Reto 4: Or

1. Objetivos:

```
- Practicar el operador `or` y usarlo para realizar filtros más complejos
```

---

2. Desarrollo:

a) Filtrando palabras

Eres el organizador del Concurso Nacional de Deletreo "Salvador Novo". Por una bella coincidencia, este año el día del concurso cae justo el mismo día que el Día del Orgullo LGBT. Dado que Salvador Novo era homosexual, te parece muy apropiado que el concurso de deletreo funcione como una celebración de su "belicosa homosexualidad" (como la llamaba Carlos Monsiváis). Se te ocurre hacer lo siguiente:

De la lista de palabras que tenías originalmente para el concurso, vas a filtrar las palabras para que sólo tengas palabras que empiecen con "l" **o** con "g" **o** con "b" **o** con "t".

Aquí está tu lista original de palabras:

```
palabras = ['cabildo', 'genocidio', 'severo', 'jarana', 'enigmático', 'jaguar', 'solidaridad', 'reivindicar', 'bálsamo', 'panteón',
            'cabestrillo', 'boicotear', 'letargo', 'jaqueca', 'tentáculo', 'legislar', 'gnomo', 'blasfemia', 'camposanto',
            'factible', 'eficaz', 'sintonía', 'lloriquear', 'fachada', 'edificante', 'pétalo', 'libélula', 'pavimento', 'llovizna',
            'racimo', 'gargantilla', 'relieve', 'bóveda', 'tecnicismo', 'terraplén', 'basílica']
```

1. Escribe 4 funciones, para cada una de las letras del acrónimo LGBT. Las funciones van a regresar `True` sólo si la palabra comienza con la letra que le corresponde.

   Por ejemplo, la función `palabra_comienza_con_l` va a regresar `True` sólo si la palabra comienza con `l`.

2. Después, define una función que sea la unión de estas 4 funciones y regrese `True` si la palabra comienza con alguna de las letras del acrónimo LGBT.

3. Finalmente filtra la `lista palabras` para tener una nueva lista que será la `lista` usada para el concurso.

> **Tip #1**: Las `strings` pueden ser accedas igual que las listas, así que si quieres acceder a la primera letra de una palabra basta con usar `palabra[0]`, como si fuera el primer índice de una `lista`.

> **Tip #2**: Hasta ahora sólo hemos usando operadores `lógicos` con 1 o 2 comparaciones. Juntar más de dos comparaciones es tan fácil como escribir:

```
comparacion_1 or comparacion_2 or comparacion_3 or comparacion_4
```

```
# Aquí va la función para filtrar "l"
def comienza_con_l(palabra):
    return palabra[0] == "l"


# Aquí va la función para filtrar "g"
def comienza_con_g(palabra):
    return palabra[0] == "g"


# Aquí va la función para filtrar "b"
def comienza_con_b(palabra):
    return palabra[0] == "b"


# Aquí va la función para filtrar "t"
def comienza_con_t(palabra):
    return palabra[0] == "t"


# Aquí va la función para unir las 4 funciones anteriores
def comienza_con_lgbt(palabra):
    return comienza_con_l(palabra) or comienza_con_g(palabra) or comienza_con_b(palabra) or comienza_con_t(palabra)
```

```
palabras_filtradas = list(filter(comienza_con_lgbt, palabras))


print(f'==Concurso Nacional de Deletreo "Salvador Novo"==\n')
print(f'Lista oficial de palabras: {palabras_filtradas}')
```

```
==Concurso Nacional de Deletreo "Salvador Novo"==

Lista oficial de palabras: ['genocidio', 'bálsamo', 'boicotear', 'letargo', 'tentáculo', 'legislar', 'gnomo', 'blasfemia', 'lloriquear',
```

▶ Solución