



UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

**Centro de
e-Learning**

UNIDAD DIDÁCTICA IV
DIPLOMATURA EN PYTHON

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning

Módulo II – Nivel Inicial I

Unidad IV.

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning



Ejercicios:

Para trabajar con funciones lambda se debe guardar en una variable y luego pasar los parámetros de entrada a la variable como al llamar a una función, de esta forma:

```
variable = lambda parámetro1, .....,parametron : operación_sobre_parametros  
print(variable(parámetro1, .....,parametroN))
```

Ejercicio 1 – Dificultad baja

Cree una función lambda que compruebe si un número es par o impar.

Ejercicio 2 – Dificultad media

Crear una función lambda que sea equivalente a la siguiente función:

```
def multiplicar_por_tres(valor):  
    res = 3 * valor  
    return res
```

Ejercicio 3 – Dificultad media

Crear una función lambda que sea equivalente a la siguiente función:

```
def sumar(valor1, valor2):  
    res = valor1 + valor2  
    return res
```

Ejercicio 4 – Dificultad alta

Crear una función lambda que tome como parámetro una frase y la escriba al revés.



Ejercicio 5 – Dificultad media

Cree un programa que utilizando una función, solicite la edad de la persona e imprima todos los años que la persona ha cumplido según el siguiente formato de ejemplo.

1, 2, 3, 4, 5

Y

5, 4, 3, 2, 1

Ejercicio 6 – Dificultad alta

Cree una función que tome la siguiente lista y reordene de menor a mayor sus componentes:

[3, 44, 21, 78, 5, 56, 9]

Ejercicio 7 – Dificultad muy alta

isinstance(x, list) permite consultar si el elementos x es del tipo lista.

A partir de la siguiente función recursiva que permite recorrer los niveles de una lista:

```
lista = ["elemento1n1", "elemento2n1", "elemento3n1",  
["elemento1n2", "elemento2n2", "elemento3n2",  
["elemento1n3", "elemento2n3", "elemento3n3"]]]
```

```
def recorre_lista(item):  
    for x in item:  
        if isinstance(x, list):  
            recorrer_lista(x)  
        else:  
            print(x)
```

```
recorrer_lista(lista)
```

Optimice el código de forma que el programa considere:

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning



- Un valor de lista por defecto
- Permita tomar un segundo parámetro que lleve un registro del nivel en el cual se encuentra (en qué grado del anidamiento)
- Permita tomar un valor por defecto de cero para el parámetro anterior.
- Presente la salida según el siguiente formato:

```
elemento1n1
elemento2n1
elemento3n1
  elemento1n2
  elemento2n2
  elemento3n2
    elemento1n3
    elemento2n3
    elemento3n3
```