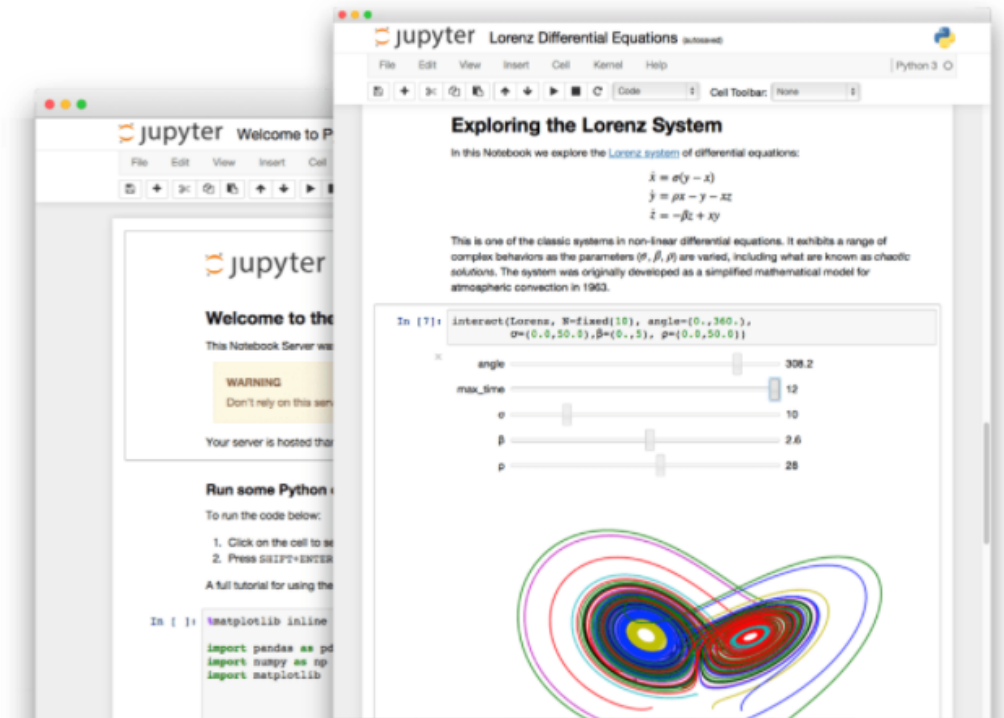


<https://www.anaconda.com/>

# Jupyter Notebooks



[Install](#) [About Us](#) [Community](#) [Events](#) [Documentation](#) [NBViewer](#) [JupyterHub](#) [Widgets](#) [Blog](#) [Security](#)



## The Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

[Try it in your browser](#)

[Install the Notebook](#)

<https://jupyter.org/>

# Spyder

[HOME](#)

## OVERVIEW

## COMPONENTS

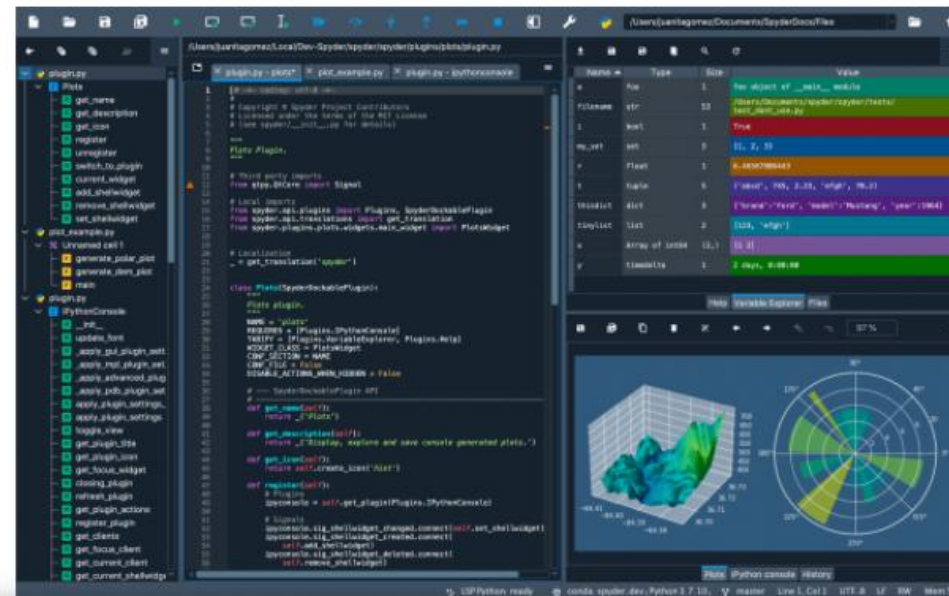
## PLUGINS

DOWNLOAD

[DONATE](#)

BLOG

DOCS



**OPEN CHAT**

<https://www.spyder-ide.org/>





# Dudas y consultas

Fin Presentación