

## Problemas de Comprensión de Listas, Tuplas y Diccionarios

1. Usa una list comprehension para crear e imprimir una lista de enteros consecutivos que comience con 1 y termine con 50.

*# tu solución aquí*

```
lista_consecutivos = [i for i in range(1, 51)]
```

```
print(lista_consecutivos)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50]
```

1. Usa una list comprehension para crear e imprimir una lista de números pares entre 2 y 200.

*# tu solución aquí*

```
lista_pares = [i for i in range(2, 201) if i % 2 == 0]
```

```
print(lista_pares)
```

```
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 166, 168, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192, 194, 196, 198, 200]
```

1. Dado el siguiente array de 10 x 4:

```
array_10x4 = [ [0.1, 0.2, 0.3, 0.4], [0.5, 0.6, 0.7, 0.8], [0.9, 1.0, 1.1, 1.2], [1.3, 1.4, 1.5, 1.6], [1.7, 1.8, 1.9, 2.0], [2.1, 2.2, 2.3, 2.4], [2.5, 2.6, 2.7, 2.8], [2.9, 3.0, 3.1, 3.2], [3.3, 3.4, 3.5, 3.6], [3.7, 3.8, 3.9, 4.0] ]
```

Usa una comprensión de lista para crear e imprimir una lista que contenga todos los elementos de este array (aplanar la lista).

*# tu solución aquí*

```
array_10x4 = [
    [0.1, 0.2, 0.3, 0.4],
    [0.5, 0.6, 0.7, 0.8],
    [0.9, 1.0, 1.1, 1.2],
    [1.3, 1.4, 1.5, 1.6],
    [1.7, 1.8, 1.9, 2.0],
    [2.1, 2.2, 2.3, 2.4],
```

```

        [2.5, 2.6, 2.7, 2.8],
        [2.9, 3.0, 3.1, 3.2],
        [3.3, 3.4, 3.5, 3.6],
        [3.7, 3.8, 3.9, 4.0]
    ]

    lista_aplanada = [elemento for fila in array_10x4 for elemento in
fila]

    print(lista_aplanada)

[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4,
1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8,
2.9, 3.0, 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 4.0]

```

1. Añade una condición a la comprensión de lista anterior para que solo se impriman los valores mayores o iguales a 0.5.

```

# tu solución aquí

lista_filtrada = [elemento for fila in array_10x4 for elemento in fila
if elemento >= 0.5]

print(lista_filtrada)

[0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8,
1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0, 3.1, 3.2,
3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 4.0]

```

1. Dado el siguiente array de 5 x 2 x 3:

```
array_5x2x3 = [ [[0.1, 0.2, 0.3], [0.4, 0.5, 0.6]], [[0.7, 0.8, 0.9], [1.0, 1.1, 1.2]], [[1.3, 1.4,
1.5], [1.6, 1.7, 1.8]], [[1.9, 2.0, 2.1], [2.2, 2.3, 2.4]], [[2.5, 2.6, 2.7], [2.8, 2.9, 3.0]] ]
```

Usa una comprensión de lista para crear e imprimir una lista que contenga todos los elementos de este array (aplanar la lista).

```

# tu solución aquí

array_5x2x3 = [
    [[0.1, 0.2, 0.3], [0.4, 0.5, 0.6]],
    [[0.7, 0.8, 0.9], [1.0, 1.1, 1.2]],
    [[1.3, 1.4, 1.5], [1.6, 1.7, 1.8]],
    [[1.9, 2.0, 2.1], [2.2, 2.3, 2.4]],
    [[2.5, 2.6, 2.7], [2.8, 2.9, 3.0]]
]

lista_aplanada = [elemento for sublista1 in array_5x2x3 for sublista2
in sublista1 for elemento in sublista2]

```

```
print(lista_aplanada)
```

```
[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0]
```

1. Añade una condición a la comprensión de lista anterior para que se imprima el último valor en cada subarray, pero solo si es menor o igual a 1.

```
# tu solución aquí
```

```
lista_filtrada = [sublista[-1] for sublista in array_5x2x3 for  
sublista in sublista if sublista[-1] <= 1]
```

```
print(lista_filtrada)
```

```
[0.3, 0.6, 0.9]
```

1. Usa una comprensión de lista para crear una lista de los primeros 10 números impares, comenzando desde 1.

```
# tu solución aquí
```

```
numeros_impares = [i for i in range(1, 20) if i % 2 != 0]
```

```
print(numeros_impares)
```

```
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19]
```

## Medio

1. Dada la cadena "programming", usa una comprensión de diccionario para contar la frecuencia de cada letra en la cadena.

```
# tu solución aquí
```

```
cadena = "programming"
```

```
frecuencia_letras = {letra: cadena.count(letra) for letra in  
set(cadena)}
```

```
print(frecuencia_letras)
```

```
{'g': 2, 'i': 1, 'p': 1, 'n': 1, 'm': 2, 'o': 1, 'r': 2, 'a': 1}
```

1. Crea una lista de tuplas que contenga cada número del 1 al 10 y su cuadrado utilizando una comprensión de lista.

```
# tu solución aquí
```

```
lista_tuplas = [(i, i**2) for i in range(1, 11)]
```

```
print(lista_tuplas)
[(1, 1), (2, 4), (3, 9), (4, 16), (5, 25), (6, 36), (7, 49), (8, 64),
(9, 81), (10, 100)]
```

1. Dada la lista mixta:

```
lista_mixta = [1, 'hola', 3, 'mundo', 5, 'python', 7, 'programacion']
```

Usa una comprensión de lista para extraer solo los elementos que son cadenas.

```
# tu solución aquí
lista_mixta = [1, 'hola', 3, 'mundo', 5, 'python', 7, 'programacion']
lista_cadenas = [elemento for elemento in lista_mixta if
isinstance(elemento, str)]
print(lista_cadenas)
['hola', 'mundo', 'python', 'programacion']
```

1. Dada la lista de números:

```
numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Crea un diccionario que contenga el cuadrado de cada número utilizando una comprensión de diccionario.

```
# tu solución aquí
numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
diccionario_cuadrados = {numero: numero**2 for numero in numeros}
print(diccionario_cuadrados)
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100}
```

1. Dada una lista de diccionarios:

```
estudiantes = [{ 'nombre': 'Juan', 'edad': 20}, { 'nombre': 'Ana', 'edad': 17}, { 'nombre':
'Luis', 'edad': 19}, { 'nombre': 'Pedro', 'edad': 22}, { 'nombre': 'Maria', 'edad': 15} ]
```

Usa una comprensión de lista para crear una lista de los nombres de los estudiantes que tienen una edad mayor a 18.

```
# tu solución aquí
```

```

estudiantes = [
    {'nombre': 'Juan', 'edad': 20},
    {'nombre': 'Ana', 'edad': 17},
    {'nombre': 'Luis', 'edad': 19},
    {'nombre': 'Pedro', 'edad': 22},
    {'nombre': 'Maria', 'edad': 15}
]

nombres_mayores_18 = [estudiante['nombre'] for estudiante in
estudiantes if estudiante['edad'] > 18]

print(nombres_mayores_18)

['Juan', 'Luis', 'Pedro']

```

1. Dada la lista:

```
mezclados = ['a', 1, 'b', 2, 'c', 3, 'd', 4]
```

Crea una lista que contenga solo los números usando una comprensión de lista.

```

# tu solución aquí

mezclados = ['a', 1, 'b', 2, 'c', 3, 'd', 4]

numeros = [elemento for elemento in mezclados if isinstance(elemento,
int)]

print(numeros)

[1, 2, 3, 4]

```

## Avanzado

1. Dada la lista de temperaturas en Celsius:

```
temperaturas_celsius = [0, 20, 100, -10, 37]
```

Usa una comprensión de lista para convertirlas a Fahrenheit y crear una lista de tuplas donde cada tupla contenga la temperatura en Celsius y su equivalente en Fahrenheit.

```

# tu solución aquí

temperaturas_celsius = [0, 20, 100, -10, 37]

temperaturas_fahrenheit = [(c, (c * 9/5) + 32) for c in
temperaturas_celsius]

print(temperaturas_fahrenheit)

```

```
[(0, 32.0), (20, 68.0), (100, 212.0), (-10, 14.0), (37, 98.6)]
```

1. Dada la lista de estudiantes con sus calificaciones:

```
estudiantes = [ {'nombre': 'Juan', 'calificacion': 85}, {'nombre': 'Ana', 'calificacion': 60}, {'nombre': 'Luis', 'calificacion': 75}, {'nombre': 'Pedro', 'calificacion': 50}, {'nombre': 'Maria', 'calificacion': 90} ]
```

Crea un diccionario que contenga los nombres de los estudiantes como claves y sus calificaciones como valores, y filtra aquellos que tienen una calificación menor a 70, almacenando los resultados en una lista de tuplas.

```
# tu solución aquí
```

```
estudiantes = [
    {'nombre': 'Juan', 'calificacion': 85},
    {'nombre': 'Ana', 'calificacion': 60},
    {'nombre': 'Luis', 'calificacion': 75},
    {'nombre': 'Pedro', 'calificacion': 50},
    {'nombre': 'Maria', 'calificacion': 90}
]

diccionario_estudiantes = {est['nombre']: est['calificacion'] for est
in estudiantes}

resultado = [(nombre, calificacion) for nombre, calificacion in
diccionario_estudiantes.items() if calificacion >= 70]

print(diccionario_estudiantes)

{'Juan': 85, 'Ana': 60, 'Luis': 75, 'Pedro': 50, 'Maria': 90}

print(resultado)

[('Juan', 85), ('Luis', 75), ('Maria', 90)]
```

1. Dada la lista de números:

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

Usa una comprensión de lista para generar una lista de cadenas que contenga la representación en texto de cada número, y crea una lista de tuplas donde cada tupla contenga el número y su representación en texto.

```
# tu solución aquí
```

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

nombres_numeros = {
```

```

1: 'uno',
2: 'dos',
3: 'tres',
4: 'cuatro',
5: 'cinco'
}

resultado = [(num, nombres_numeros[num]) for num in numeros]

print(resultado)

[(1, 'uno'), (2, 'dos'), (3, 'tres'), (4, 'cuatro'), (5, 'cinco')]

```

1. Crea una lista de tuplas que contenga solo los números de Fibonacci menores a 100 utilizando una comprensión de lista, comenzando desde 0, y en cada tupla, incluye el índice del número en la secuencia.

```

# tu solución aquí

fibonacci = [0, 1]

while True:
    siguiente = fibonacci[-1] + fibonacci[-2]
    if siguiente >= 100:
        break
    fibonacci.append(siguiente)

resultado = [(i, num) for i, num in enumerate(fibonacci)]

print(resultado)

[(0, 0), (1, 1), (2, 1), (3, 2), (4, 3), (5, 5), (6, 8), (7, 13), (8, 21), (9, 34), (10, 55), (11, 89)]

```

1. Dado el diccionario:

```
datos = {'a': 5, 'b': 15, 'c': 8, 'd': 12}
```

Usa una comprensión de diccionario para crear un nuevo diccionario que contenga solo las claves y valores donde los valores sean mayores a 10, y almacena los resultados en una lista de tuplas.

```

# tu solución aquí

datos = {'a': 5, 'b': 15, 'c': 8, 'd': 12}

nuevo_diccionario = {k: v for k, v in datos.items() if v > 10}

resultado = [(k, v) for k, v in nuevo_diccionario.items()]

```

```
print(nuevo_diccionario)
{'b': 15, 'd': 12}
print(resultado)
[('b', 15), ('d', 12)]
```

1. Dada la lista de listas:

```
listas_numeros = [ [1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12] ]
```

Usa una comprensión de lista anidada para extraer todos los elementos que son divisibles por 3 y crea una lista de tuplas donde cada tupla contenga el número y su lista de origen.

```
# tu solución aquí
listas_numeros = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9],
    [10, 11, 12]
]

resultado = [(num, sublista) for sublista in listas_numeros for num in
sublista if num % 3 == 0]

print(resultado)

[(3, [1, 2, 3]), (6, [4, 5, 6]), (9, [7, 8, 9]), (12, [10, 11, 12])]
```

1. Dado el texto:

texto = "El sol brilla y la luna brilla en la noche. El sol es hermoso y la luna también es hermosa. En un día claro, el sol ilumina el paisaje y la luna refleja su luz en el agua. A veces, el cielo está nublado y el sol se oculta detrás de las nubes, pero la luna sigue brillando. Las estrellas también son visibles en la oscuridad, creando un hermoso espectáculo en el cielo. La luna llena es especialmente mágica y muchas personas la admiran en silencio. El ciclo de la luna, desde la luna nueva hasta la luna llena, es fascinante para los astrónomos y amantes de la naturaleza."

Crea un diccionario que contenga las palabras del texto y el número de veces que aparecen, utilizando una comprensión de diccionario, ignorando mayúsculas y minúsculas, y almacena los resultados en una lista de tuplas con la palabra y su frecuencia.



```
# tu solución aquí
```

```
texto = """El sol brilla y la luna brilla en la noche. El sol es  
hermoso y la luna también es hermosa.  
En un día claro, el sol ilumina el paisaje y la luna refleja su luz en  
el agua.  
A veces, el cielo está nublado y el sol se oculta detrás de las nubes,  
pero la luna sigue brillando.  
Las estrellas también son visibles en la oscuridad, creando un hermoso  
espectáculo en el cielo.  
La luna llena es especialmente mágica y muchas personas la admiran en  
silencio.  
El ciclo de la luna, desde la luna nueva hasta la luna llena, es  
fascinante para los astrónomos y amantes de la naturaleza."""
```

```
palabras = texto.lower().replace('.', '').replace(',', '').split()
```

```
frecuencias = {palabra: palabras.count(palabra) for palabra in  
set(palabras)}
```

```
resultado = [(palabra, frecuencias[palabra]) for palabra in  
frecuencias]
```

```
print(frecuencias)
```

```
{'espectáculo': 1, 'las': 2, 'brillando': 1, 'admiran': 1,  
'naturaleza': 1, 'creando': 1, 'su': 1, 'ciclo': 1, 'detrás': 1,  
'oculta': 1, 'la': 12, 'y': 6, 'paisaje': 1, 'nubes': 1, 'se': 1,  
'agua': 1, 'para': 1, 'en': 6, 'fascinante': 1, 'nueva': 1, 'está': 1,  
'ilumina': 1, 'hermosa': 1, 'sol': 4, 'claro': 1, 'visibles': 1,  
'oscuridad': 1, 'son': 1, 'hermoso': 2, 'silencio': 1, 'hasta': 1,  
'un': 2, 'nublado': 1, 'desde': 1, 'veces': 1, 'astrónomos': 1, 'el':  
9, 'personas': 1, 'amantes': 1, 'también': 2, 'luz': 1, 'noche': 1,  
'sigue': 1, 'especialmente': 1, 'pero': 1, 'brilla': 2, 'cielo': 2,  
'es': 4, 'día': 1, 'los': 1, 'muchas': 1, 'de': 3, 'llena': 2,  
'mágica': 1, 'luna': 8, 'refleja': 1, 'a': 1, 'estrellas': 1}
```

```
print(resultado)
```

```
[('espectáculo', 1), ('las', 2), ('brillando', 1), ('admiran', 1),  
(('naturaleza', 1), ('creando', 1), ('su', 1), ('ciclo', 1), ('detrás',  
1), ('oculta', 1), ('la', 12), ('y', 6), ('paisaje', 1), ('nubes', 1),  
(('se', 1), ('agua', 1), ('para', 1), ('en', 6), ('fascinante', 1),  
(('nueva', 1), ('está', 1), ('ilumina', 1), ('hermosa', 1), ('sol', 4),  
(('claro', 1), ('visibles', 1), ('oscuridad', 1), ('son', 1),  
(('hermoso', 2), ('silencio', 1), ('hasta', 1), ('un', 2), ('nublado',  
1), ('desde', 1), ('veces', 1), ('astrónomos', 1), ('el', 9),  
(('personas', 1), ('amantes', 1), ('también', 2), ('luz', 1), ('noche',  
1), ('sigue', 1), ('especialmente', 1), ('pero', 1), ('brilla', 2),  
(('cielo', 2), ('es', 4), ('día', 1), ('los', 1), ('muchas', 1), ('de',
```

```
3), ('llena', 2), ('mágica', 1), ('luna', 8), ('refleja', 1), ('a',  
1), ('estrellas', 1)]
```