Introducción a IPython

Jesús Espino García

Kaleidos
Python Madrid 2011

15 de Julio de 2011

¿Qué es IPython?

- Interprete de python con esteroides.
- Entorno pensado para la programación interactiva.
- Entorno pensado para la programación exploratoria.

¿Para qué sirve IPython?

- Depuración.
- Profiling.
- Micro prototipos.
- Pruebas de bibliotecas.
- ...

Tab completion

- Autocompletado con tabulador.
- Lista los atributos de un objeto.
- También autocompleta nombres de fichero.

Exploración de objetos

- IPython define macros para la exploración de objetos.
- %pdoc muestra el docstring de un objeto.
- %pdef muestra la definición de un objeto (método, clase).
- %psource muestra la implementación de un objeto (método, clase).
- %pfile muestra el código de todo el fichero donde se define un objeto (método, clase).
- %psearch busca objetos que corresponda con un patrón.
- obj? o obj?? muestra la información, o mas información sobre un objeto.
- %pdef? nos mostraría la información sobre pdef.

Trabajando con archivos

- IPython define una macro %run para ejecutar el contenido de un fichero.
- Cada %run lee y ejecuta todo el fichero (no es un import).
- El comando %run tiene 3 flags importantes.
  - -t mide el tiempo de ejecución.
  - -d ejecuta el código bajo pdb (en realidad ipdb).
  - -p ejecuta el código bajo un profiler.
- %pycat nos muestra el contenido de un fichero con resaltado de sintaxis

Usandolo como una shell

- IPython tiene autocompletado de nombres de ficheros y directorios.
- Cada %run lee y ejecuta todo el fichero (no es un import).
- Tambien tiene algunos comandos útiles típicos de una shell
  - %alias/ %unalias define un alias para un comando
  - %ls lista los ficheros
  - %cd me mueve de directorio
  - %pwd me devuelve el directorio actual
  - %env me devuelve el entorno actual
  - %pushd me mueve a un directorio y guarda el anterior en una pila
  - %popd saca un directorio de la pila y me mueve a el
  - %dirs me muestra la pila de directorios
  - %dhist histórico de directorios
  - !command ejecuta system(command)
  - !!command ejecuta el comando y me devuelve un objeto SList
  - %bg ejecuta una función en segundo plano
  - %bookmark añadir directorios a un lista de favoritos

Profiling

- %time ejecuta una función y dice el tiempo que ha tardado en ejecutar
- %prun ejecuta una función a través del profiler
- %timeit ejecuta una función N veces y dice el tiempo que ha tardado en ejecutar

Input/Output caches

Cache de Salida

- Todo resultado de ejecución es guardado en la cache de salida Out.
- El objeto Out es una lista donde el id es el numero de linea que se ejecuto.
- También existe un alias _N ara cada entrada en la cache de salida.
- También existen tres alias _, __ y ___ que se refieren a las tres ultimas lineas ejecutadas.

Cache de Entrada

- Todo ejecución de lineas se almacena en la cache de entrada In
- Es una lista de strings.
- Si queremos volver a ejecutar una serie de lineas podemos usar "exec In[23:27]+In[5:11]"
- También existe un alias _iN ara cada entrada en la cache de salida.
- También existen tres alias _i, _ii y _iii que se refieren a las tres ultimas lineas ejecutadas.

Macros

IPython nos permite definir macros que luego podamos ejecutar y guardar.

- %macro Define una nueva macro con un nombre y unas lineas de In.
- %edit Nos permite editar macros ya creadas (y mucho mas).
- %store Nos permite dar persistencia a una macro (y a otras variables).

Perfiles

- Existe una configuración de IPython para el usuario.
- IPython permite definir perfiles de configuración para tareas concretas.
- Por ejemplo podemos definir un perfil para matemáticas, o interacción con bbdd.

Referencias

- Proyecto ipython: http://ipython.org.
- Documentación de ipython: http://ipython.org/ipython-doc/stable/html/index.html.
- Introducción a IPython: ? (En el interprete de ipython).
- Repositorio Git: https://github.com/ipython
- Python-ES: python-es@python.org

Dudas

. . .