mouredev pro

ENUNCIADOS EJERCICIOS

CURSO DE PYTHON INTERMEDIO

Dates

1. Crea una variable con la fecha y hora actual.
2. Imprime solo el año, mes y día de la fecha actual.
3. Crea una fecha específica: 25 de diciembre de 2025 y muéstrala.
4. Muestra solo la hora, los minutos y los segundos de un objeto time.
5. Calcula cuántos días faltan para el 1 de enero del año siguiente.
6. Crea una función que reciba una fecha y devuelva su timestamp.
7. Suma 30 días a la fecha actual usando timedelta.
8. Crea una fecha y añade 1 mes (consejo: hazlo sumando 30 días como simplificación).
9. Compara dos fechas y muestra cuál es anterior.
10. Crea una lista con varias fechas y ordénalas cronológicamente.

List comprehension

1. Genera una lista utilizando comprensión con los números del 0 al 10.

2. Crea una lista utilizando comprensión con los cuadrados de los números del 1 al 10.

3. Genera una lista utilizando comprensión con los números pares del 0 al 20.

4. Convierte una lista de temperaturas en Celsius a Fahrenheit utilizando comprensión.

5. Crea una lista utilizando comprensión con los caracteres de una cadena.

6. Filtra una lista de palabras y deja solo las que tienen más de 4 letras utilizando comprensión.

7. Aumenta en 5 cada número de una lista con comprensión usando una función externa.

8. Crea una lista de booleanos que indique si cada número es mayor que 10 utilizando comprensión.

9. Multiplica solo los números impares por 3 en una lista utilizando comprensión.

10. Usa comprensión de listas anidada para generar una matriz 3x3 con números del 1 al 9.

Lambdas

1. Crea una lambda que sume dos números.
2. Crea una lambda que calcule el cuadrado de un número.
3. Crea una lambda que devuelva el mayor de dos números.
4. Crea una lambda que sume 10 a un número dado.
5. Crea una lambda que devuelva el último carácter de una cadena.
6. Crea una lambda que indique si una palabra tiene más de 6 letras.
7. Crea una lambda que convierta una cadena a minúsculas.
8. Crea una lambda que devuelva True si un número es positivo.
9. Crea una lambda que devuelva "Cadena vacía" si el string está vacío.
10. Crea una lambda que calcule el precio final con un impuesto añadido del 21%.

Funciones de orden superior

1. Crea una función que reciba una función y un número, y devuelva el resultado de aplicar la función al número.

2. Crea una función que reciba dos números y una función, y devuelva el resultado de sumar los dos números y aplicar la función.

3. Crea una función que devuelva otra función que sume un número fijo.

4. Usa map() con lambda para multiplicar cada número de una lista por 10.

5. Usa filter() con lambda para quedarte solo con los números pares.

6. Usa reduce() con lambda para obtener la suma total de una lista.

7. Escribe una función que devuelva una función que reciba un nombre y devuelva "Hola, ".

8. Crea una función que reciba una lista y una función, y cuente cuántos elementos cumplen con la función.

9. Crea una función que reciba dos funciones y un número, y las aplique en orden.

10. Crea una función que reciba una lista y una función, y aplique esa función a cada elemento usando un bucle (sin map).

Tipos de error

1. Genera un SyntaxError al imprimir una cadena sin paréntesis.

2. Genera un NameError intentando usar una variable no definida.

3. Genera un IndexError accediendo a un índice inexistente de una lista.

4. Genera un ModuleNotFoundError al importar un módulo inexistente.

5. Genera un AttributeError accediendo a un atributo que no existe.

6. Genera un KeyError al acceder a una clave inexistente de un diccionario.

7. Genera un TypeError usando tipos incorrectos (índice string en lista).

8. Genera un ImportError al importar una función que no existe desde un módulo.

9. Genera un ValueError intentando convertir un string no numérico a entero.

10. Intenta detectar si un error ocurre usando try-except con un KeyError.

Manejo de ficheros

1. Crea un archivo de texto y escribe en él la frase "Hola desde Python".

2. Abre un archivo y lee todo su contenido.

3. Añade una nueva línea al final del archivo con el texto "Línea añadida".

4. Lee solo los primeros 10 caracteres del archivo.

5. Usa seek para volver al inicio del archivo y leer desde ahí.

6. Lee e imprime el contenido línea por línea usando readline.

7. Lee todas las líneas del archivo en una lista y recórrelas con un bucle.

8. Crea un archivo nuevo que sobrescriba si ya existe, y escribe varias líneas.

9. Usa una función para abrir un archivo, escribir texto y cerrarlo automáticamente con with.

10. Lee un archivo línea por línea y muestra solo las que contienen la palabra "Python".

Expresiones regulares

1. Busca si una cadena empieza por "Hola".
2. Busca la palabra "Python" en una cadena aunque esté en minúsculas.
3. Encuentra todas las apariciones de la palabra "curso" en una cadena.
4. Reemplaza todas las apariciones de "lección" por "LECCIÓN".
5. Divide un texto en partes separadas por comas.
6. Busca la primera palabra que comience con "A" o "a".
7. Encuentra todas las palabras en una cadena que terminen en "ción".
8. Verifica si una cadena contiene solo números.
9. Reemplaza todos los números de una cadena por el texto "[número]".
10. Encuentra todas las palabras de 4 letras exactas en una cadena.

Manejo de paquetes

1. Importa el módulo math y muestra el valor de pi.

2. Crea un array de números usando numpy y multiplícalo por 3.

3. Muestra la versión instalada de numpy.

4. Realiza una petición HTTP con requests a una API pública y muestra el código de estado.

5. Importa una función llamada sum_two_values desde un paquete personalizado mypackage.arithmetics y utilízala.

6. Usa pandas para crear un DataFrame con nombres en español.

7. Ejecuta el comando para instalar el paquete requests desde la terminal.

8. Usa requests para obtener datos de una API y extrae solo los nombres de los primeros Pokémon.

9. Muestra todos los paquetes instalados con pip desde la terminal.

10. Escribe una línea de código que muestre la ayuda sobre el paquete numpy desde Python.


{m}pro