

Módulos

¿Qué es un Módulo?

Un módulo es un archivo que contiene un conjunto de códigos o un conjunto de funciones que pueden ser incluidos en una aplicación. Un módulo podría ser un archivo que contenga una sola variable, una función o una gran base de código.

Creación de un Módulo

Para crear un módulo escribimos nuestro código en un script de Python y lo guardamos como un archivo .py. Cree un archivo llamado mymodule.py dentro de la carpeta de su proyecto. Escribamos algún código en este archivo.

```
pythonCopy code
# archivo mymodule.py
def generar_nombre_completo(nombre, apellido):
    return nombre + ' ' + apellido
```

Cree un archivo main.py en el directorio de su proyecto e importe el archivo mymodule.py.

Importar un Módulo

Para importar el archivo usamos la palabra clave import y el nombre del archivo solamente.

```
pythonCopy code
# archivo main.py
import mymodule
print(mymodule.generar_nombre_completo('Facundo', 'Borrazás')) # Facundo Borrazás
```

Importar Funciones de un Módulo

Podemos tener muchas funciones en un archivo y podemos importar todas las funciones de manera diferente.

```
pythonCopy code
# archivo main.py
```

```

from mymodule import generar_nombre_completo, sumar_dos_numeros, persona, gravedad
print(generar_nombre_completo('Facundo', 'Borrazás'))
print(sumar_dos_numeros(1,9))
masa = 100;
peso = masa * gravedad
print(peso)
print(persona['firstname'])

```

Importar Funciones de un Módulo y Renombrar

Durante la importación podemos cambiar el nombre del módulo.

```

pythonCopy code
# archivo main.py
from mymodule import generar_nombre_completo as nombre_completo, sumar_dos_numeros as total, persona as p, gravedad as g
print(nombre_completo('Facundo', 'Borrazás'))
print(total(1, 9))
masa = 100;
peso = masa * g
print(peso)
print(p)
print(p['firstname'])

```

Importar Módulos Incorporados

Al igual que otros lenguajes de programación, también podemos importar módulos importando el archivo/función usando la palabra clave import.

Importemos el módulo común que usaremos la mayor parte del tiempo.

Algunos de los módulos incorporados comunes son: math, datetime, os, sys, random, statistics, collections, json, re.

Módulo OS

Usando el módulo os de Python es posible realizar automáticamente muchas tareas del sistema operativo. El módulo OS en Python proporciona funciones para crear, cambiar el directorio de trabajo actual y eliminar un directorio (carpeta), obtener sus contenidos, cambiar e identificar el directorio actual.

```
pythonCopy code
# importar el módulo
import os
# Crear un directorio
os.mkdir('nombre_directorio')
# Cambiar el directorio actual
os.chdir('ruta')
# Obtener el directorio de trabajo actual
os.getcwd()
# Eliminar directorio
os.rmdir()
```

Módulo Sys

El módulo sys proporciona funciones y variables utilizadas para manipular diferentes partes del entorno de ejecución de Python. La función sys.argv devuelve una lista de argumentos de línea de comandos pasados a un script de Python. El elemento en el índice 0 de esta lista es siempre el nombre del script, en el índice 1 está el argumento pasado desde la línea de comandos.

Ejemplo de un archivo script.py:

```
pythonCopy code
import sys
#print(sys.argv[0], argv[1],sys.argv[2]) # esta línea imprimiría: nombre_archivo argumento1 argumento2
print('¡Bienvenido {}. ¡Disfruta el desafío de {}!'.format(
    sys.argv[1], sys.argv[2]))
```

Ahora, para comprobar cómo funciona este script, escribí en la línea de comandos:

```
bashCopy code
python script.py Facundo 30DaysOfPython
```

El resultado:

```
bashCopy code
¡Bienvenido Facundo. ¡Disfruta el desafío de 30DayOfPython!
```

Algunos comandos útiles de sys:

```
pythonCopy code
# para salir de sys
sys.exit()
# Para saber el tamaño entero más grande que puede tomar
sys.maxsize
# Para conocer la ruta del entorno
sys.path
# Para conocer la versión de Python que estás usando
sys.version
```

Módulo Statistics

El módulo statistics proporciona funciones para estadísticas matemáticas de datos numéricos. Las funciones estadísticas populares que se definen en este módulo son: mean, median, mode, stdev, etc.

```
pythonCopy code
from statistics import * # importar todos los módulos de estadísticas
edades = [20, 20, 4, 24, 25, 22, 26, 20, 23, 22, 26]
print(mean(edades))      # ~22.9
print(median(edades))    # 23
print(mode(edades))      # 20
print(stdev(edades))     # ~2.3
```

Módulo Math

Módulo que contiene muchas operaciones y constantes matemáticas.

```
pythonCopy code
import math
print(math.pi)           # 3.141592653589793, constante pi
```

```
print(math.sqrt(2))      # 1.4142135623730951, raíz cuadrada
print(math.pow(2, 3))    # 8.0, función exponencial
print(math.floor(9.81))  # 9, redondeando hacia abajo
print(math.ceil(9.81))   # 10, redondeando hacia arriba
print(math.log10(100))   # 2, logaritmo con 10 como base
```

Ahora, hemos importado el módulo `math` que contiene muchas funciones que pueden ayudarnos a realizar cálculos matemáticos. Para verificar qué funciones tiene el módulo, podemos usar `help(math)` o `dir(math)`. Esto mostrará las funciones disponibles en el módulo. Si queremos importar solo una función específica del módulo, lo importamos de la siguiente manera:

```
pythonCopy code
from math import pi
print(pi)
```

También es posible importar múltiples funciones a la vez

```
pythonCopy code
from math import pi, sqrt, pow, floor, ceil, log10
print(pi)                # 3.141592653589793
print(sqrt(2))            # 1.4142135623730951
print(pow(2, 3))          # 8.0
print(floor(9.81))        # 9
print(ceil(9.81))         # 10
print(math.log10(100))    # 2
```

Pero si queremos importar todas las funciones en el módulo de matemáticas podemos usar `*`.

```
pythonCopy code
from math import *
print(pi)                # 3.141592653589793, constante pi
```

```
print(sqrt(2))          # 1.4142135623730951, raíz cuadrada
print(pow(2, 3))        # 8.0, exponencial
print(floor(9.81))      # 9, redondeando hacia abajo
print(ceil(9.81))       # 10, redondeando hacia arriba
print(math.log10(100))  # 2
```

Cuando importamos también podemos cambiar el nombre de la función.

```
pythonCopy code
from math import pi as PI
print(PI) # 3.141592653589793
```

Módulo de Cadena

Un módulo de cadena es un módulo útil para muchos propósitos. El ejemplo a continuación muestra algunos usos del módulo de cadena.

```
pythonCopy code
import string
print(string.ascii_letters) # abcdefghijklmnopqrstuvwxyzABCDEF
                              GHIJKLMNOPQRSTUVWXYZ
print(string.digits)        # 0123456789
print(string.punctuation)   # !"#$%&'()*+,-./:;<=>?@[\\]^_`{|}~
```

Módulo Aleatorio

A estas alturas ya estás familiarizado con la importación de módulos. Hagamos una importación más para familiarizarnos mucho con ella. Importemos el módulo aleatorio que nos da un número aleatorio entre 0 y 0.9999.... El módulo aleatorio tiene muchas funciones pero en esta sección solo usaremos random y randint.

```
pythonCopy code
from random import random, randint
print(random()) # no toma argumentos; devuelve un valor e
```

```
entre 0 y 0.9999  
print(randint(5, 20)) # devuelve un número entero aleatorio  
entre [5, 20] inclusive
```