UNIVERSIDAD NACIONAL
GUILLERMO BROWN

Introducción a la Programación

Algóritmos y Estructuras de Datos

-00184-

Dr. Diego Agustín Ambrossio

Anl. Sis. Angel Leonardo Bianco

Contenido:

- Módulos en Python
  - Comando import
  - Formas de importación
  - Módulos más utilizados

Módulos en Python

Nos permiten definir nuestras propias funciones, bloques de código, métodos, objetos, tipos de datos, etc.

El comando import nos permite utilizar funciones y objetos definidos fuera de las funciones predefinidas en Python.

¿Por qué estas funciones no están dentro de Python?
Son muchas y muy específicas a cada dominio. A raíz de esto, aparecen los siguientes problemas:

- Uso ineficiente de memoria.
- Conflictos de nombres.

Usar módulos nos permite trabajar de forma colaborativa, en equipos. De esta manera se podrán abordar proyectos de gran magnitud.

Cómo usar Import:

Hay dos fromas de usar el comando ***import***. Importar el modulo entero o un objeto del módulo.

Importar sólo un módulo:

import nombre_módulo

Luego podremos llamar a la cualquier función definida dentro del módulo nombre_módulo, de la siguiente forma: **nombre_módule.func**.

```
#Ejemplo nombramiento directo
import random
random.randrange(100)

import numpy
numpy.pi  #numpy.pi es la
aproximación del número pi.
```

También es posible cambiar el nombre de los módulos, por otro, generalmente más corto.

import nombre_módulo as nuevo_nombre

```
import math as m # el nommbre 'math' no es asociado al
modulo (el nombre de variable esta "libre")
                 # m es el objeto que contiene las
funciones del módulo math.

print(m.pi) # funciona...
math.pi # NameError
```

Cómo usar Import:

Importar objetos de un módulo:

La otra manera de importar funciones (objetos, métodos, etc.) definidas en un módulo es asignando un nuevo monbre para ellas:

from nombre_módulo import algo_1, algo_2, ... as nuevo_mombre_1, nuevo_nombre_2, ...

```
from math import pi as sliceofpie # sliceofpie es un número con el valor de math.pi

from math import pi # el comando "as" es opcional
pi == sliceofpie # Son Iguales
```

Si queremos importar un módulo de manera "completa", es decir, todas las funciones, objetos, etc, invocamos el siguiente comando:

from nombre_módulo import *

```
#import math
#math.pi

from random import *
randrange(100)

from math import *
pi


import random
for x in  dir(random): #Funciones provistas por
Python (built-in functions).
    print (x, end= " - ")

help(random)
```

Módulos más utilizados: MATH

Para la mayoría de los módulos "comunes" de Python, la documentación es extensiva y clara.
La librería math contiene funciones comunes (logaritmos, exponenciales, trigonométricas, ...) y algunas constantes (π, e, ...), además de algunas otras definiciones, por ejemplo inf (+∞) y nan que "No Representa a un Número" (Not A Number) y es del tipo número de punto flotante".

```
from math import inf,nan

print(inf+inf)
print(1/inf)
print(inf-inf)# "Indeterminado" aunque no genera error
print()
print(nan)
```

Módulos más utilizados: NUMPY

Su Documentación completa está disponible aquí: http://www.numpy.org.
Particularmente interesante para problemas matemáticos.
Nos ofrece un nuevo tipo de dato "arreglo" (array), también llamados "vectores", cuando todos los elementos son del mismo tipo.

```
import numpy as np

v=np.array([[1,1],[1,1]])# Crea un arreglo de 2 x 2 (es decir,
una matríz).
                         # v= [ 1 | 1
                         #      1 | 1 ]
print(v)

print([[1,1],[1,1]])

type(v)
```

Módulos más utilizados: NUMPY

Además tenemos provisto el tipo Matríz.

```
M=np.matrix([[0,1j],[1j,0]]) # Tipo : matrix

print(M.conjugate()) # Retorna wl conjugado de la matríz.

print(M*M)
```

Módulos más utilizados: SCIPY

Su Documentación completa está disponible aqui: https://www.scipy.org.
Contiene a los módulos numpy (analisis numerico), sympy (computación simbolica), y otras heramientas útiles que veremos luego.

Módulos más utilizados: RANDOM

Como lo indica su nombre, se utiliza para generar valores (u objetos) aleatorios.

```
import random

random.random() # Retorna un número de punto flotante
aleatorio entre 0 y 1 (igual que la calculadora)

random.randrange(0,4) # Retorna un entero aleatorio entre 0 y
3 (incluido)

Meses=['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','
Oct','Nov','Dec']

random.shuffle(Meses) #Mezcla los meses (la lista Meses ha
sido cambiada)random.sample(Meses,k=5) # Retorna una muestra
de 5 elementos de la población Meses
#Comparar con las opciones: random.shuffle(L) and
random.sample(L,len(L))
```

Módulos más utilizados: RANDOM

```
#Ejercicio: Paradoja del Cumpleaños. Dentro de un grupo de 10
personas, asumiendo que sus cumpleños siguen una distribución
normal sobre los 365 dias de año. Calcular una aproximación
de la probabilidad que dos personas cumplan años el mismo
día.
#Mismo ejercicio, pero para 30 personas.
```

Módulos más utilizados: RANDOM

```
# Ejercicio : Supongamos que hacemos una "tirada" de  10
dados.
    # Ganaremos $200 si la suma de los valores esta entre los
10 y 23.
    # Perderemos $1600 si la suma de los valores es 24 o 48.
    # Ganaremos $2500 si la suma de los valores es 42.
    # Perderemos $500 en cualquier otro caso.
  # Deberiamos jugar a este juego?
```

Importar módulos escritos por el usuario:

Podemos importar nuestras propios módulos, para ello, veremos como método de encapsilación: **Clases**.

METODO AXIOMATICO