

07.1.Ejercicios

August 11, 2024

1 Ejercicios

1.1 Listas por compresión

1. Crea una lista que contenga los cuadrados de los números del 1 al 10.

```
[ ]: cuadrados = [x**2 for x in range(1, 11)]  
print(cuadrados) # Salida: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

2. Genera una lista que contenga los números pares del 1 al 20.

```
[ ]: pares = [x for x in range(1, 21) if x % 2 == 0]  
print(pares) # Salida: [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

3. Dada una lista de palabras, crea una nueva lista que contenga la longitud de cada palabra.

```
[ ]: palabras = ["manzana", "banana", "cereza", "durazno"]  
longitudes = [len(palabra) for palabra in palabras]  
print(longitudes) # Salida: [7, 6, 6, 7]
```

[7, 6, 6, 7]

4. Crea una lista que contenga los números del 1 al 30 que sean divisibles por 3.

```
[ ]: divisibles_por_3 = [x for x in range(1, 31) if x % 3 == 0]  
print(divisibles_por_3) # Salida: [3, 6, 9, 12, 15, 18, 21, 24, 27, 30]
```

[3, 6, 9, 12, 15, 18, 21, 24, 27, 30]

5. Dada una palabra, crea una lista que contenga solo las vocales de la palabra.

```
[ ]: palabra = "computadora"  
vocales = [letra for letra in palabra if letra in 'aeiou']  
print(vocales) # Salida: ['o', 'u', 'a', 'o', 'a']
```

['o', 'u', 'a', 'o', 'a']

6. Crea una lista que contenga todos los números primos entre 1 y 50 usando una lista por comprensión.

```
[ ]: def es_primo(n):
    return n > 1 and all(n % i != 0 for i in range(2, int(n**0.5) + 1))

primos = [x for x in range(1, 51) if es_primo(x)]
print(primos) # Salida: [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]
```

[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]

7. Genera una lista de listas donde cada sublista contenga un número y su cuadrado, para los números del 1 al 5.

```
[ ]: lista_de_listas = [[x, x**2] for x in range(1, 6)]
print(lista_de_listas) # Salida: [[1, 1], [2, 4], [3, 9], [4, 16], [5, 25]]
```

[[1, 1], [2, 4], [3, 9], [4, 16], [5, 25]]

8. Dada una lista de palabras, crea una nueva lista que contenga solo aquellas palabras que tienen más de 5 letras.

```
[ ]: palabras = ["sol", "luna", "estrellas", "cometa", "galaxia"]
palabras_largas = [palabra for palabra in palabras if len(palabra) > 5]
print(palabras_largas) # Salida: ['estrellas', 'cometa', 'galaxia']
```

['estrellas', 'cometa', 'galaxia']

9. Genera una lista de tuplas, donde cada tupla contenga un número y su cubo, para los números del 1 al 5.

```
[ ]: lista_de_tuplas = [(x, x**3) for x in range(1, 6)]
print(lista_de_tuplas) # Salida: [(1, 1), (2, 8), (3, 27), (4, 64), (5, 125)]
```

[(1, 1), (2, 8), (3, 27), (4, 64), (5, 125)]

10. Dada una lista de listas, usa una lista por comprensión para aplanarla en una sola lista.

```
[ ]: lista_de_listas = [[1, 2, 3], [4, 5], [6, 7, 8, 9]]
lista_aplanada = [elemento for sublista in lista_de_listas for elemento in sublista]
print(lista_aplanada) # Salida: [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

[1, 2, 3, 4, 5, 6, 7, 8, 9]

1.2 Expresiones generadoras

11. Crea una expresión generadora que calcule la suma de los cuadrados de los números del 1 al 10.

```
[ ]: suma_cuadrados = sum(x**2 for x in range(1, 11))
print(suma_cuadrados) # Salida: 385
```

385

12. Genera una expresión generadora que produzca los números pares entre 1 y 20.

```
[ ]: pares = (x for x in range(1, 21) if x % 2 == 0)
print(*pares, sep=", ") # Salida: 2, 4, 6, 8, 10, 12, 14, 16, 18, 20
```

2, 4, 6, 8, 10, 12, 14, 16, 18, 20

13. Dada una lista de palabras, usa una expresión generadora para calcular la longitud total de todas las palabras.

```
[ ]: palabras = ["manzana", "banana", "cereza", "durazno"]
longitud_total = sum(len(palabra) for palabra in palabras)
print(longitud_total) # Salida: 26
```

26

14. Crea una expresión generadora que filtre los números divisibles por 3 en el rango de 1 a 30.

```
[ ]: divisibles_por_3 = (x for x in range(1, 31) if x % 3 == 0)
print(*divisibles_por_3, sep=", ") # Salida: 3, 6, 9, 12, 15, 18, 21, 24, 27, 30
```

3, 6, 9, 12, 15, 18, 21, 24, 27, 30

15. Genera una expresión generadora que calcule las raíces cuadradas de los números pares entre 1 y 20.

```
[ ]: import math
raices_cuadradas = (math.sqrt(x) for x in range(2, 21, 2))
print(*raices_cuadradas, sep=", ")
# Salida: 1.4142135623730951, 2.0, 2.8284271247461903, 3.4641016151377544, 4.0,
4.47213595499958, 5.0990195135927845, 5.656854249492381, 6.324555320336759,
7.0710678118654755
```

1.4142135623730951, 2.0, 2.449489742783178, 2.8284271247461903,
3.1622776601683795, 3.4641016151377544, 3.7416573867739413, 4.0,
4.242640687119285, 4.47213595499958

16. Dada una lista de números, utiliza una expresión generadora para calcular el producto de todos los elementos.

```
[ ]: import math
numeros = [1, 2, 3, 4, 5]
producto = math.prod(x for x in numeros)
print(producto) # Salida: 120
```

120

17. Dada una lista de cadenas, utiliza una expresión generadora para concatenarlas en una sola cadena.

```
[ ]: cadenas = ["Hola", " ", "Mundo", "!"]
      resultado = ''.join(cadena for cadena in cadenas)
      print(resultado) # Salida: "Hola Mundo!"
```

Hola Mundo!

18. Usa una expresión generadora para sumar los primeros N números primos.

```
[ ]: def es_primo(n):
      return n > 1 and all(n % i != 0 for i in range(2, int(n**0.5) + 1))

      N = 10
      primos_sum = sum(x for x in range(2, 100) if es_primo(x)) # Generamos una
      ↪ lista amplia para cubrir los primeros 10 primos
      print(primos_sum) # Salida: 129 (suma de los primeros 10 números primos)
```

1060

19. Dada una lista de palabras, utiliza una expresión generadora para filtrar solo aquellas palabras que tengan más de 5 letras.

```
[ ]: palabras = ["sol", "luna", "estrellas", "cometa", "galaxia"]
      palabras_largas = (palabra for palabra in palabras if len(palabra) > 5)
      print(*palabras_largas, sep=", ") # Salida: "estrellas", "cometa", "galaxia"
```

estrellas, cometa, galaxia

20. Usa una expresión generadora para generar números aleatorios entre 1 y 100, pero detente cuando el número generado supere 90.

```
[ ]: import random
      numeros_al_azar = (num for num in iter(lambda: random.randint(1, 100), None))
      for num in numeros_al_azar:
          print(num, end=", ")
          if num > 90:
              break # Detenemos la iteración cuando el número es mayor a 90
```

59, 45, 92,