Taller de Física Computacional

Nociones de Módulos

Cristián G. Sánchez y Carlos J. Ruestes

2020

Módulos

Módulos

De acuerdo al Glosario de Python:

> *Un* módulo *es un objeto que sirve como una unidad organizacional de código Python. Los módulos tienen un* espacio de nombres *que contienen objetos de Python arbitrarios. Los módulos se cargan en Python a través del proceso de* importación*.*

- En la práctica los módulos permiten construir bibliotecas de funciones, clases, variables con valores, etc..
- Los módulos se escriben en un archivo `nombre.py` que debe residir en una carpeta accesible a Python.
- Los módulos pueden organizarse en paquetes que pueden a su vez contener otros sub-módulos y/o sub-paquetes.

Módulos

Módulos

La Biblioteca estándar de python contiene decenas de módulos entre los que podemos nombrar por ejemplo:

- `math` Implementa funciones trascendentes y otras funciones matemáticas.
- `cmath` Implementa funciones matemáticas definidas para argumentos complejos.
- `os` Implementa interfases con el sistema operativo.
- Hay otros para el manejo de fechas, cadenas de caracteres, almacenamiento de objetos, compresión, criptografía, etc..

Módulos para programación científica

Los paquetes que más utilizaremos en el curso son:

NumPy

NumPy: Biblioteca de funciones y clases para cálculo numérico, en particular para arreglos multidimensionales. Muy recientemente, luego de 15 años de existencia, el equipo de desarrollo NumPy tiene un paper en nature.

matplotlib

matplotlib: Biblioteca de funciones y clases para visualización.

SciPy

SciPy: Una lista demasiado larga de funciones, clases, constantes y otros muy útiles en programación científica en general.

Módulos para programación científica

Otros proyectos importantes dentro del ecosistema de programación científica en Python incluyen pandas y SymPy. Estos paquetes implementan funcionalidades de estadística y análisis de datos y matemática simbólica respectivamente. Estos proyectos, entre otros, son financiados por la ONG NumFOCUS.

Sintaxis

Para importar un módulo o un módulo parte de un paquete con todos sus métodos pueden usarse las sintaxis:

```
import modulo
from paquete import módulo
```

o

```
import modulo as alias
from paquete import módulo as alias
```

En el caso de usar un alias este reemplaza el nombre del módulo para llamar a sus métodos.

Sintaxis

Para importar un método o conjunto de métodos de un módulo:

```
from modulo import método1, método2, método3
```

Lo mismo vale para importar módulos enteros de un paquete, etc..
Para importar todos los métodos de un módulo

```
from modulo import *
```

Si bien la primera forma se usa, la segunda no se recomienda porque "pisa" todo el espacio de nombres en el cual se hace la importación, es decir, se sobreescriben todos los nombres que haya ya definidos que sean iguales a los del módulo.

Sintaxis

Sintaxis

Los métodos, clases, etc. de un módulo se acceden usando la notación que vimos para las propiedades y métodos de un objeto:

```
modulo.funcion(parametro)
```

o

```
alias.función(parámetro)
```

Síntesis y recursos:

- Documentación de NumPy
- Funciones matemáticas en NumPy