

Ejercicio 29.1

Un puesto de helados es un tipo específico de restaurante. Escribí una clase llamada `PuestoDeHelados` que herede de la clase `Restaurante` que escribiste en el Ejercicio 27.1 o el Ejercicio 28.1. Cualquiera de las dos versiones de la clase sirve, elegí la que más te guste.

Agregá un atributo llamado `sabores`, que almacene una lista con los sabores de helado. Escribí un método que muestre estos sabores. Creá una instancia de `PuestoDeHelados` y llamá a este método.

Ejercicio 29.2

Un administrador es un tipo especial de usuario. Escribí una clase llamada `Administrador` que herede de la clase `Usuario` que escribiste en el ejercicio 27.3 o el ejercicio 28.2.

Agregá un atributo llamado `privilegios`, que almacene una lista de strings como "puede agregar publicaciones", "puede eliminar publicaciones", "puede bloquear usuarios", etc.

Escribí un método llamado `mostrar_privilegios()`, que liste los privilegios del administrador. Creá una instancia de `Administrador` y llamá a tu método.

Ejercicio 29.3

Escribí una clase separada llamada `Privilegios`. La clase debería tener un único atributo, `privilegios`, que almacene una lista de strings como se describió en el Ejercicio 29.2.

Mové el método `mostrar_privilegios()` a esta clase. Creá una instancia de `Privilegios` como un atributo dentro de la clase `Administrador`.

Creá una nueva instancia de `Administrador` y usá el método para mostrar sus privilegios.

Ejercicio 29.4

Usando:

```
class AutoElectrico(Auto):  
    """Representa aspectos de un auto, específicos de los  
    vehículos eléctricos."""  
    def __init__(self, marca, modelo, año):  
        """        Inicializa los atributos de la clase padre.  
        Luego inicializa los atributos específicos de un auto  
        eléctrico.  
        """  
        super().__init__(marca, modelo, año)  
        self.bateria = Bateria()
```

```

mi_leaf = AutoElectrico('nissan', 'leaf', 2024)
print(mi_leaf.obtener_nombre_descriptivo())
mi_leaf.bateria.describir_bateria()
mi_leaf.bateria.obtener_autonomia()

class Bateria:

    """Definimos una nueva clase llamada Bateria que no hereda de
    ninguna otra clase."""

    def __init__(self, TamañoBateria=40):

        """Inicializa la batería con un tamaño predeterminado de
        40 si no se proporciona otro valor."""

        self.TamañoBateria = TamañoBateria

```

Agregá un método a la clase Bateria llamado `actualizar_bateria()`. Este método debería verificar el tamaño de la batería y establecer la capacidad en 65 si aún no lo está.

Creá un auto eléctrico con una batería de tamaño predeterminado, llamá al método `obtener_autonomia()` una vez y luego llamalo de nuevo después de actualizar la batería. Deberías notar un aumento en la autonomía del auto.

Ejercicio 30.1

Usando tu última versión de la clase `Restaurant`, guardala en un módulo separado. Creá un archivo aparte que importe `Restaurant`. Luego, creá una instancia de `Restaurant` y llamá a uno de sus métodos para comprobar que la importación funciona correctamente.

Ejercicio 30.2

Partí de tu trabajo en el ejercicio 29.3. Guardá las clases `User`, `Privileges` y `Admin` en un módulo. Creá un archivo aparte, generá una instancia de `Admin` y llamá al método `show_privileges()` para verificar que todo esté funcionando bien.

Ejercicio 30.3

Guardá la clase `User` en un módulo y las clases `Privileges` y `Admin` en otro módulo separado. Luego, en un tercer archivo, creá una instancia de `Admin` y llamá a `show_privileges()` para comprobar que todo sigue funcionando correctamente.

Ejercicio 31.1

Dado: Creá una clase Die con un atributo llamado sides, que tenga un valor por defecto de 6. Escribí un método llamado roll_die() que imprima un número aleatorio entre 1 y la cantidad de lados que tenga el dado. Creá un dado de 6 caras y tiralo 10 veces.

Después, creá un dado de 10 caras y uno de 20 caras. Tiralos 10 veces cada uno.

Ejercicio 31.2

Lotería: Hacé una lista o tupla con 10 números y 5 letras. Seleccioná al azar 4 números o letras de la lista e imprimí un mensaje diciendo que cualquier boleto que tenga esos 4 números o letras gana un premio.

Ejercicio 31.1

Análisis de Lotería: Usá un bucle para ver qué tan difícil puede ser ganar en la lotería que acabás de modelar. Creá una lista o tupla llamada my_ticket. Escribí un bucle que siga sacando números al azar hasta que salga tu boleto. Imprimí un mensaje diciendo cuántas veces tuvo que ejecutarse el bucle hasta que ganaste.

Ejercicio 31.4

Módulo de Python de la Semana: Un excelente recurso para explorar la biblioteca estándar de Python es un sitio llamado Python Module of the Week. Andá a <https://pymotw.com> y mirá el índice. Buscá un módulo que te parezca interesante y leé sobre él, quizás empezando con el módulo random.

Ejercicio 32.1

Guardá en un archivo los primeros un millón de dígitos de pi. <https://www.piday.org/million/>. También tenés un archivo en el classroom.

Preguntale al usuario su fecha de nacimiento (puede ser un string de dígitos todos seguidos, como por ejemplo: 121209) y que le diga al usuario si esa secuencia aparece entre los primeros un millón de dígitos de pi.

Sería algo como:

Ingresa tu cumpleaños, en la forma ddmmaa: 120372

¡Tu cumpleaños aparece en los primeros un millón de dígitos de pi!

Ejercicio 32.2

Abrí un archivo en blanco en tu editor de texto y escribí unas líneas resumiendo lo que aprendiste hasta ahora sobre Python. Empezá cada línea con la frase En Python podés.... Guardá

el archivo como `learning_python.txt` en el mismo directorio donde tenés los ejercicios de esta clase. Luego, escribí un programa que lea el archivo y muestre su contenido dos veces:

Leyendo todo el archivo de una sola vez y mostrándolo en pantalla.

Almacenando las líneas en una lista y luego recorriéndolas con un bucle para imprimir cada una.

Ejercicio 32.3

Podés usar el método `replace()` para reemplazar cualquier palabra en una cadena por otra. Acá tenés un ejemplo rápido que muestra cómo reemplazar 'dog' por 'cat' en una oración:

```
>>> message = "Me gustan mucho los perros."
>>> message.replace('perros', 'gatos')
'Me gustan mucho los gatos.'
```

Leé cada línea del archivo `learning_python.txt` que creaste antes y reemplazá la palabra `Python` por el nombre de otro lenguaje, como `C`. Mostrá cada línea modificada en pantalla.