



FUNCIONES ANÓNIMAS – LAMBDA – ARGUMENTOS INDETERMINADOS

- Haremos uso de funciones anónimas con parámetros, sin especificar cuántos por posición ni nombre.

```
por_posicion_argumentos = lambda *args : len(args)
print(por_posicion_argumentos(2, 2, 34, 4)) # 4

por_nombre_argumentos = lambda **kwargs : kwargs
print(por_nombre_argumentos(name="Anartz", lastname="Mugika"))
# {'name': 'Anartz', 'lastname': 'Mugika'}

por_pos_nombre_args = lambda *args , **kwargs : kwargs.get('key', False)
print(por_pos_nombre_args(name="Anartz")) # False
```



FUNCIONES ANÓNIMAS – LAMBDA – DENTRO DE OTRA FUNCIÓN

- ▶ El poder de lambda se muestra mejor cuando lo usamos como función anónima dentro de otra función.
- ▶ Tenemos en el siguiente caso una definición de función que toma un argumento y ese número se multiplica por un número desconocido que asignamos posteriormente.

```
def multiplier(n):  
    return lambda a : a * n  
  
mydoubler = multiplier(2) # lambda a: a * 2  
mytripler = multiplier(3) # lambda a: a * 3  
  
print(mydoubler(11))      # a= 11 => lambda a: a * 2 = 22  
print(mytripler(11))      # a= 11 => lambda a: a * 3 = 33
```