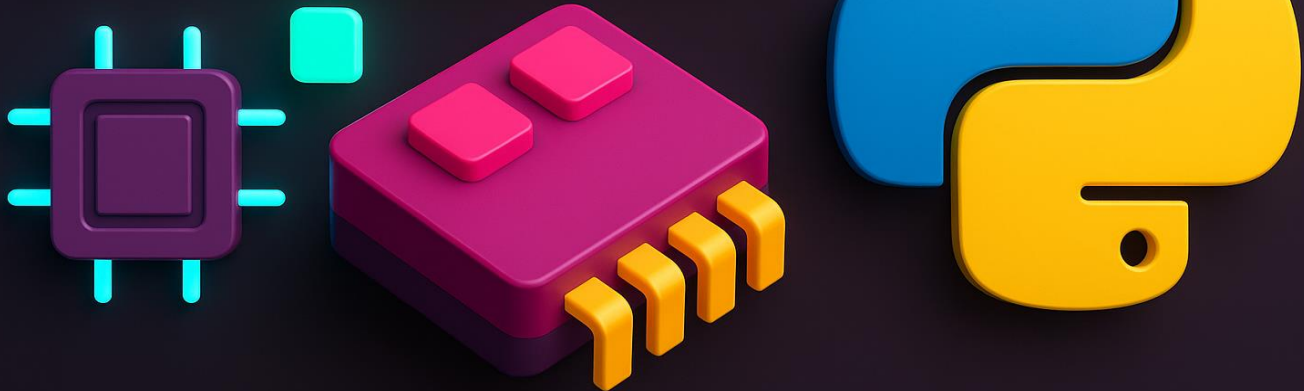


# MANEJO DE MEMORIA



## Recordatorio: Cómo Funciona Realmente el Manejo de Variables en Python

### Introducción

En esta lección vamos a reforzar el concepto real de cómo se gestionan las variables en Python a nivel de memoria. Aunque anteriormente trabajamos con una explicación simplificada, ahora veremos qué ocurre realmente con los objetos, las direcciones de memoria y el recolector de basura. Esta comprensión es clave para dominar el modelo de objetos de Python.

#### ◆ Paso 1: ¿Qué pasa cuando se crea una variable?

✦ Cuando escribimos una línea como:

<https://www.globalmentoring.com.mx>

```
edad = 28
```

🔍 Lo que sucede en memoria no es que `edad` almacene directamente el número 28, sino que:

- Se **crea un objeto** de tipo entero con el valor 28 en una región de memoria conocida como **heap**.
- Se crea la **variable** `edad` que **almacena una referencia** (dirección de memoria) a ese objeto.

Lo mismo ocurre con las siguientes líneas:

```
altura = 1.65  
pais = "México"
```

- `altura` apunta a un objeto de tipo `float`.
- `pais` apunta a un objeto de tipo `str`.

🧠 Cada uno de estos objetos reside en una dirección diferente en memoria.

---

## ◆ Paso 2: ¿Qué ocurre al modificar el valor?

Cuando luego cambiamos el valor de `edad`, por ejemplo:

```
edad = 30
```

✨ Lo que pasa es que:

- Se **crea un nuevo objeto** de tipo entero con el valor 30.
- La variable `edad` **apunta ahora a este nuevo objeto**.
- El objeto anterior (28) **pierde su referencia**, por lo tanto, queda "huérfano".

Lo mismo sucede con:

```
altura = 1.68
```

💡 El objeto 1.65 también se vuelve inaccesible y es eliminado eventualmente por el recolector de basura de Python.

---

## 🗑️ ¿Qué es el recolector de basura?

💻 Python tiene un **recolector de basura automático**, que:

- Detecta los objetos que **ya no tienen ninguna referencia**.
- Los **elimina automáticamente** para liberar memoria.

Tú no tienes que preocuparte por liberar memoria manualmente, ¡Python lo hace por ti! 🐍

---

## 🧩 Resumen Visual (conceptual)

🔗 Cuando se ejecutan estas instrucciones:

```
edad = 28  
edad = 30
```

⬇ Internamente:

- Se crea un objeto 28 → edad apunta a él.
  - Luego se crea un nuevo objeto 30 → edad cambia su referencia.
  - El objeto 28 queda sin uso y será eliminado automáticamente.
- 

## ✅ Conclusión

En esta lección reforzaste cómo funciona **realmente** el manejo de variables en Python:

- Cada valor es un **objeto** en memoria.
- Las variables **apuntan** a esos objetos.
- Cuando cambias el valor, no se reemplaza el contenido del objeto, sino que se **crea uno nuevo**.
- Python se encarga de **limpiar la memoria** mediante su recolector de basura 🧹.

Este entendimiento te ayudará a tener un control más preciso del flujo de datos en tus programas, especialmente cuando empieces a trabajar con objetos más complejos.

---

Sigue adelante con tu aprendizaje 🚀, ¡el esfuerzo vale la pena!

¡Saludos! 🙌

Ing. Marcela Gamiño e Ing. Ubaldo Acosta

Fundadores de [GlobalMentoring.com.mx](https://www.globalmentoring.com.mx)