

Ejercicio 9.1

Pensá en al menos cinco lugares del mundo que te gustaría visitar.

- Guardá las ubicaciones en una lista. Asegurate de que la lista no esté en orden alfabético.
- Imprimí tu lista en su orden original. No te preocupes por formatearla de manera ordenada; simplemente imprimila como una lista en bruto en Python.
- Usa `sorted()` para imprimir tu lista en orden alfabético sin modificar la lista original.
- Mostrá que tu lista sigue en su orden original imprimiéndola nuevamente.
- Usa `sorted()` para imprimir tu lista en orden alfabético inverso sin cambiar el orden de la lista original.
- Mostrá que tu lista sigue en su orden original imprimiéndola otra vez.
- Usá `reverse()` para cambiar el orden de tu lista. Imprimí la lista para mostrar que su orden ha cambiado.
- Usá `reverse()` nuevamente para devolver la lista a su orden original. Imprimí la lista para mostrar que ha vuelto a su orden inicial.
- Usá `sort()` para ordenar tu lista en orden alfabético. Imprimí la lista para mostrar que su orden ha cambiado.
- Usá `sort()` para ordenar tu lista en orden alfabético inverso. Imprimí la lista para mostrar que su orden ha cambiado.

Ejercicio 9.2

Trabajando con uno de los programas de los ejercicios 8.1 a 8.4:

- Usa `len()` para imprimir un mensaje que indique cuántas personas estás invitando a cenar.

Ejercicio 10

Pensá en al menos tres tipos de pizza que te encanten. Guardá estos nombres en una lista y usá un bucle `for` para imprimir el nombre de cada pizza.

- Modificá el bucle `for` para imprimir una oración que use el nombre de la pizza en lugar de solo imprimir su nombre. Para cada pizza, deberías tener una línea de salida con una afirmación simple, como: Me gusta la pizza de pepperoni.
- Agregá una línea al final de tu programa, fuera del bucle `for`, que diga cuánto te gusta la pizza. La salida debería tener tres o más líneas sobre los tipos de pizza que te gustan y, al final, una frase extra, como: ¡Me encanta la pizza!

Ejercicio 11

Hacé una lista con los números del uno al un millón y después usá `min()` y `max()` para asegurarte de que tu lista realmente empieza en uno y termina en un millón.

También, usá la función `sum()` para ver qué tan rápido Python puede sumar un millón de números.

Ejercicio 12

Empezá con tu programa del Ejercicio 10.

Hacé una copia de la lista de pizzas y llamala pizzas_amigo. Después, hacé lo siguiente:

- Agregá una nueva pizza a la lista original.
- Agregá una pizza diferente a la lista pizzas_amigo.
- Demostrá que tenés dos listas separadas. Imprimí el mensaje “Mis pizzas favoritas son:” y usá un bucle for para imprimir la primera lista.
- Imprimí el mensaje “Las pizzas favoritas de mi amigo/a son:” y usá un bucle for para imprimir la segunda lista.
- Asegurate de que cada pizza nueva esté guardada en la lista correspondiente.

Ejercicio 13.1

Imaginá que un alien acaba de ser derribado en un juego. Creá una variable llamada color_alien y asigne un valor: 'verde', 'amarillo' o 'rojo'.

Escribí una estructura if para comprobar si el color del alien es 'verde'. Si lo es, mostrará un mensaje diciendo que el jugador acaba de ganar 5 puntos.

Hacé una versión de este programa que pase la condición del if y otra que falle (la versión que falla no mostrará ningún mensaje).

Ejercicio 13.2

Elegí un color para el alien, como hiciste en el ejercicio 13.1, y escribí una estructura if-else.

Si el color del alien es 'verde', mostrará un mensaje diciendo que el jugador ganó 5 puntos por derribar al alien.

Si el color del alien no es 'verde', mostrará un mensaje diciendo que el jugador ganó 10 puntos.

Hacé una versión del programa que haga que se ejecute el bloque if y otra donde se ejecute el bloque else.

Ejercicio 13.3

Convertí tu estructura if-else del ejercicio 13.2 en una estructura if-elif-else.

Si el alien es 'verde', mostrará un mensaje diciendo que el jugador ganó 5 puntos.

Si el alien es 'amarillo', mostrará un mensaje diciendo que el jugador ganó 10 puntos.

Si el alien es 'rojo', mostrará un mensaje diciendo que el jugador ganó 15 puntos.

Si el alien tiene otro color, mostrará un mensaje diciendo que el jugador ganó 0 puntos.

Escribí tres versiones del programa, asegurándote de que cada mensaje se imprima para el color de alien correspondiente.

Ejercicio 13.4

Escribí una estructura if-elif-else que determine la etapa de la vida de una persona. Definí un valor para la variable edad, y luego:

Si la persona tiene menos de 2 años, imprimí un mensaje diciendo que es un bebé.

Si la persona tiene al menos 2 años pero menos de 4, imprimí un mensaje diciendo que es un nene/a chiquito/a.

Si la persona tiene al menos 4 años pero menos de 13, imprimí un mensaje diciendo que es un nene/a.

Si la persona tiene al menos 13 años pero menos de 20, imprimí un mensaje diciendo que es un adolescente.

Si la persona tiene al menos 20 años pero menos de 65, imprimí un mensaje diciendo que es un adulto.

Si la persona tiene 65 años o más, imprimí un mensaje diciendo que es un adulto mayor.

Ejercicio 13.5

Hacé una lista con tus frutas favoritas y luego escribí una serie de declaraciones if independientes para verificar si ciertas frutas están en tu lista.

Creá una lista con tres frutas favoritas y llamala `frutas_favoritas`.

Escribí cuatro declaraciones if. Cada una debe comprobar si un tipo de fruta está en tu lista. Si la fruta está en la lista, el bloque if tiene que imprimir un mensaje como: "¡Te re copan las frutillas!".

Ejercicio 14.1

Hacé una lista con cinco o más nombres de usuario, incluyendo el nombre 'admin'. Imaginá que estás escribiendo un código que muestra un saludo a cada usuario después de que inicien sesión en un sitio web.

Recorré la lista y mostrales un saludo a cada usuario.

Si el nombre de usuario es 'admin', imprimí un saludo especial, como:

"Hola admin, ¿querés ver un informe de estado?"

Si el usuario no es 'admin', imprimí un saludo genérico, como:

"Hola Jaden, gracias por volver a iniciar sesión."

Ejercicio 14.2

Agregale un chequeo al código del ejercicio anterior para asegurarte de que la lista de usuarios no esté vacía.

Si la lista está vacía, imprimí el mensaje:

"¡Necesitamos encontrar algunos usuarios!"

Ejercicio 14.3

Hacé lo siguiente para crear un programa que simule cómo los sitios web verifican que cada usuario tenga un nombre único.

Creá una lista con cinco o más nombres de usuario y llamala `usuarios_actuales`.

Creá otra lista con cinco nombres de usuario y llamala `usuarios_nuevos`. Asegurate de que uno o dos nombres de `usuarios_nuevos` ya existan en `usuarios_actuales`.

Recorré la lista `usuarios_nuevos` y fijate si cada nombre de usuario ya fue usado.

Si ya existe, imprimí un mensaje diciendo que la persona tiene que elegir otro nombre.

Si no fue usado, imprimí un mensaje diciendo que el nombre de usuario está disponible.

Asegurate de que la comparación no distinga entre mayúsculas y minúsculas. Por ejemplo, si 'Juan' ya fue usado, 'JUAN' no debería ser aceptado. Para esto, podés hacer una copia de `usuarios_actuales` con todas las versiones en minúsculas.

Ejercicio 14.4

Los números ordinales indican la posición en una lista, como 1ero o 2do.

Guardá los números del 1 al 9 en una lista.

Recorré la lista con un bucle.

Usá una estructura `if-elif-else` dentro del bucle para imprimir el sufijo ordinal correcto para cada número. O sea, si el número es 1, debería imprimirse "1ero".

El resultado debería verse con cada número ordinal en una línea distinta.