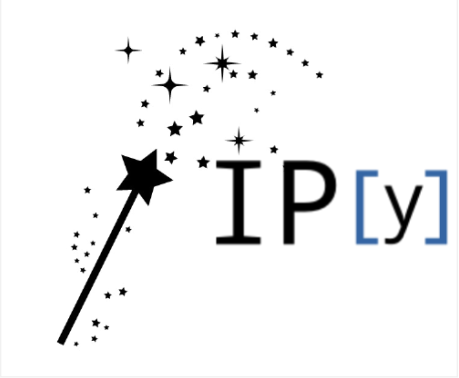
Python 3 para impacientes

"Simple es mejor que complejo" (Tim Peters)

jueves, 7 de agosto de 2014

Funciones mágicas de IPython



IPython tiene un conjunto de funciones predefinidas llamadas "mágicas" que representa una de las mejoras más importantes que aporta en relación al intérprete de Python. Hay dos tipos de funciones mágicas:

Funciones mágicas orientadas a líneas

Están precedidas por el carácter **"%"** y reciben como argumento el resto de la línea, donde se pasan sin necesidad de usar paréntesis o comillas. Ejemplo:

: %run script-01.py

El carácter **"%"** puede omitirse cuando se va a ejecutar una sola línea y cuando una característica de IPython denominada "automágica" está activada (por defecto lo está):

: run script-01.py

Este comportamiento se puede cambiar con la función mágica **%automagic**. La función %automagic activa o desactiva la característica automágica de una función. Si está desactivada obligatoriamente tendremos que usar el prefijo **"%"** cuando escribimos función mágicas.

Es recomendable el uso del carácter **"%"** cuando se utilizan funciones mágicas, incluso en líneas aisladas, porque si no lo usamos tendrán la prioridad más baja posible en la resolución de nombres que hace IPython. Si se utiliza el carácter **"%"** se podrán utilizar, sin problemas, variables con los mismos nombres que las funciones mágicas:

: cd ipython # %cd es una función que permite cambiar el directorio activo
/home/usuario/ipython

: cd=1 # Ahora "cd" es una variable

: cd .. # Y la función no funciona porque el sistema da prioridad a la variable
File "<ipython-input-3-9fedb3aff56c>", line 1
cd ..
^
SyntaxError: invalid syntax

: %cd .. # Si anteponemos el signo "%" funcionará como es debido
/home/usuario

: del cd # Si borramos la variable "cd" podremos usar la función sin "%"

: cd ipython
/home/usuario/ipython

Buscar

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 concebid...

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

Funciones mágicas orientadas a celdas

Este tipo de funciones están precedidas por "**%%**" y pueden recibir como argumento varias líneas.

Por ejemplo, la función mágica **%%writefile** permite almacenar en un archivo un conjunto de líneas. El archivo se guardará cuando avancemos con **[Enter]** después de dejar un línea en blanco. Si deseamos cancelar la entrada en cualquier momento pulsaremos **[Ctrl+c]**.

: %%writefile NombreArchivo
: linea 1
: linea 2
: linea 3
:
Writing NombreArchivo

Con la función mágica **%ls** podemos listar el directorio y comprobar que se ha creado el archivo. Para mostrar su contenido podemos ejecutar un comando del sistema:

: !cat NombreArchivo

Consultar la lista de funciones mágicas

Con la función **%lsmagic** se pueden listar todas las funciones mágicas disponibles. Con **%magic** se puede obtener información general sobre qué son las funciones mágicas.

Obtener información de ayuda

Para obtener información de ayuda sobre una función:

: %NombreFunciónMágica?

y para obtener información más detallada y si es posible mostrar código fuente:

: %NombreFunciónMágica??

Alias

La función mágica **%alias** permite definir otras funciones que pueden ejecutar comandos de forma abreviada, incluyendo parámetros.

: %alias mispy dir *.py

: mispy
script-01.py script-02.py script-03.py

In: usuario = "Manuel"
In: sistema = "PC00101"
In: %alias acceso echo "%s ha accedido a %s"

In: acceso Manuel PC00101
Manuel ha accedido a PC00101

In: acceso Manuel
ERROR: Alias <acceso> requires 2 arguments, 1 given.

In: acceso $usuario $sistema
Manuel ha accedido a PC00101

In: acceso {usuario} {sistema}
Manuel ha accedido a PC00101

Si se ejecuta %alias sin parámetros se mostrarán todos los alias existentes:

: %alias
Total number of aliases: 17
:
[('acceso', 'echo "%s ha accedido a %s"'),
 ('cat', 'cat'),
 ('clear', 'clear'),

La función mágica **%rehashx** actualiza la tabla de alias con todos los archivos ejecutables accesibles desde la variable de entorno **PATH**.

Crear funciones mágicas

A continuación, se muestra cómo crear una función mágica. Básicamente, se trata de editar el

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

código de la función mágica en nuestro editor favorito y, una vez creado, ejecutarlo dentro de una sesión interactiva de IPython. Después de su ejecución podremos usar la función mágica cada vez que sea necesario.

Dentro del perfil de un usuario cualquiera IPython hay una carpeta llamada "**startup**" donde podemos almacenar los scripts de funciones mágicas para que cuando iniciemos una sesión de trabajo con IPython se ejecuten automáticamente.

La función mágica que vamos a crear la guardaremos en un archivo llamado "**plataforma.py**" y hará uso del módulo **platform** de Python que muestra información de nuestro equipo (nombre, sistema operativo, versión, arquitectura, etc.). La función mágica se llamará "**%pf**" y tendrá dos usos posibles. Si ejecutamos "**%pf**" mostrará una información más reducida y si ejecutamos "**%pf all**" mostrará una información más completa.

- Crear la función %pf (plataforma.py):

```
from IPython.core.oinspect import getdoc
import platform

ip = get_ipython()

def platf(obj):
    print("Nombre :", platform.node())
    print("Sistema :", platform.system())
    print("Release :", platform.release())
    print("Version :", platform.version())
    if obj.upper() == "ALL":
        print("Equipo :", platform.machine())
        print("Procesador :", platform.processor())
        print("Arquitectura :", platform.architecture())
        print("Distribución :", platform.dist())
        print("Python Ver. :", platform.python_version())

ip.register_magic_function(platf, "line", "pf")
```

- Ejecutar el script que define la función mágica:

: %run plataforma.py

- Utilizar la función mágica "%pf":

Para mostrar información reducida

: %pf
Nombre : EQ001
Sistema : Windows
Release : 7
Version : 6.1.7601

Para mostrar información detallada

: %pf all
Nombre : EQ001
Sistema : Windows
Release : 7
Version : 6.1.7601
Equipo : x86
Procesador : x86 Family 6 Model 42 Stepping 7, GenuineIntel
Arquitectura : ('32bit', 'WindowsPE')
Distribución : ('', '', '')
Python Ver. : 3.4.1

Consultar GLOSARIO de Funciones Mágicas IPython

Ir al índice del tutorial de IPython

Publicado por Pherkad en 2:29

Etiquetas: IPython, Jupyter

Entrada más reciente Inicio Entrada antigua