**Notas de Python.**

**Índice**.

1. **Módulos**.
   1. **Más sobre módulos**.
      1. Ejecutando módulos como scripts.
      2. La ruta de búsqueda del módulo.
      3. “Compilado” archivos de Python.
   2. **Módulos standard**.
   3. **La función** dir()
   4. **Paquetes**.
      1. **Importando** \* From Package
      2. **Referencia Intra-package.**
      3. **Paquetes en múltiples directorios.**

**Módulos.**

Si sale del intérprete de Python y vuelve a ingresar, las definiciones que usted realizo (funciones y variables) se pierden. Por lo tanto, si usted quiere escribir un programa algo más largo, es mejor utilizar un editor de texto para preparar la entrada para el intérprete y ejecutarlo con ese archivo como entrada. Esto se conoce como crear un script. A medida que su programa crece, puede dividirlo en varios archivos para facilitar el mantenimiento. También puede usar una función práctica que haya escrito en varios programas sin copiar su definición en cada programa.

Para respaldar esto, Python tiene una forma de poner definiciones en un archivo y usarlas en un script o en una instancia interactiva del intérprete.

Un módulo es un archivo que contiene definiciones y sentencias de Python. El nombre del archivo es el nombre del módulo con el sufijo .py adjunto. Dentro de un módulo, el nombre del módulo (como una cadena) está disponible como el valor de la variable global \_\_name\_\_. Por ejemplo use su editor de texto favorito para crear un archivo llamado fibo.py en el directorio actual con los siguientes contenidos.

# Fibonacci numbers module

def fib(n): # write Fibonacci series up to n

a, b = 0, 1

while a < n:

print(a, end=' ')

a, b = b, a+b

print()

def fib2(n): # return Fibonacci series up to n

result = []

a, b = 0, 1

while a < n:

result.append(a)

a, b = b, a+b

return result

Ahora ingrese al intérprete de Python e importe esté modulo con el siguiente comando:

>>> import fibo

Esto no ingresa los nombres de las funciones definidas en fibo directamente en la tabla del símbolo actual; solo ingresa el nombre del módulo fibo allí. Usando el nombre del módulo puede acceder a las funciones.

>>> fibo.fib(1000)

0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987

>>> fibo.fib2(100)

[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

>>> fibo.\_\_name\_\_

'fibo'

Si tiene la intención de utilizar una función a menudo, puede asignarla a un nombre local:

>>> fib = fibo.fib

>>> fib(500)

0 1 1 2 3 5 8 13 21 34 55 89 144 233 377

***Fuente:***[*https://docs.python.org/3/tutorial/modules.html#modules*](https://docs.python.org/3/tutorial/modules.html#modules)

*Septiembre de 2018*