

## ✓ LISTAS DE COMPRESIÓN

### Generamos una lista con los caracteres de una cadena

```

1 def lista_letras(palabra):
2     lista = []
3     for caracter in palabra:
4         lista.append(caracter)
5     return lista
6
7 def main():
8     cadena = input("Dame una cadena: ")
9     lista = lista_letras(cadena)
10    print(lista)
11
12 main()

```

↗ Dame una cadena: hola a todos  
['h', 'o', 'l', 'a', ' ', ' ', 'a', ' ', ' ', 't', 'o', 'd', 'o', 's']

```

1 # Listas de compresión
2 def lista_letras2(palabra):
3     lista = [caracter for caracter in palabra]
4     return lista
5
6 def main():
7     cadena = input("Dame una cadena: ")
8     lista = lista_letras2(cadena)
9     print(lista)
10
11 main()

```

↗ Dame una cadena: hola a todos  
['h', 'o', 'l', 'a', ' ', ' ', 'a', ' ', ' ', 't', 'o', 'd', 'o', 's']

### Generar una lista con números del 0 al 10

```

1 def lista_0_10():
2     lista = []
3     for i in range(0, 11):
4         lista.append(i)
5     return lista
6
7 def main():
8     lista = lista_0_10()
9     print(lista)
10
11 main()

```

↗ [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

```

1 # Lista de compresión
2 def lista_0_10():
3     lista = [i for i in range(0, 11)]
4     return lista
5
6 def main():
7     lista = lista_0_10()
8     print(lista)
9
10 main()

```


↗ [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

**Elevar al cuadrado los números del 0 a 10**

```

1 def lista_cuadrado():
2     lista = []
3     for i in range(0, 11):
4         lista.append(i**2)
5     return lista
6
7 def main():
8     lista = lista_cuadrado()
9     print(lista)
10
11 main()


```

 [0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

```

1 def lista_cuadrado():
2     lista = [i**2 for i in range(0, 11)]
3     return lista
4
5 def main():
6     lista = lista_cuadrado()
7     print(lista)
8
9 main()

```


 [0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

**Genera una lista con los valores pares entre 0 y 10**

```

1 def lista_pares():
2     lista = []
3     for i in range(0, 11):
4         if i % 2 == 0:
5             lista.append(i)
6     return lista
7
8 def main():
9     lista = lista_pares()
10    print(lista)
11
12 main()


```

 [0, 2, 4, 6, 8, 10]

```

1 # Listas de compresión
2 def lista_pares():
3     lista = [i for i in range(0, 11) if i % 2 == 0]
4     return lista
5
6 def main():
7     lista = lista_pares()
8     print(lista)
9
10 main()

```

 [0, 2, 4, 6, 8, 10]

**Evalua x a la 2 + 2 del -10 al 10 con paso de 2 y guardo el resultado en una lista**

```


1 def funcion():
2     lista = []
3     for i in range(-10, 11, 2):
4         lista.append(i**2 + 2)

```

```

5  return lista
6
7  def main():
8      lista = funcion()
9      print(lista)
10
11 main()


```

 [102, 66, 38, 18, 6, 2, 6, 18, 38, 66, 102]

```

1 # Lista de compresión
2 def funcion():
3     lista = [i**2 + 2 for i in range(-10, 11, 2)]
4     return lista
5
6 def main():
7     lista = funcion()
8     print(lista)
9
10 main()

```

 [102, 66, 38, 18, 6, 2, 6, 18, 38, 66, 102]


## ✓ Laboratorio: Listas de compresión

- Juan Pérez
- A08888888

```

1 def mayor_numero(num):
2     lista = []
3     for i in range(0, 11):
4         if i > num:
5             lista.append(i)
6     return lista
7
8 def main():
9     n = int(input("Dame un número entre 0 y 10: "))
10    lista = mayor_numero(n)
11    print(lista)
12
13 main()


```

 Dame un número entre 0 y 10: 2  
[3, 4, 5, 6, 7, 8, 9, 10]

```

1 # Listas de compresión
2 def mayor_numero2(num):
3     lista = [i for i in range(0, 11) if i > num]
4     return lista
5
6 def main():
7     n = int(input("Dame un número entre 0 y 10: "))
8     lista = mayor_numero2(n)
9     print(lista)
10
11 main()

```

 Dame un número entre 0 y 10: 2  
[3, 4, 5, 6, 7, 8, 9, 10]

### Regrea una lista con las frutas con determinada letra

```

1 def frutas_con_letra(lista_frutas, letra):
2     lista = []

```

```

3  for fruta in lista_frutas:
4      if letra.upper() in fruta.upper():
5          lista.append(fruta)
6  return lista
7
8 def main():
9  frutas = ["Manzana", "Plátano", "Sandía", "Melón", "Papaya"]
10 letra = input("Dame una letra: ")
11 lista = frutas_con_letra(frutas, letra)
12 print(lista)
13
14 main()

```

↳ Dame una letra: l  
['Plátano', 'Melón']

```

1 def frutas_con_letra(lista_frutas, letra):
2     lista = [fruta for fruta in lista_frutas if letra.upper() in fruta.upper()]
3     return lista
4
5 def main():
6     frutas = ["Manzana", "Plátano", "Sandía", "Melón", "Papaya"]
7     letra = input("Dame una letra: ")
8     lista = frutas_con_letra(frutas, letra)
9     print(lista)
10
11 main()

```

↳ Dame una letra: a  
['Manzana', 'Plátano', 'Sandía', 'Papaya']

### Sustituye frutas

```

1 def sustituye_fruta(lista_frutas, fruta1, fruta2):
2     lista = []
3     for fruta in lista_frutas:
4         if fruta.lower() == fruta1.lower():
5             lista.append(fruta2)
6         else:
7             lista.append(fruta)
8     return lista
9
10 def main():
11     frutas = ["Manzana", "Plátano", "Sandía", "Melón", "Papaya"]
12     f1 = input("Dame la fruta 1: ")
13     f2 = input("Dame la fruta 2: ")
14     lista = sustituye_fruta(frutas, f1, f2)
15     print(lista)
16
17 main()

```

↳ Dame la fruta 1: sandía  
Dame la fruta 2: Uvas  
['Manzana', 'Plátano', 'Uvas', 'Melón', 'Papaya']

```

1 # Listas de compresión
2 def sustituye_fruta(lista_frutas, fruta1, fruta2):
3     lista = [fruta2 if fruta.lower() == fruta1.lower() else fruta for fruta in lista_frutas]
4     return lista
5
6 def main():
7     frutas = ["Manzana", "Plátano", "Sandía", "Melón", "Papaya"]
8     f1 = input("Dame la fruta 1: ")
9     f2 = input("Dame la fruta 2: ")
10    lista = sustituye_fruta(frutas, f1, f2)
11    print(lista)
12

```

```
13 main()
```

```
↵ Dame la fruta 1: melón  
Dame la fruta 2: Mamey  
['Manzana', 'Plátano', 'Sandía', 'Mamey', 'Papaya']
```