

★ Python 3 para impacientes ★

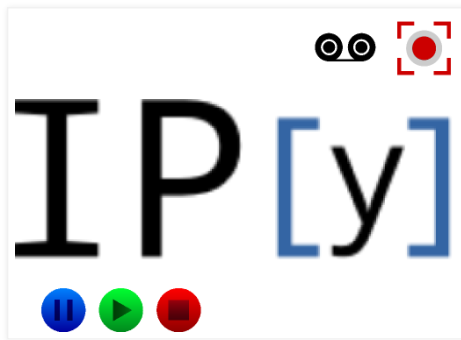


"Simple es mejor que complejo" (Tim Peters)

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

viernes, 8 de agosto de 2014

Registrar y ejecutar sesiones de IPython



Registrar una sesión de trabajo

Es posible registrar todas las entradas (y salidas) de una sesión en un archivo "LOG" añadiendo al ejecutable "ipython" el argumento "--logfile=archivolog", o bien, activando dicho registro durante una sesión interactiva con la función mágica %logstart.

Un archivo "LOG" puede ser ejecutado cuando se deseen repetir los mismos comandos de una sesión anterior.

Para activar el registro de entradas durante una sesión de trabajo:

: %logstart -o [nombearchivolog] [modo]

- La opción "-o" hará que se registren también las salidas.
- Si no se especifica nombre de archivo se utilizará el nombre predeterminado "ipython_log.py" y será almacenado en el directorio actual con el modo "rotate" activado por defecto.
- Si se especifica un nombre de archivo se almacenará en el directorio actual usando el modo "backup".

Modos de registro:

- **over:** Sobre escribir el archivo de log.
- **backup:** Renombrar (si existe) a log_name~ y comenzar un nuevo archivo log_name.
- **append:** Añadir la información al archivo existente.
- **rotate:** Crear "LOGs" numerados con la información de sesiones anteriores.

Ejemplo:

```
: %logstart sesion-01.ipy # Comenzará el registro en modo backup
```

Detener y reanudar el registro

Las funciones %logoff y %logon permiten detener temporalmente y reanudar el registro en un archivo que previamente se había iniciado con la función %logstart. Con la función %logstop terminaremos definitivamente el registro y con la función mágica %logstate podremos consultar en cualquier momento el estado del registro.

Para detener el registro temporalmente:

```
: %logoff
```

Para iniciar de nuevo el registro:

```
: %logon
```

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 valores...

[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 detener el registro definitivamente y cerrar el archivo LOG:

```
: %logstop
```

Para consultar el estado del registro:

```
: %logstate
```

Ejemplo:

```
In [1]: %logstart sesion-01.ipynb
```

Activating auto-logging. Current session state plus future input saved.

```
Filename      : sesion-1.py
```

```
Mode          : backup
```

```
Output logging : False
```

```
Raw input log  : False
```

```
Timestamping   : False
```

```
State          : active
```

```
In [2]: !ls *.ipynb
```

```
total 1
```

```
-rw-rw-r-- 1 usuario usuario 238 jul 22 23:21 sesion-1.ipynb
```

```
In [3]: %pwd
```

```
Out[3]: '/home/usuario/Local'
```

```
In [4]: %logstop
```

```
In [5]: %logstate
```

```
Logging has not been activated.
```

Contenido del archivo de registro "sesion-1.ipynb":

```
# IPython log file
```

```
get_ipython().magic('logstart sesion-1.ipynb')
```

```
get_ipython().system('ls *.ipynb')
```

```
get_ipython().magic('pwd')
```

```
get_ipython().magic('logstop')
```

Editar y Ejecutar un archivo de LOG

Para ejecutar un archivo de "LOG" lo primero que debemos hacer es editar su contenido para suprimir aquellas líneas que no sean necesarias y cambiar lo que consideremos oportuno.

Para editar un archivo y ejecutar su contenido inmediatamente después:

```
: %edit sesion-01.ipynb
```

En un equipo con GNU/Linux se editarán los archivos con el editor "vi". Si no conoce los comandos de este editor puede cambiarlo por otro que le resulte más cómodo: Ver [Editar y depurar scripts en IPython](#).

Para ejecutar un archivo directamente:

```
: %run sesion-01.ipynb
```

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

Publicado por Pherkad en [4:53](#)



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](#)