

# ★ Python 3 para impacientes ★

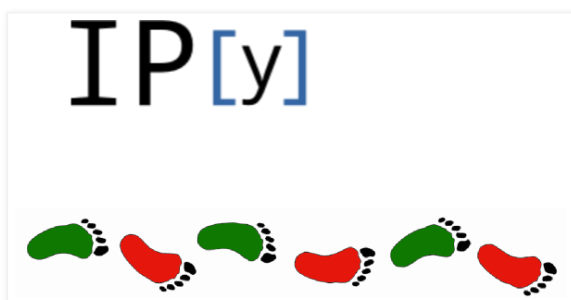


"Simple es mejor que complejo" (Tim Peters)

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

jueves, 7 de agosto de 2014

## Historial de entradas y salidas en IPython



Durante una sesión de trabajo IPython almacena tanto los comandos introducidos en las líneas de entrada "In" como los resultados que se originan en las líneas de salida "Out".

Como ya tratamos en el capítulo de "[Facilidades para la escritura](#)" con las teclas de **[Fecha arriba]** y **[Flecha abajo]** accedemos al historial de entradas asociado al usuario actual.

### Accediendo al historial de la sesión actual

Mientras trabajamos IPython va numerando secuencialmente las entradas "In" y las salidas "Out". Debemos tener en cuenta que "In" es una lista y "Out" un diccionario Python y ambos objetos podemos referenciarlos por su nombre y un número de secuencia: **In[Número]** y **Out[Número]**.

Ejemplos:

```
In [1]: 1 + 3
Out[1]: 4
```

```
In [2]: Out[1]
4
```

```
In [3]: In[1]
1 + 3
```

### Otras modos de acceder al historial:

Acceder al valor de la última, penúltima y antepenúltima entrada: `_i`, `_ii`, `_iii`, respectivamente:

```
In [4]: _iii
Out[4]: '1 + 3'
```

Mostrar todas las entradas del historial de la sesión actual: `In` ó `_ih`

```
In [5]: In
Out[5]: ['1 + 3', 'Out[1]', 'In[1]', '_iii', 'In']
```

Mostrar una entrada por su número: `_iNúmero` ó `_ih[Número]`

```
In [6]: _i4
Out[6]: '_iii'
```

Acceder al valor de la última, penúltima y antepenúltima salida, introducir uno, dos o tres guiones bajos (`_`, `__`, `___`), respectivamente:

```
In [7]: ___
Out[7]: ['1 + 3', 'Out[1]', 'In[1]', '_iii', 'In', '_i4', '___']
```

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

Mostrar una salida por su número: `_Número`

```
In [8]: _6
Out[8]: 'iii'
```

Mostrar todas las salidas del historial de la sesión actual: `Out` ó `_oh`

```
In [9]: Out
{1: 4,
 2: 4,
 3: '1 + 3',
 4: '1 + 3',
 5: ['1 + 3', 'Out[1]', 'In[1]', 'iii', 'In', '_i4', '_, '6', 'Out'],
 6: 'iii',
 7: ['1 + 3', 'Out[1]', 'In[1]', 'iii', 'In', '_i4', '_, '6', 'Out'],
 8: 'iii'}
```

### Accediendo al historial de cualquier sesión

La función mágica `%history` se utiliza también para mostrar el historial de entradas y salidas, permitiéndonos acceder a los historiales de sesiones anteriores. A continuación, se muestran varios ejemplos de uso.

```
: %hist o %history [Inicio-Fin] # Mostrar entradas desde inicio a fin
: %hist o %history -g cadena # Mostrar entradas que contengan cadena

: %hist -o ó %history -o [Inicio-Fin] # Mostrar entradas y salidas.
: %hist 1-10 -pof file.txt # Guardar historia en archivo. (-p) añade prompt
: %hist 20/1-10 # Listar 10 entradas de la sesión 20
: %hist ~2/1-10 # Listar 10 entradas de la penúltima sesión
: %hist ~2/1-10 -n # Listar 10 entradas de la penultima sesión numerando
```

El historial de entrada de sesiones anteriores se almacena en un base de datos pero es posible configurar IPython para que guarde también el historial de salida.

### Ejecutar entradas anteriores con `%rerun` y `%macro`

Se pueden ejecutar varias entradas anteriores mediante el uso de las funciones mágicas `%rerun` y `%macro`.

```
: %rerun 1-4 # ejecutar entradas de la 1 a la 4
: %rerun 4 # ejecutar entrada número 4

: %macro calcula 1-4 1-2 # crear macro "calcula" con entradas 1 a 4 y 1 a 2
: calcula # ejecutará la macro
: print(calcula) # mostrará el contenido de la macro "calcula"
```

Varias funciones mágicas pueden utilizar el historial de entrada: `%edit`, `%rerun`, `%recall`, `%macro`, `%save` y `%pastebin`.

### Editar una entrada anterior con `%recall` o `%rep`

Con las funciones mágicas `%recall` o `%rep` podemos editar una entrada anterior antes de su ejecución.

```
In [10]: %recall 1 # Editar la entrada número 1
In [10]: 1 + 3 | # Una vez modificada aceptar los cambios con [Enter]
```

### La base de datos del historial

El historial se guarda en una base de datos del tipo sqlite que podemos explorar con herramientas como **SQLite Database Browser** o **SQLite Manager**.

Para conocer la ubicación de la base de datos del historial:

```
In [11]: get_ipython().history_manager.hist_file
Out[11]: '/home/usuario/.ipython/profile_default/history.sqlite'
```

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

Publicado por Pherkad en [12:45](#)



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

[Entrada más reciente](#)

[Inicio](#)

[Entrada antigua](#)

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](#)

