

Desenvolver con Python. Editores.

Objetivos

- Revisar os IDEs por defecto para Python en Raspbian
- Valorizar Spyder como IDE
- Instalar Spyder dende o terminal
- Revalidar o cinto amarelo do NINJA do terminal

Desenvolver con Python. Editores.

Contidos

- IDLE
- Geany
- Spyder
- 'Ola mundo' en Python sobre Raspbian

Desenvolver con Python. Editores.

- Python é unha linguaxe de programación fácil de aprender, moi popular e á vez potente.
- Emprégase en entornos profesionais
- Sintaxe moi parecida á de C, con algunhas características que a fan máis amigable
- Ven con baterías incluídas
- Guido van Rossum desenvolveuna para o ensino de programación.
- A Raspberry Foundation escolleuna como linguaxe estándar para a Raspi

Desenvolver con Python. Editores.

- Python sofreu unha gran mudanza dende as versións 2.x para as 3.x
- Actualmente a versión máis extendida é a 3.x
- Pode acontecer que algunhas funcións das 3.x non se executen na 2.x.
- Nos empregaremos a 3.5 que é a que ven por defecto na Raspbian actualmente (podemos actualizar para a 3.7 empregando apt-get)
- Os editores por defecto nesta versión Raspbian son Geany e IDLE

Desenvolver con Python. Editores.

The image shows a screenshot of a Raspberry Pi desktop environment accessed via Remote Viewer. The desktop background is a landscape with a sunset over a field. The desktop environment is running Raspbian. The top panel shows the system tray with icons for Bluetooth, Wi-Fi, and audio, along with the system clock showing 12:50. The taskbar contains several open windows: 'Sen título - Geany', 'Python 3.5.3 Shell', and another instance of 'Sen título - Geany'. A menu is open over the desktop, listing various applications under the 'Programación' (Programming) category. The menu items are: Geany, Python 3 (IDLE), Scratch, Scratch 2, Sense HAT Emulator, Sonic Pi, Thonny Python IDE, and Thonny (Simple Mode). Other categories in the menu include 'Oficina' (Office), 'Internet', 'Xogos' (Games), 'Accesorios' (Accessories), 'Help', 'Preferencias' (Preferences), 'Run...', and 'Shutdown...'. In the bottom right corner, there is a notification area with a clock showing 12:49:30 and 12:49:31, and buttons for 'Estado' (Status), 'Compilador' (Compiler), and 'Mensaxes' (Messages). A 'Python 3.5.3 Shell' terminal window is open in the foreground, displaying the following text:

```
Python 3.5.3 (default, Sep 27 2018, 17:25:39)
[GCC 6.3.0 20170516] on linux
Type "copyright", "credits" or "license()" for more information.
>>>
```

Desenvolver con Python. Editores.

- Para empezar a desenvolver a miña escolla é Spyder
- É un editor que inclúe moitas opcións útiles para os rapaces que empezan
- Permite executar paso a paso un script e revisar o estado das variables
- Ten un 'aire' co IDE de Matlab
- Inicialmente desenvolvido pola comunidade python argentina:

<http://www.python.org.ar/>

- Para instalar:

```
sudo apt-get install spyder3
```

Desenvolver con Python. Spyder.

- En wikipedia:
https://en.wikipedia.org/wiki/Spyder_%28software%29
- Páxina oficial:
<https://www.spyder-ide.org/>

The screenshot displays the Spyder Python IDE interface. The main window is divided into several panes:

- Project Explorer:** Shows the file structure of the current project, including folders like 'Data', 'Images', and 'Scripts', and files like 'interpolation.py'.
- Editor:** Contains the Python code for 'interpolation.py'. The code includes imports for NumPy and SciPy, data generation, spline fitting, and plotting.
- Variable Explorer:** Displays the state of variables in the current namespace. It shows a table with columns for Name, Type, Size, and Value. Variables include 'array_int8', 'array_uint32', 'bars', 'df', 'filename', 'list_test', 'mnews', 'r', 'radius', 'region', 'rgb', and 'series'.
- Console:** Shows the output of the code execution, including the execution of 'plt.show()' and the rendering of a 3D surface plot and a 2D polar plot.

```
7 import pylab
8 from numpy import cos, linspace, pi, sin, random
9 from scipy.interpolate import splprep, splev
10
11 #%% Generate data for analysis
12
13 # Make ascending spiral in 3-space
14 t = linspace(0, 1.75 * 2 * pi, 100)
15
16 x = sin(t)
17 y = cos(t)
18 z = t
19
20 # Add noise
21 x += random.normal(scale=0.1, size=x.shape)
22 y += random.normal(scale=0.1, size=y.shape)
23 z += random.normal(scale=0.1, size=z.shape)
24
25
26 #%% Perform calculations
27
28 # Spline parameters
29 smoothness = 3.0 # Smoothness parameter
30 k_param = 2 # Spline order
31 nests = -1 # Estimate of number of knots needed (-1 = maximal)
32
33 # Find the best points
34 knot_points, u = splprep([x, y, z], s=smoothness, k=k_param, nests=-1)
35
36 # Evaluate spline, including interpolated points
37 new_x, new_y, new_z = splev(linspace(0, 1, 400), knot_points)
38
39
40 #%% Plot results
41
42 # TODO: Rewrite to avoid code smell
43 pylab.subplot(2, 2, 1)
44 data = pylab.plot(x, y, "bo", label="Data with X-Y Cross Section")
45 fit = pylab.plot(new_x, new_y, "r-", label="Fit with X-Y Cross Section")
46 pylab.legend()
47 pylab.xlabel("x")
48 pylab.ylabel("y")
49
50 pylab.subplot(2, 2, 2)
51 data = pylab.plot(x, z, "bo", label="Data with X-Z Cross Section")
52 fit = pylab.plot(new_x, new_z, "r-", label="Fit with X-Z Cross Section")
53 pylab.legend()
54 pylab.xlabel("x")
```

Desenvolver con Python. Spyder.

The image shows a screenshot of the Spyder Python IDE running on a Raspberry Pi desktop. The window title is "pi's X desktop (raspberrypi:1) (1) - Remote Viewer". The interface includes a menu bar (File, View, Send key, Help), a toolbar, and a main workspace divided into several panels:

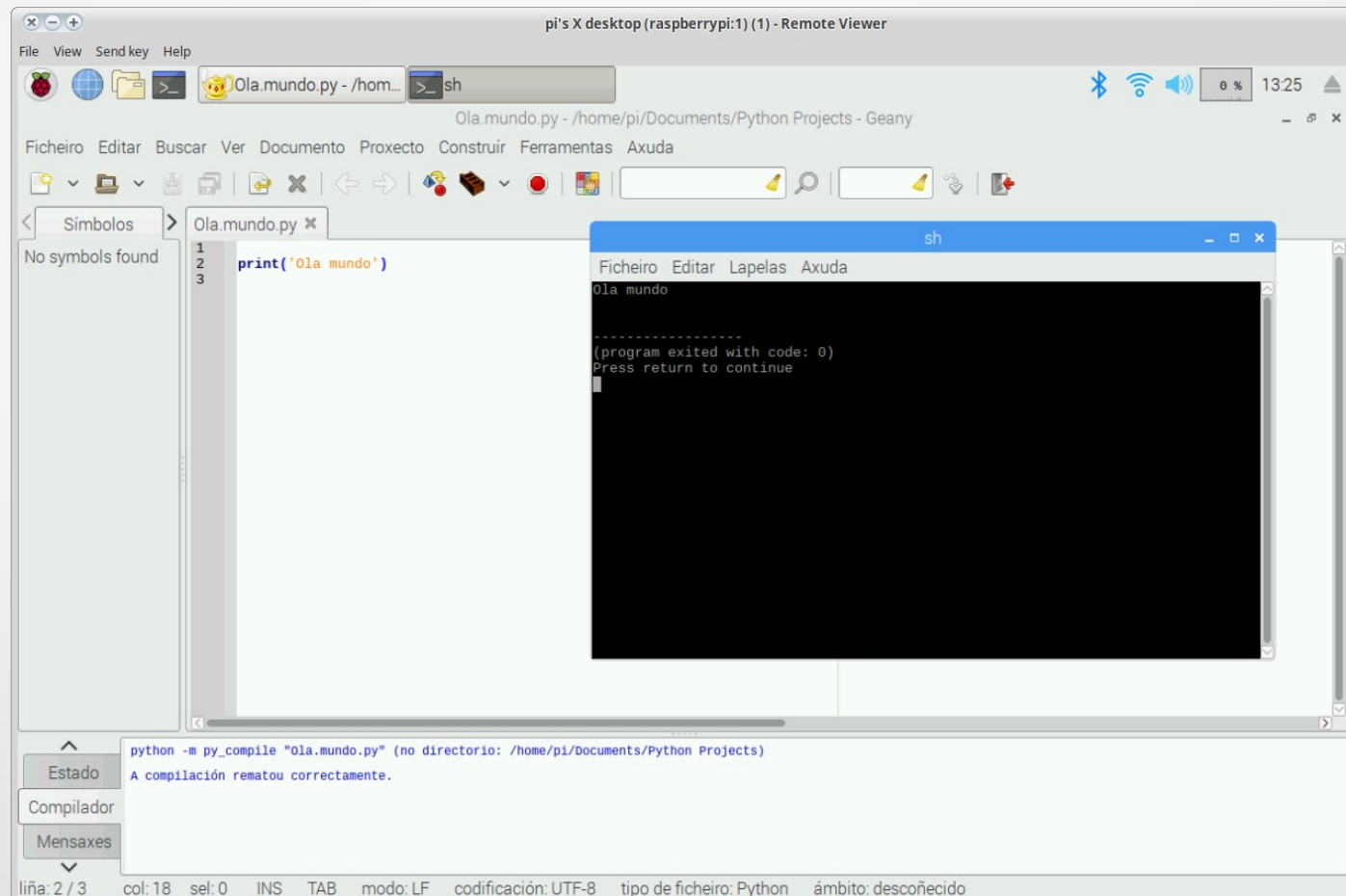
- Editor:** Displays a file named "temp.py" with the following code:

```
1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4
5 This is a temporary script file.
6 """
7
8
```
- Help:** A "Usage" dialog box is open, providing instructions on how to use the help system. It states: "Here you can get help of any object by pressing **Ctrl+H** in front of it, either on the Editor or the Console." It also mentions that help can be shown automatically after writing a left parenthesis next to an object, and provides a link to the "tutorial" for new users.
- Variable explorer, File explorer, Help:** These panels are visible at the bottom right of the workspace.
- IPython console:** A terminal window titled "Console 1/A" showing the output of the Python interpreter. It displays the Python version (3.5.3), the IPython version (5.1.0), and a list of available commands: "?", "%quickref", "help", and "object?".

At the bottom of the window, a status bar shows the following information: Permissions: RW, End-of-lines: LF, Encoding: UTF-8, Line: 8, Column: 1, Memory: 53 %.

Desenvolver con Python. Ola mundo.

- Non podemos empezar a desenvolver con Python sen inauguralo co consabido 'Ola mundo'.



The screenshot shows a remote viewer of a Raspberry Pi desktop. The main window is a code editor named 'Geany' with the file 'Ola.mundo.py' open. The code in the editor is:

```
1  
2 print('Ola mundo')  
3
```

A terminal window titled 'sh' is open in the foreground, showing the output of the program:

```
Ola mundo  
  
-----  
(program exited with code: 0)  
Press return to continue
```

At the bottom of the screen, a status bar shows the command used to compile the script:

```
python -m py_compile "ola.mundo.py" (no directorio: /home/pi/Documents/Python Projects)  
A compilación rematou correctamente.
```

The status bar also displays: liña: 2 / 3 col: 18 sel: 0 INS TAB modo: LF codificación: UTF-8 tipo de ficheiro: Python ámbito: descoñecido