

1. Vea el video en el que se explican los aspectos básicos del uso de módulos en Python, disponible [aquí](#).
2. Copie los códigos que escribió anteriormente y que definen su implementación de la función `mifactorial(n)`, y además el código que calcula su aproximación de  $\pi$  (almacenada ahora en la variable `mipi`) en un nuevo archivo llamado `mimodulo.py`. Este archivo puede usarse para definir un nuevo módulo. A continuación, en una sesión interactiva de Python, importe su función usando primero `import mimodulo` y llame a las funciones que están ahí definidas.
3. El factorial es una función comúnmente usada, y ya está implementada en diversos módulos populares de Python, por ejemplo, en el módulo `math`. Para verificar esto, importe el módulo `math` y verifique que la función `math.factorial` entrega los mismos valores ya calculados por usted. Lo mismo ocurre con el valor del número  $\pi$  (`math.pi`).
4. Aproveche que tiene cargado el módulo `math` e investigue qué funciones y variables están definidas en este módulo. Para esto, en una sesión interactiva de Python ejecute `dir(math)`, o bien el comando `help(math)` para revisar qué contiene. Alternativamente, o revise la [documentación en línea](#) disponible.
5. Una instalación típica de Python incluye los módulos de la “*librería estandar*”, con diversas herramientas para realizar una gran variedad de tareas en Python. Ver por ejemplo, [esta página](#) para breve introducción.
6. En el módulo `glob` de la librería estándar está implementada la función `glob.glob`, que crea una *lista* de los strings de los nombres de los archivos y/o carpetas disponibles en la carpeta del computador en la que se está ejecutando un programa. Para verificar esto, ejecute

```
import glob

lista_todos = glob.glob('*')
print('Lista de todos los archivos y carpetas en la carpeta actual:')
print(lista_todos)

lista_py = glob.glob('*.py')
print('Lista de todos los archivos .py en la carpeta actual:')
print(lista_py)
print('Numero total de archivos .py en la carpeta actual = ', len(lista_py))
```

7. En el módulo `os` puede encontrar herramientas para manipular archivos. Por ejemplo `os.rename(antiguo,nuevo)` renombra el archivo cuyo nombre original está dado por el string `antiguo` a un nuevo nombre correspondiente al string `nuevo`. Así, si en la carpeta donde se está ejecutando su programa existe un archivo llamado “01.py”, entonces el comando

```
os.rename('01.py', '01-old.py')
```

lo renombrará a “01-old.py”

8. Note que el módulo `os` también implementa una función similar a `glob.glob`, llamada `os.listdir`. Para comprobarlo, ejecute

```
os.listdir()
```

9. Usando lo anterior, escriba un programa en Python que renombre todos los archivos de extensión `.txt` en una carpeta, asignándoles un nuevo nombre con un número correlativo (es decir `'01.txt'`, `'02.txt'`, `'03.txt'`, etc.).