

1. Modifique el código que creó para calcular el factorial de un número entero (guía 12) para que ahora se defina una función `mifactorial`, de modo que el factorial de n se pueda luego llamar como `mifactorial(n)`.
2. La exponencial e^x de un número real x puede ser calculada usando la siguiente serie

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots \quad (1)$$

Reutilizando su código para calcular factoriales, escriba un programa que pregunte al usuario el valor de x y calcule e imprima el valor de e^x , usando la expresión (1). Para el cálculo considere 100 términos en la suma (1), es decir, que el programa calcule la suma hasta el término $x^{99}/99!$.

Nota 1: En el caso $n = 0$, se define el factorial de 0 igual al valor 1, es decir, $0! := 1$.

Nota 2: El “truncar” la serie (es decir, evaluarla hasta cierto número de términos) tiene como consecuencia que el valor calculado es sólo una *aproximación* del valor exacto (e^x). Esta aproximación es mejor si se incluyen más términos.

3. Utilice el programa que acaba de escribir para calcular (una aproximación de) el valor de e (el número de Euler). Compare su resultado con el valor listado en este [artículo de wikipedia](#).
4. Escriba un programa que evalúe (una aproximación de) el número π . Para esto, use la siguiente expresión en serie (desarrollada por el gran [Leonhard Euler](#)),

$$\pi = \sum_{n=0}^{\infty} \frac{2^{n+1}(n!)^2}{(2n+1)!} = \left[2^1 \frac{(0!)^2}{1!} + 2^2 \frac{(1!)^2}{3!} + 2^3 \frac{(2!)^2}{5!} + 2^4 \frac{(3!)^2}{7!} + \dots \right]. \quad (2)$$

Lo anterior es un ejemplo de un método con el que se puede calcular el valor de π , con precisión cada vez mayor al agregar más y más términos. Dado que π es un número irracional, sólo se conoce su valor (calculado con métodos similares) hasta un cierto número de decimales. El record actual lo tiene Shigeru Kondo, quien logró calcular π con 10000000000000 decimales!. Compare el valor que usted obtenido con el listado en en este [artículo de wikipedia](#).

5. Copie los códigos que escribió anteriormente y que definen su implementación de la función `mifactorial(n)`, y además el código que calcula de su aproximación de π (almacenada ahora en la variable `mipi`) en un nuevo archivo llamado `misfunciones.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.
6. 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 π (`math.pi`).
7. Aproveche que tiene cargado el módulo `math` e investigue qué funciones y variables están definidas en este módulo. Para esto, ejecute `dir(math)`, o bien el comando `help(math)` para revisar qué contiene. Alternativamente, o revise la [documentación en línea](#) disponible.

8. Una instalación típica de Python incluye los módulos de la “*librería estandar*”, con diversas herramienta para realizar una gran variedad de tareas en Python. Ver por ejemplo, [este tutorial](#) para breve introducción.
9. 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))
```

10. 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”

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

```
os.listdir()
```

12. 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.).