

Ejercicio 33.1

Escribí un programa que le pida al usuario su nombre. Cuando responda, escribí su nombre en un archivo llamado `guest.txt`.

Ejercicio 33.2

Escribí un bucle `while` que le pida a los usuarios su nombre. Recolectá todos los nombres que se ingresen y después escribilos en un archivo llamado `guest_book.txt`. Asegurate de que cada entrada aparezca en una nueva línea del archivo.

Ejercicio 33.3

Un problema común al pedir una entrada numérica es que la gente ingrese texto en lugar de números. Cuando intentás convertir la entrada a un `int`, vas a obtener un `ValueError`.

Escribí un programa que pida dos números. Sumalos e imprimí el resultado.

Capturá el `ValueError` si alguna de las entradas no es un número y mostrá un mensaje de error amistoso.

Probá tu programa ingresando dos números y luego ingresando texto en lugar de un número.

Ejercicio 33.4

Encapsulá tu código del ejercicio 33.3 en un bucle `while` para que el usuario pueda seguir ingresando números, incluso si comete un error e ingresa texto en lugar de un número.

Ejercicio 33.5

Creá dos archivos, `gatos.txt` y `perros.txt`. Guardá al menos tres nombres de gatos en el primer archivo y tres nombres de perros en el segundo.

Escribí un programa que intente leer estos archivos e imprimir el contenido en pantalla.

Encapsulá tu código en un bloque `try-except` para capturar el error `FileNotFound`, y mostrá un mensaje amistoso si falta alguno de los archivos.

Mové uno de los archivos a otra ubicación en tu sistema y asegurate de que el código del bloque `except` se ejecute correctamente.

Ejercicio 33.6

Modificá tu bloque except del ejercicio 33.5 para que falle en silencio si alguno de los archivos falta.

Ejercicio 33.7

Visitá Project Gutenberg y elegí algunos textos que te gustaría analizar.

Descargá los archivos de texto de esas obras o copió el texto sin formato desde el navegador a un archivo de texto en tu computadora.

Podés usar el método `count()` para saber cuántas veces aparece una palabra o frase en un string.

Por ejemplo, el siguiente código cuenta cuántas veces aparece 'row' en un string:

```
>>> linea = "Row, row, row your boat"
>>> linea.count('row')
2
>>> linea.lower().count('row')
3
```

Fijate que al convertir el string a minúsculas con `lower()` podés captar todas las apariciones de la palabra que estás buscando, sin importar cómo esté escrita.

Escribí un programa que lea los archivos que encontraste en Project Gutenberg y determine cuántas veces aparece la palabra 'the' en cada texto.

Esto va a ser una aproximación, porque también va a contar palabras como 'then' o 'there'.

Probá contar 'the ' (con un espacio) y fijate cuánto más bajo es el conteo.

Ejercicio 33.8

Escribí un programa que le pida a la persona su número favorito. Usá `json.dumps()` para guardar ese número en un archivo.

Después escribí un programa separado que lea ese valor y muestre el mensaje:

“¡Sé cuál es tu número favorito! Es ____.”

Ejercicio 33.9

Combiná los dos programas que escribiste en el ejercicio anterior (33.8) en un solo archivo.

Si el número ya está guardado, mostralo a la persona como su número favorito.

Si no, pedile su número favorito y guardalo en un archivo. Ejecutá el programa dos veces para verificar que funcione correctamente.

Ejercicio 33.10

El ejemplo del programa que recuerda el nombre del usuario solo guarda una pieza de información: el nombre de la persona.

Expandí este ejemplo pidiendo dos datos más sobre la persona (por ejemplo, edad y ciudad), y guardá toda la información que recolectes en un diccionario.

Escribí ese diccionario en un archivo usando `json.dumps()`, y luego leelo usando `json.loads()`.

Mostrá un resumen que diga exactamente qué recuerda el programa sobre la persona.

Ejercicio 33.11

La versión final del programa del ejercicio 33.10 asume que la persona ya ingresó su nombre o que es la primera vez que se ejecuta el programa.

Deberíamos modificarlo en caso de que la persona actual no sea la misma que usó el programa la vez anterior.

Antes de mostrar el mensaje de bienvenida en `saludar_usuario()`, preguntá si ese nombre es correcto.

Si no lo es, llamá a `obtener_nuevo_usuario()` para pedir el nombre correcto.

Ejercicio 33.12 (Easter Egg)

La paradoja del cumpleaños dice que la probabilidad de que dos personas en una sala tengan el mismo cumpleaños es mayor al 50 %, siempre que n , la cantidad de personas en la sala, sea mayor a 23.

Esta propiedad no es realmente una paradoja, pero a mucha gente le resulta sorprendente.

Diseñá un programa en Python que pueda probar esta paradoja mediante una serie de experimentos con cumpleaños generados aleatoriamente, y que la pruebe para los valores:

$n = 5, 10, 15, 20, \dots, 100$.

Ejercicio 34.1

Escribí una función que acepte dos parámetros: el nombre de una ciudad y el de un país. La función debería devolver una única cadena con el formato:

Ciudad, País, como por ejemplo: Santiago, Chile.

Guardá la función en un módulo llamado `city_functions.py` y guardá este archivo en una nueva carpeta para que `pytest` no intente ejecutar las pruebas que ya escribiste.

Creá un archivo llamado `test_cities.py` que pruebe la función que acabás de escribir.

Escribí una función llamada `test_city_country()` para verificar que llamar a tu función con valores como `'santiago'` y `'chile'` devuelve la cadena correcta. Ejecutá la prueba y asegurate de que `test_city_country()` pase.

Ejercicio 34.2

Modificá tu función para que ahora requiera un tercer parámetro: `population` (población). Ahora debería devolver una única cadena con el formato:

Ciudad, País – población xxx, como por ejemplo: Santiago, Chile – población 5000000.

Ejecutá la prueba otra vez y asegurate de que `test_city_country()` falle esta vez.

Modificá la función para que el parámetro de población sea opcional. Ejecutá la prueba y asegurate de que `test_city_country()` vuelva a pasar.

Escribí una segunda prueba llamada `test_city_country_population()` para verificar que podés llamar a tu función con los valores `'santiago'`, `'chile'` y `population=5000000`.

Ejecutá las pruebas una vez más y asegurate de que esta nueva prueba pase.