

estudiada. incluir un ejemplo por cada una.

```
In [6]: # Ejercicio 1 sin Lambda ()

numero = 9
obtener_raiz = numero ** 0.5
print(f'La raiz cuadrada del {numero} es : {obtener_raiz}')

La raiz cuadrada del 9 es : 3.0
```

```
In [11]: # Ejercicio 1 con Lambda()

x= lambda raiz: raiz ** 0.5
print(x(9))

3.0
```

```
In [14]: # ejercicios sin map()

frase = "Ejercicios del modulo 11 de EBAC"
div_palabra = frase.split() # Dividimos la cadena en palabras
largo = len(div_palabra) # Encuentra la longitud de la lista de palabras

print("La frase se divide en :", largo)

La frase se divide en : 6
```

```
In [17]: # ejercicio con map()

def large(frase):
    div_palabra = frase.split()
    return len(div_palabra)

frase = ['Ejercicios del modulo 11 de EBAC']
largo = list(map(large, frase))
print('La frase se divide en :',largo)

La frase se divide en : [6]
```

In [36]: *# ejercicio sin reduce()*

```
lista_nums = [5,2,7,8]

total = 1

for numero in lista_nums:
    total *= numero
print(total)
```

560

In [28]: *# ejercicio con reduce()*

```
from functools import reduce

def producto(a, b):
    return a * b

lista_num = [5,2,7,8]

reduce(producto, lista_num)
```

Out[28]: 560

In [55]: *# ejercicio 4 sin filter*

```
def extract_num_mayu(listado):
    num_mayus = []
    for elemento in listado:
        if isinstance(elemento, str) and elemento[0].isupper():
            num_mayus.append(elemento)
        elif isinstance(elemento, int):
            num_mayus.append(elemento)
    return num_mayus

palabras = ['tv','flor','Hola', 'Perro', 1, 45, 'Ebac', 'juego','fruta','Raul',7,'mesa']
resultado = extract_num_mayu(palabras)
print(resultado)

['Hola', 'Perro', 1, 45, 'Ebac', 'Raul', 7]
```

In [59]: *# ejercicio usando filter*

```
palabras = ['tv','flor','Hola', 'Perro', 1, 45, 'Ebac', 'juego','fruta','Raul',7,'mesa']

extraer = list(filter(lambda x: (isinstance(x, str) and x[0].isupper()) or isinstance(x, int), palabras))
extraer
```

Out[59]: ['Hola', 'Perro', 1, 45, 'Ebac', 'Raul', 7]

