

# Ejercicios Adicionales Tema 4

## Tus mejores elecciones

---

- Crea una lista para almacenar tus preferencias (colores, presidentes, lo que sea)
- Para cada elección, muestra en la pantalla un mensaje como: "My #1 choice is blue."
- **Bonus:** Cambiar el mensaje para que muestre "My 1st choice", "My 2nd choice", "My 3rd choice", eligiendo el sufijo adecuado (en inglés) según el número.

## Rellenar lista

---

Escribe una función que cree una lista con un valor dado y el número de elementos a crear (repetidos).

Ejemplo de uso:

```
valueToFill = 'a'
lista = fill(valueToFill, 3) # ['a', 'a', 'a']
```

## Invertir lista

---

Escribe una función que invierta los elementos de una lista. El propósito del ejercicio es simplemente practicar el uso de listas. No usar la función reverse.

Ejemplo de uso:

```
mireverse([1,2,3]) # [3, 2, 1]
```

## Compactar lista

---

Escribe una función que “compacte” los elementos de una lista eliminando todos los elementos “innecesarios” (que evalúan a falso):

Ejemplo de uso:

```
compact([1,3,False,4,5,0]) // [1, 3, 4, 5]
```

## Lista sin duplicados

---

Escribir una función que devuelva una lista sin elementos duplicados.

Ejemplo de uso:

```
unique([1,2,1,2,2,2,2,3,4,5,3,3,5]); # [1, 2, 3, 4, 5]
```

## La Ficha para Recetas

---

*¡Nunca olvides otra receta!*

- Crea un diccionario para almacenar información de tu receta favorita. Debería tener propiedades para **título**, **raciones** e **ingredientes** (una lista de strings).
- Mostrar en líneas separadas, la información de la receta. Por ejemplo:
  - Mole
  - Raciones: 2
  - Ingredientes:
  - canela
  - comino
  - cacao

## La Lista de Lectura

---

*¡Una lista de qué libros has leído y cuales te gustaría leer!*

- Crea una lista de diccionarios, donde cada diccionario describe un libro y tiene propiedades para el **título**, **autor** y **leído** (indicando si ya se ha leído).
- Itera a través de la lista de libros. Para cada libro, mostrar el título y autor así: "El Hobbit por J.R.R. Tolkien".
- Ahora usa un if/else para indicar si el libro se ha leído o no. Si se ha leído, mostrar algo así como: '[Leído] El Hobbit por J.R.R. Tolkien', y si no, '[Pendiente de leer] El Señor de los Anillos por J.R.R. Tolkien.'

## La Base de Datos de Películas

*¡Como IMDB, pero mucho más pequeña!*

- Crea un diccionario para almacenar información sobre tu película favorita: **título**, **duración** y **actores** (una lista de strings).
- Crea una función para mostrar la información de la película recibiendo un diccionario de tipo película. Algo así como: "Puff the Magic Dragon dura 30 minutos. Actores: Puff, Jackie, Living Sneezes."

## Contando repeticiones

---

- Escribir una función que a partir de una lista devuelva un diccionario que incluya cuántas veces aparece cada elemento en una lista.
- Por ejemplo:

```
sample_list = [11, 45, 8, 11, 23, 45, 23, 45, 89]
Printing count of each item    {11: 2, 45: 3, 8: 1, 23: 2, 89: 1}
```

- Posteriormente, crea una tupla a partir de la lista sin duplicados.

## Guardando notas

---

- Escribir un programa en Python que permita guardar las notas obtenidas por los estudiantes. Para un mismo estudiante se pueden tener varias notas.
- En un diccionario, las claves serán los nombres de los estudiantes y los valores serán listas con las notas de cada estudiante. El programa pedirá por teclado repetidamente nombres de estudiantes y notas hasta que escriba 'fin'. Tras esto, se mostrará un listado con las notas de cada estudiante.

## Los meses del año

---

Escribir un programa en Python que almacene en una tupla los nombres de los meses del año y en otra tupla cuántos días tiene cada mes. Después, se pedirá al usuario que introduzca un número de mes y el programa responderá con un mensaje del tipo "Enero tiene 31 días". Si el mes es incorrecto, se mostrará un mensaje de error. El programa debe terminar cuando se introduzca el valor 0.

Bonus: escribir una función que recibe el número de mes del año y retorna una tupla con el nombre del mes y el número de días de ese mes.

## Sumando tuplas y diccionarios

---

- Escribe un programa que calcule la suma de los elementos de una tupla suponiendo que todos ellos son números válidos.
- Realizar lo mismo para los valores contenidos en un diccionario.

## Calculando precios

---

Escribe un programa que guarde en un diccionario los precios de ciertos productos mediante referencias. Después el programa pregunta al usuario una referencia y el número de unidades y si el producto existe se muestra el precio total. Si el producto no existe se muestra un mensaje indicativo.

## La Caja Registradora

---

Escribe una función llamada `cashRegister` que toma un diccionario lista de la compra. El diccionario contiene nombres de productos y precios (itemName: itemPrice). La función debería devolver el precio total de la lista de la compra.

Ejemplo

```
# Entrada
cartForParty = {
    banana: 1.25,
    handkerchief: 0.99,
    Tshirt: 25.01,
    apple: 0.60,
    nalgene: 10.34,
    proteinShake: 22.36
}

# Salida
print(cashRegister(cartForParty)) # 60.55
```