



ProfesorenVideo.com

Parámetros en Python

ARGUMENTOS:

Son los '**elementos**' que se **envían** en la llamada de una función.

instruccion

Instrucción

```
generar_reporte (5, edad, lista)
```

instruccion

PARÁMETROS:

Son las **variables** que '**reciben**' lo enviado, en la definición de una función.

```
def generar_reporte( para1, para2, para3):  
    instruccion  
    instruccion  
    [ return ]
```

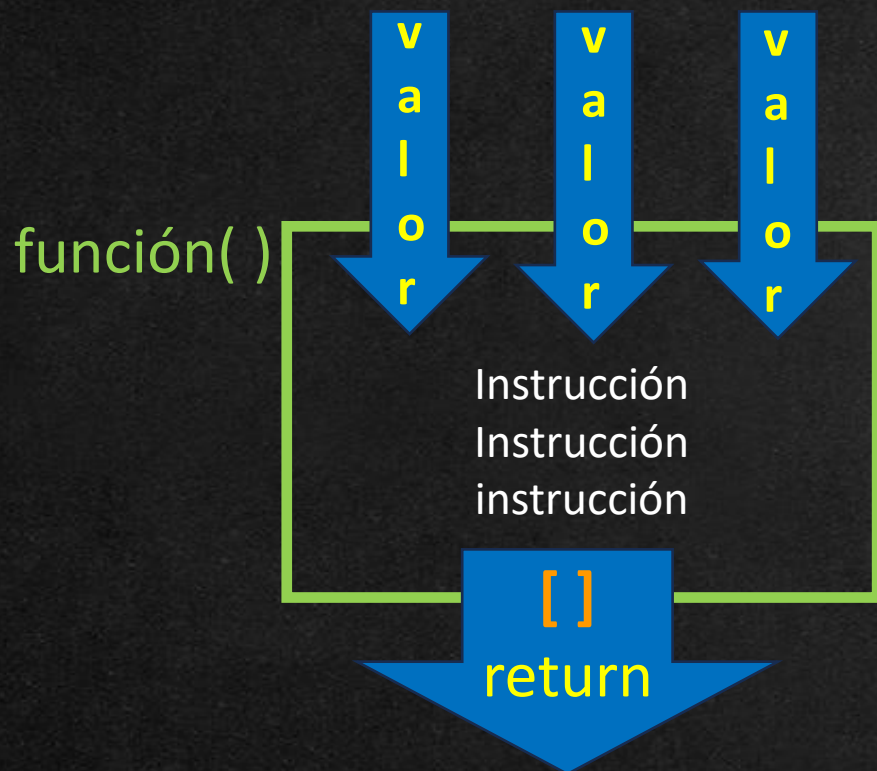
El proceso que implica el uso de ambos (**argumentos** y **parámetros**) se denomina: **PASE DE PARÁMETROS**



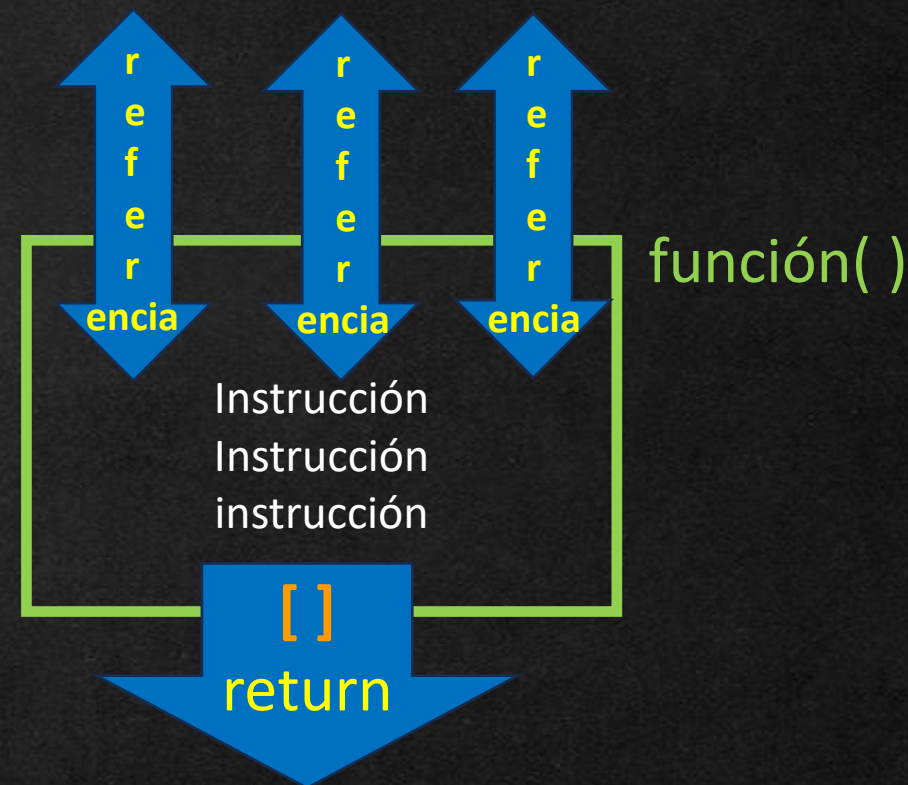
ProfesorenVideo.com

Parámetros en Python

PARÁMETROS POR VALOR



PARÁMETROS POR REFERENCIA

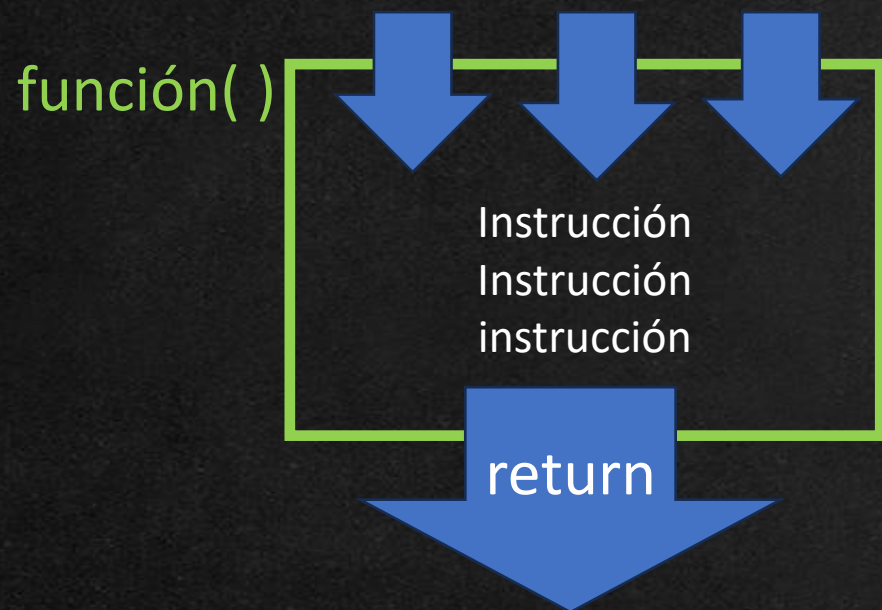




ProfesorenVideo.com

Parámetros en Python

PARÁMETROS POR VALOR



Tipo de datos de los argumentos por VALOR?

- Números enteros (int)
- Números flotantes (float)
- Números complejos (complex)
- Booleanos (bool)
- Cadenas de caracteres (str)
- Tuplas (tuple)
- Rangos (range)
- Objetos congelados (frozenset)
- Objetos personalizados definidos por el usuario (si están diseñados para ser inmutables)

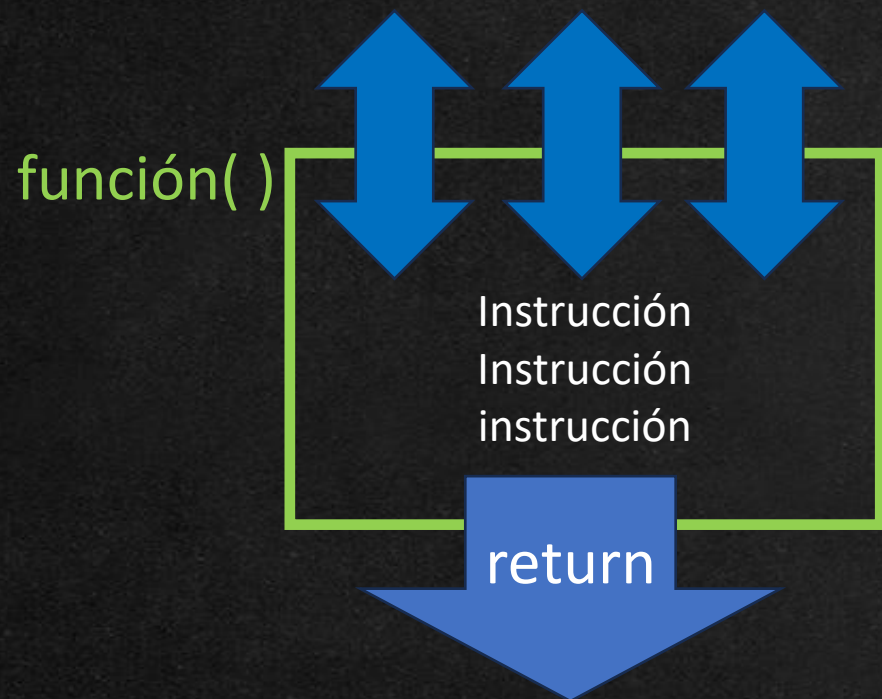
Se les conoce como **INMUTABLES**



ProfesorenVideo.com

Parámetros en Python

PARÁMETROS POR REFERENCIA



Tipo de datos de los argumentos por REFERENCIA?

- Listas (**list**)
- Dicionarios (**dict**)
- Conjuntos (**set**)
- Arreglos Nativos (**array**) y arreglos **NumPy**
- Objetos personalizados definidos por el usuario (si **implementan métodos para modificar su estado interno**)

Se les conoce como **MUTABLES**



Parámetros en Python

Y cuál es la explicación técnica que hace la diferencia entre ambos tipos de pase de parámetros ??

Pase de parámetro por valor: En la llamada se envía un **VALOR** y en la definición de la función se crea **una copia del valor**, por lo tanto existen dos(2) elementos diferentes y al modificar la copia, el original se mantiene intacto.

Pase de parámetro por referencia: En la llamada de la función se envía una **REFERENCIA DE LA VARIABLE** (dirección de memoria donde se ubica), por lo tanto cuando se hace una modificación dentro de la función, ésta se hace realmente en la dirección enviada y con ello afecta al argumento correspondiente.



ProfesorenVideo.com

Parámetros en Python

Y cuál es la explicación técnica que hace la diferencia entre ambos tipos de pase de parámetros ??

cantidad = 1000

lista = [2, 3, 5, 7, 9, 11]

2	3	5	7	9	11
0	1	2	3	4	5

B8CF

→ Dirección de memoria

Ejemplo de llamada:

funcion_ejemplo(5 , cantidad, lista)

copias

5

1000

B8CF



def funcion_ejemplo(para1, para2, para3)

instruccion 5 -> 6

Instrucción 1000 -> 1200

Instrucción B8CF ↔ ?