

# ★ Python 3 para impacientes ★

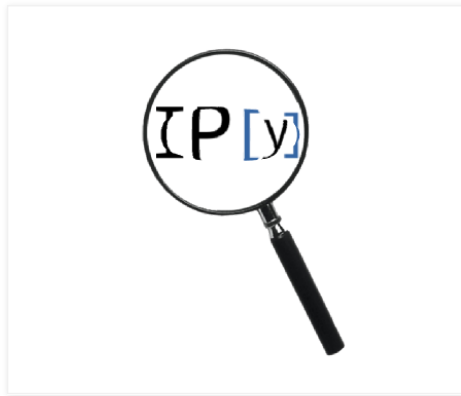


"Simple es mejor que complejo" (Tim Peters)

Python	IPython	EasyGUI	Tkinter	JupyterLab	Numpy
--------	---------	---------	---------	------------	-------

viernes, 8 de agosto de 2014

## Explorar objetos en IPython



Desde el primer momento la ayuda de IPython nos muestra cómo consultar información de un objeto. Lo más frecuente es escribir el nombre del objeto precedido o seguido de uno o dos signos de interrogación:

: **objeto?** # Mostrar información de un objeto  
: **objeto??** # Mostrar además código fuente si es posible

Sin embargo, para obtener información específica de un objeto podemos utilizar las funciones mágicas: **%pdoc**, **%pdef**, **%psource** y **%pfile**.

### Mostrar documentación

Mostrar cadena de documentación o texto de ayuda de un objeto: **%pdoc objeto**

```
: import sys
: %pdoc sys
Class docstring:
    This module provides access to some objects used or maintained by the
    interpreter and to functions that interact strongly with the interpreter.

Dynamic objects:

argv -- command line arguments; argv[0] is the script pathname if known
path -- module search path; path[0] is the script directory, else "
modules -- dictionary of loaded modules
...
...
```

### Mostrar línea de definición

Mostrar la línea de definición de un objeto que se pueda invocar: **%pdef objeto** (Si el objeto es una clase mostrará la información del constructor).

```
: %pdef alias
alias(self, parameter_s="")

: %pdef writefile
writefile(self, line, cell)

: %pdef sys
Object is not callable.
```

### Mostrar código fuente

Buscar

#### Python para impacientes

[Python](#)  
[IPython](#)  
[EasyGUI](#)  
[Tkinter](#)  
[JupyterLab](#)  
[Numpy](#)

#### Anexos

[Guía urgente de MySQL](#)  
[Guía rápida de SQLite3](#)

#### Entradas + populares

##### [Dar color a las salidas en la consola](#)

En Python para dar color a las salidas en la consola (o en la terminal de texto) existen varias posibilidades. Hay un método basado ...

##### [Instalación de Python, paso a paso](#)

Instalación de Python 3.6 A finales de 2016 se produjo el lanzamiento de Python 3.6 . El propósito de esta entrada es mostrar, pas...

##### [Variables de control en Tkinter](#)

Variables de control Las variables de control son objetos especiales que se asocian a los widgets para almacenar sus valore...

##### [Añadir, consultar, modificar y suprimir elementos en Numpy](#)

Acceder a los elementos de un array. [], []. ... Acceder a un elemento de un array. Para acceder a un elemento se utiliz...

##### [Cálculo con arrays Numpy](#)

Numpy ofrece todo lo necesario para obtener un buen rendimiento cuando se trata de hacer cálculos con arrays. Por como está concebido...

##### [Operaciones con fechas y horas. Calendarios](#)

Los módulos datetime y calendar amplían las posibilidades del módulo time que provee funciones para manipular expresiones de ti...

##### [Tkinter: interfaces gráficas en Python](#)

Introducción Con Python hay muchas posibilidades para programar una interfaz gráfica de usuario ( GUI ) pero Tkinter es fácil d...

##### [Convertir, copiar, ordenar, unir y dividir arrays Numpy](#)

Esta entrada trata sobre algunos métodos que se utilizan en Numpy para convertir listas en arrays y viceversa; para copiar arrays d...

##### [Tkinter: Tipos de ventanas](#)

Ventanas de aplicación y de diálogos En la entrada anterior tratamos los distintos gestores de geometría que se utilizan para di...

##### [El módulo random](#)

El módulo random de la librería estándar de Python incluye un conjunto de funciones

Para mostrar si es posible el código fuente de un objeto: **%psource objeto**

```
: %psource lsmagic
@line_magic
def lsmagic(self, parameter_s=""):
    """List currently available magic functions."""
    return MagicsDisplay(self.shell.magics_manager)
```

### Mostrar archivo de código fuente

Para mostrar el archivo de código fuente completo en el que se define un objeto, abriéndolo en la línea donde empieza su definición: **%pfile objeto**

```
: %pfile lsmagic

"""Implementation of basic magic functions.
"""
#-----
# Copyright (c) 2012 The IPython Development Team.
#
# Distributed under the terms of the Modified BSD License.
#
# The full license is in the file COPYING.txt, distributed with this software.
#-----

#-----
# Imports
#-----

# Stdlib
import io
import json
import sys
from pprint import pformat
...
...
```

### Mostrar información de los identificadores: %who / %whos

Las funciones mágicas **%who** / **%whos** proporcionan información sobre identificadores (de variables, listas, módulos, etc.) que se han definido o cargado de forma interactiva en una sesión de trabajo (no sobre lo que se ha definido o cargado de los archivos de configuración).

**%who** sólo visualizará la lista actual de identificadores mientras que **%whos** ampliará la información mostrando una tabla con algunos detalles básicos de cada identificador.

```
: %who
cadena  cadena2  iva  lista  math  os  sys

: %whos
Variable Type  Data/Info
-----
cadena  str    Andalucía
cadena2 bytes   b'Andalu\x1a'
iva     int    22
lista   list   n=11
math    module <module 'math' (built-in)>
os      module <module 'os' from '/usr/lib/python3.4/os.py'>
sys     module <module 'sys' (built-in)>
```

### Variables de entorno

Las variables de entorno pueden capturarse en una sesión de trabajo (de una vez) con la función mágica **%env**. Esta función devuelve un diccionario que contiene todas las variables de entorno con sus valores. Para acceder al valor de una variable utilizaremos como clave su propio nombre.

```
: variables = %env
: variables?      # Es un diccionario con las variables de entorno
Type: dict
...
: variables
{'CLICOLOR': '1',
 'CLUTTER_IM_MODULE': 'xim',
 'DBUS_SESSION_BUS_ADDRESS': 'unix:abstract=/tmp/dbus-cdk72jbhpm',
 'DEFAULTS_PATH': '/usr/share/gconf/xubuntu.default.path',
```

que permiten obtener de distintos modos números a...

#### Archivo

agosto 2014 (15) ▼

#### python.org



#### pypi.org



#### Sitios

- [ActivePython](#)
- [Anaconda](#)
- [Bpython](#)
- [Django](#)
- [Flask](#)
- [Ipython](#)
- [IronPython](#)
- [Matplotlib](#)
- [MicroPython](#)
- [Numpy](#)
- [Pandas](#)
- [Pillow](#)
- [PortablePython](#)
- [PyBrain](#)
- [PyCharm](#)
- [PyDev](#)
- [PyGame](#)
- [Pypi](#)
- [PyPy](#)
- [Pyramid](#)
- [Python.org](#)
- [PyTorch](#)
- [SciPy.org](#)
- [Spyder](#)
- [Tensorflow](#)
- [TurboGears](#)

```
'DESKTOP_SESSION': 'xubuntu',  
'DISPLAY': ':0.0',  
'GDMSESSION': 'xubuntu',  
'GDM_LANG': 'es_ES',  
etc...
```

```
: variables["LOGNAME"]  
'antonio'
```

```
: variables["SHELL"]  
'/bin/bash'
```

[Ir al índice del tutorial de IPython](#)

Publicado por Pherkad en [0:13](#)



Etiquetas: [IPython](#), [Jupyter](#)

[Entrada más reciente](#)

[Inicio](#)

[Entrada antigua](#)

2014-2020 | Alejandro Suárez Lamadrid y Antonio Suárez Jiménez, Andalucía - España  
. Tema Sencillo. Con la tecnología de [Blogger](#).