

Una Introducción rápida a IPython

Que es IPython?

IPython es un *shell* interactivo de Python. Empezado en 2001 por Ferando Perez, IPython se ha convertido en una de las herramientas mas importantes para computación científica en Python. Ofrece varias herramientas para escribir código y facilitar análisis de datos.

Que hace especial a IPython?

El IPython shell tiene varias ventajas sobre el shell estándar de Python. Estas son:

- Introspección mejorada.
- Pretty Printing
- Syntax highlighting
- Completación por tab
- Integración al systema operativo
- Una libreta interactiva por medio del browser (IPython Notebook) que provee un formato que es fácil de compartir y de utilizar en software de control de versiones
- Provee mas información sobre errores que el shell estándar de Python
- Capacidad para manejar texto vía Markdown o HTML
- Capacidad para multiples lenguajes (Ruby, Perl, Bash, R, entere otros)
- Capacidad para computación en paralelo.

Cómo usar IPython?

IPython shell

- Ejecuta `ipython`
- Funciona como el *Notebook* pero no puedes guardar tu trabajo y compartirlo. (Al menos que uses `%logging`, busca abajo!)

Qt shell

- Ejecuta `ipython qtconsole`
- Es un GUI console.
- Ejecuta `ipython qtconsole --pylab=inline` para generar graficas en la misma ventana.



IPython Notebook

- Ejecuta ipython notebook
- Una celda contiene código o texto. Para ejecutar código, presiona Ctrl+Enter o Ctrl+Return
- Una vez ejecutada una celda, las variables o funciones declaradas persisten para las siguientes celdas.
- Quieres hacer graficas con el IPython notebook? Ejecuta el comando ipython notebook --pylab=inline para hacerlo.
- Acuerdate de guardar!
- Necesitas mas ayuda? En una celda corre %quickref

In [21]: %quickref

Tab Completion

Busca todos las variables, objetos y funciones en el *namespace* actual

```
In [1]: #Declaro la variable i
i = 0
```

```
In [ ]: #Marca tab después de la i. Busca todos las variables, objetos y funciones en el n
#i<Tab>
i
```

```
In [ ]: #Marca tab después del punto. Te dira todos los metodos del objeto. Sirve para fun
#i.<Tab.
i.
```

```
In [22]: def hello_world(s):
        """Función simple"""
        print ('Hola Mundo y %s!') % s

hello_world('Ana')
```

Hola Mundo y Ana!

```
In [ ]: hello_world.
```

```
In [10]: #módulo de manejo de fechas/tiempo en Python
import datetime
```

```
In [ ]: datetime.
```

```
In [ ]: #Funciona también para el sistema operativo. Debe funcionar en Windows y Bash.
!cd
```



Introspección

Introspección es uno de las mejores características de Python. El language puede describir casi todo de si mismo. En IPython esta característica es mas avanzada.

- '?' te describirá el objeto, función, modulo o comando del sistema operativo
- '??' va mas allá: si hay código para la función de la regresa.
- Marca la orilla de la ventana para cerrarla

```
In [13]: #Con la variable i ya declarada, se escribe el '?' al lado. Te describira el objet
i?
```

```
In [15]: #Con la función hello_world ya declarada
hello_world?
```

```
In [16]: #Trae el código
hello_world??
```

```
In [25]: #Info para un módulo
datetime?
```

```
In [24]: #Funciona con el sistema operativo también
!cd?
```

Magic Commands

Son intrucciones especiales en IPython que son atajos para tareas comunes. Todos las instrucciones comienzan con el %.

```
In [26]: #Para mas información, ejecuta esta celda
%magic
```

Algunos comandos notables:

- %run: permite ejecutar un programa. Entre sus parámetros hay para ejecutar un programa con el debugger de Python (pdb). Luego de ejecutar, se tiene acceso a todo lo que estaba en el programa.
- %timeit: cuanto tiempo tomo ejecutar una función
- %debug: luego de una excepción, entrar %debug inmediatamente te lleva la excepción con mas información de lo que te daría el *shell* estándar. No puede correrse en el *Notebook*, solamente en el *IPython shell*.
- %prun: *profiling* para encontrar donde en el código se pasa el mayor tiempo.
- Introspección funciona con *magic commands*! Intenta %run?.

```
In [20]: %run?
```



Conclusión

IPython hace desarrollar, probar, guardar y compartir código mucho mas fácil. Une varias librerías (científicas como numpy y matplotlib; de desarrollo como pdb y CProfile) y hace su uso intuitivo. Todo el ecosistema de IPython hace mucho mas de lo descrito aquí (ej: computación en paralelo) pero espero que esta introducción rápida los motive a aprender mas sobre el.

Algunos Recursos

- [Página principal de IPython](#)
- [Para principiantes](#)
- [El fundador de IPython, Fernando Perez, hablando de la programación en las ciencias](#)

