

★ Python 3 para impacientes ★

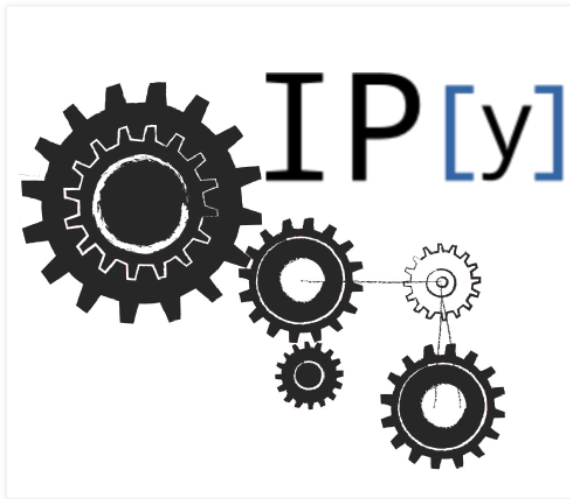


"Simple es mejor que complejo" (Tim Peters)

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

domingo, 10 de agosto de 2014

Editar y depurar scripts en IPython



Salvar y ejecutar código

La función **%save** almacena entradas en un archivo que puede ser ejecutado en cualquier momento con la función **%run**. Después de ejecutar el script se cargarán todos sus objetos (variables, listas, tuplas, funciones, etc.) en el espacio de nombre IPython.

Ejemplo:

```
In [1]: a = int(input("Unidades: "))
Unidades :5
```

```
In [2]: b = int(input("Precio: "))
Precio :10
```

```
In [3]: print("A pagar: ", a * b)
A pagar 50
```

Para salvar las entradas anteriores en un archivo indicaremos el nombre del archivo y las entradas desde-hasta a almacenar:

```
In [4]: %save script-01.py 1-3
The following commands were written to file `script-01.py`:
a = int(input("Unidades: "))
b = int(input("Precio: "))
print("A pagar", a*b)
```

Para ejecutar el código del script "script-01.py".

```
%run script-01.py
```

La función **%run** tiene parámetros especiales para medir el tiempo de ejecución de los scripts (-t), para depurar (-d) y para generar perfiles (-p).

Editar código

La función **%edit** permite editar scripts en nuestro editor favorito. Después de guardar IPython ejecutará inmediatamente el código editado.

```
%edit script-01.py
```

Por defecto, en GNU/Linux se utilizará el editor **"vi"** y en Windows **"notepad"** aunque en ambos

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

casos pueden cambiarse por otro que nos resulte más cómodo.

El editor **"vi"** es un editor bastante completo pero para su uso es necesario conocer una lista de comandos. Si estamos intentando editar en GNU/Linux y se ha iniciado **"vi"**: para salir sin guardar, teclear:

:q! (escribir dos puntos, **"q"** y el signo de cierre de admiración **"!"**)

Para cambiar el editor por defecto en GNU/Linux por **"nano"**, **"leafpad"**, **"gedit"** u otro que sea más manejable, declarar desde la consola la siguiente variable:

```
export EDITOR=nano
```

Después, si accedemos a IPython y utilizamos la función **%edit** editaremos de forma más cómoda los scripts. Si se desea hacer permanente este cambio se puede incluir la línea en el archivo **.bashrc** del usuario que se encuentra en **/home/usuario/.bashrc**

Para cambiar el editor por defecto en Windows declarar la variable **EDITOR** en la consola con el nombre del ejecutable de nuestro editor favorito:

```
set EDITOR=mieditor.exe
```

Para que sea permanente crear la variable de entorno **EDITOR** en Configuración del Sistema, en Propiedades del Equipo y añadir también la ruta del ejecutable a la variable de entorno **PATH**.

Depurar código

Después de que se produzca una excepción se puede llamar a la función **%debug** para utilizar el depurador Python (**pdb**) y examinar el origen del problema. Alternativamente, se puede usar la función mágica **%pdb** para depurar los scripts. Entre otras opciones permite mostrar el valor de variables, visualizar el código, ejecutar sentencias y rastrear hacia adelante o atrás el origen de un problema.

Puede ser una forma eficiente de depurar código, permitiendo eliminar en muchos casos la necesidad de mostrar el resultado de sentencias o recurrir a herramientas externas.

También, se puede examinar la ejecución de un programa desde el comienzo con:

```
%run -d programa.py
```

Medir el tiempo de ejecución

En el ejemplo siguiente se crea una función Python que es ejecutada con la función mágica **%timeit** para medir el tiempo de ejecución.

```
In: def imprime(numero):
    for elemento in range(numero):
        print(elemento)
```

```
In: %timeit imprime(100)
```

Relacionado:

- [Solucionando errores con el depurador](#)
- [El depurador PuDB](#)
- [Facilitando la depuración de programas con breakpoint\(\)](#)

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

Publicado por Pherkad en 2:55



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](#)