

Python 2023

Ejemplos de uso de:

- list comprehension
- filter
- map
- lambda
- con listas y con diccionarios

En los ejemplos en algunos casos se pusieron diferentes formas de resolver el mismo ejercicio para comparar las resoluciones, en otros casos dejamos para que las resuelvan ustedes.

1. Crear una lista de números pares del 0 al 10:

con for:

```
In [4]: pares = []
for num in range(11):
    if num % 2 == 0:
        pares.append(num)
```

con list comprehension:

```
In [6]: pares = [num for num in range(11) if num % 2 == 0]
```

2. Crear una lista de los cuadrados de los números del 1 al 5:

```
In [10]: cuadrados = []
for num in range(1, 6):
    cuadrados.append(num**2)
```

```
In [11]: cuadrados
```

```
Out[11]: [1, 4, 9, 16, 25]
```

```
In [9]: cuadrados = [num**2 for num in range(1, 6)]
cuadrados
```

```
Out[9]: [1, 4, 9, 16, 25]
```

3. Crear una lista de las vocales en una cadena de texto:

```
In [18]: cadena = "Hola, ¿cómo estás?"
vocales = []
for letra in cadena.lower():
    if letra in "aeiou":
        vocales.append(letra)
```

Usando filter

```
In [16]: vocales = list(filter(lambda letra: letra in 'aeiou', cadena))
```

```
In [19]: vocales
```

```
Out[19]: ['o', 'a', 'o', 'e']
```

Usando list comprehension

```
In [12]: cadena = "Hola, ¿cómo estás?"  
vocales = [letra for letra in cadena.lower() if letra in "aeiou"]
```

4. Crear una lista de las palabras en una cadena de texto:

```
In [20]: cadena = "Esto es una cadena de texto"  
palabras = []  
for palabra in cadena.split():  
    palabras.append(palabra )  
palabras
```

```
Out[20]: ['Esto', 'es', 'una', 'cadena', 'de', 'texto']
```

```
In [21]: cadena = "Esto es una cadena de texto"  
palabras = [palabra for palabra in cadena.split()]  
palabras
```

```
Out[21]: ['Esto', 'es', 'una', 'cadena', 'de', 'texto']
```

5. Crear una lista de las letras únicas en una cadena de texto:

```
In [24]: cadena = "Esto es una cadena de texto abril 2023"  
letras_unicas = []  
for letra in set(cadena):  
    if letra.isalpha():  
        letras_unicas.append(letra)  
letras_unicas
```

```
Out[24]: ['n', 's', 'i', 'l', 'e', 'u', 'E', 'c', 'b', 't', 'r', 'a', 'd', 'x', 'o']
```

```
In [25]: cadena = "Esto es una cadena de texto 2023"  
letras_unicas = [letra for letra in set(cadena) if letra.isalpha()]  
letras_unicas
```

```
Out[25]: ['d', 'u', 'n', 's', 't', 'E', 'x', 'e', 'a', 'c', 'o']
```

6. Crear una lista de los elementos comunes en dos listas:

```
In [27]: lista1 = [1, 2, 3, 4, 5]  
lista2 = [3, 4, 5, 6, 7]  
comunes = []  
for num in lista1:  
    if num in lista2:  
        comunes.append(num )  
comunes
```

```
Out[27]: [3, 4, 5]
```

Usando filter

```
In [31]: lista1 = [1, 2, 3, 4, 5]  
lista2 = [3, 4, 5, 6, 7]  
comunes = list(filter(lambda num: num in lista2, lista1))  
comunes
```

```
Out[31]: [3, 4, 5]
```

```
In [28]: lista1 = [1, 2, 3, 4, 5]
        lista2 = [3, 4, 5, 6, 7]
        comunes = [num for num in lista1 if num in lista2]
```

7. Conversión de cadenas a enteros

```
In [32]: str_nums = ['1', '2', '3', '4', '5']
        nums = []
        for str_num in str_nums:
            nums.append(int(str_num))
        print(nums)
```

```
[1, 2, 3, 4, 5]
```

```
In [33]: str_nums = ['1', '2', '3', '4', '5']
        nums = list(map(lambda x: int(x), str_nums))
        print(nums)
```

```
[1, 2, 3, 4, 5]
```

8. Selección de elementos de una lista

```
In [34]: nums = [1, 2, 3, 4, 5]
        even_nums = []
        for num in nums:
            if num % 2 == 0:
                even_nums.append(num)
        print(even_nums)
```

```
[2, 4]
```

```
In [35]: nums = [1, 2, 3, 4, 5]
        even_nums = list(filter(lambda x: x % 2 == 0, nums))
        print(even_nums)
```

```
[2, 4]
```

9. Suma de elementos de dos listas

```
In [36]: list1 = [1, 2, 3, 4, 5]
        list2 = [6, 7, 8, 9, 10]
        sums = []
        for i in range(len(list1)):
            sums.append(list1[i] + list2[i])
        print(sums)
```

```
[7, 9, 11, 13, 15]
```

```
In [40]: list1 = [1, 2, 3, 4, 5]
        list2 = [6, 7, 8, 9, 10]
        sums = list(map(lambda x, y: x + y, list1, list2))
        print(sums)
```

```
[7, 9, 11, 13, 15]
```

10. Conversión de grados Celsius a Fahrenheit

```
In [41]: celsius_temps = [0, 10, 20, 30, 40]
        fahrenheit_temps = []
        for temp in celsius_temps:
            fahrenheit_temps.append((9/5)*temp + 32)
        print(fahrenheit_temps)
```

```
[32.0, 50.0, 68.0, 86.0, 104.0]
```

```
In [38]: celsius_temps = [0, 10, 20, 30, 40]
fahrenheit_temps = list(map(lambda x: (9/5)*x + 32, celsius_temps))
print(fahrenheit_temps)
```

```
[32.0, 50.0, 68.0, 86.0, 104.0]
```

11. Encontrar los valores máximos en una lista de listas:

```
In [42]: data = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
max_values = []
for sublist in data:
    max_value = max(sublist)
    max_values.append(max_value)
max_values
```

```
[3, 6, 9]
```

```
In [43]: data = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
max_values = list(map(lambda x: max(x), data))
max_values
```

```
[3, 6, 9]
```

12. Convertir los valores de un diccionario en enteros:

```
In [44]: my_dict = {"a": "1", "b": "2", "c": "3"}

result = dict(map(lambda x: (x[0], int(x[1])), my_dict.items()))

result
```

```
Out[44]: {'a': 1, 'b': 2, 'c': 3}
```

13. Multiplicar los valores de un diccionario por una constante:

```
In [46]: my_dict = {"a": 1, "b": 2, "c": 3}
constant = 2

result = dict(map(lambda x: (x[0], x[1]*constant), my_dict.items()))

result
```

```
Out[46]: {'a': 2, 'b': 4, 'c': 6}
```

14. Convertir las claves de un diccionario a mayúsculas:

```
In [48]: my_dict = {"a": 1, "b": 2, "c": 3}

result = dict(map(lambda x: (x[0].upper(), x[1]), my_dict.items()))

result
```

```
Out[48]: {'A': 1, 'B': 2, 'C': 3}
```

15. Filtrar un diccionario para eliminar valores por debajo de un cierto umbral:

```
In [50]: my_dict = {"a": 1, "b": 2, "c": 3, "d": 4}
         threshold = 3

         result = dict(filter(lambda x: x[1] >= threshold, my_dict.items()))

         result
```

```
Out[50]: {'c': 3, 'd': 4}
```

16. Obtener el valor máximo de un diccionario:

```
In [52]: my_dict = {"a": 1, "b": 2, "c": 3}

         result = max(my_dict.values())

         result
```

```
Out[52]: 3
```

17. Obtener la clave del valor máximo en un diccionario:

```
In [55]: my_dict = {"a": 1, "b": 2, "c": 3}

         result = max(my_dict.items(), key=lambda x: x[1])[0]

         result
```

```
Out[55]: 'c'
```

18. Obtener la suma de los valores en un diccionario:

```
In [56]: my_dict = {"a": 1, "b": 2, "c": 3}

         result = sum(my_dict.values())

         result
```

```
Out[56]: 6
```

19. Eliminar todas las claves que contengan la letra "a":

```
In [59]: my_dict = {"manzana": 1, "banana": 2, "tomate": 3}

         result = dict(filter(lambda x: "e" not in x[0], my_dict.items()))

         result
```

```
Out[59]: {'manzana': 1, 'banana': 2}
```

20. Unir dos diccionarios en uno solo:

```
In [61]: diccionario1 = {"a": 1, "b": 2}
         diccionario2 = {"c": 3, "d": 4}
         diccionario_final = dict(map(lambda x: (x[0], x[1]), list(diccionario1.items()) + list(diccionario2.items())))

         diccionario_final
```

```
Out[61]: {'a': 1, 'b': 2, 'c': 3, 'd': 4}
```

21. Obtener el valor mínimo de un diccionario:

```
In [63]: diccionario = {"a": 1, "b": 2, "c": 3}
minimo = min(list(map(lambda x: x[1], diccionario.items())))
minimo
```

Out[63]: 1