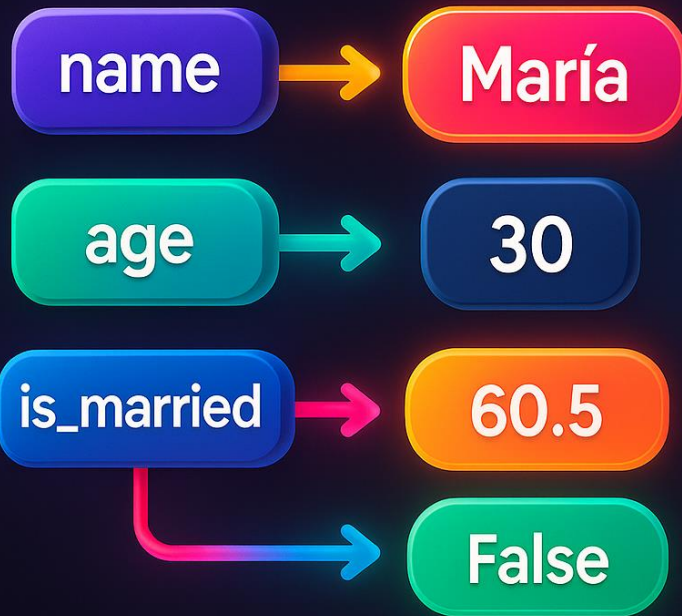


# VARIABLES



## Modificar el Valor de Variables en Python

### Introducción

En esta lección aprenderás cómo modificar el valor de una variable ya existente en Python. Veremos cómo se sobrescriben los valores anteriores en memoria, cómo Python permite cambiar el tipo de dato de una variable durante la ejecución y cómo evitar errores comunes al usar variables no declaradas.

#### ◆ Paso 1: Crear o abrir el archivo del proyecto

Primero asegúrate de estar trabajando en el proyecto previamente creado. Si no lo tienes, crea un nuevo archivo llamado `ModificarVariables.py` dentro de tu proyecto `Variables`.

## ◆ Paso 2: Declarar e imprimir valores iniciales

### Descripción:

Vamos a declarar las variables `edad`, `altura` y `pais`, luego mostraremos sus valores iniciales.

### Archivo y ruta:

`Variables/variables.py`

### Código:

```
edad = 28
altura = 1.65
pais = 'México'

print("Valores iniciales:")
print("Edad:", edad)
print("Altura:", altura)
print("País:", pais)
```

### Explicación:

- Se definen las variables con valores iniciales.
- Se imprimen en consola utilizando `print()`.

---

## ◆ Paso 3: Modificar los valores de las variables

### Descripción:

Vamos a asignar nuevos valores a `edad` y `altura`, y mostrar la diferencia respecto a los valores iniciales.

### Código:

```
edad = 30
altura = 1.68

print("Valores modificados:")
print("Edad:", edad)
print("Altura:", altura)
print("País:", pais)
```

### Explicación:

- Al modificar la variable, se **sobrescribe el valor anterior** en memoria.
  - El valor de `pais` se mantiene igual, ya que no fue modificado.
- 

## ◆ Paso 4: Asignar otro tipo de dato a una variable

### Descripción:

Vamos a reasignar un valor de tipo diferente a la variable `edad`.

### Código:

```
edad = "30"  
print("Edad ahora como cadena:", edad)
```

### Explicación:

- Ahora `edad` contiene una cadena de texto.
  - Python permite cambiar el tipo dinámicamente.
- 

## ◆ Paso 5: Manejo de errores por variables no declaradas

### Descripción:

Veamos qué pasa si intentamos imprimir una variable que no ha sido definida aún.

### Código:

```
# print(telefono) # Esta línea genera un NameError  
  
telefono = 55443322  
print("Teléfono:", telefono)
```

### Explicación:

- Si intentas acceder a una variable antes de declararla, obtendrás un `NameError`.
- Una vez definida, ya puedes acceder sin problemas.



## Código final del archivo trabajado

```
# Variables en Python

# Declaracion e inicializacion de variables
edad = 28
altura = 1.65
pais = 'México'

# Acceder a las variables
print('Valores iniciales: ')
print('Edad:', edad)
print('Altura:', altura)
print('Pais:', pais)

# Modificar el valor de una variable
edad = 30
altura = 1.68

# Acceder a las variables
print('Valores modificados: ')
print('Edad:', edad)
print('Altura:', altura)
print('Pais:', pais)

# En python el tipo es dinamico
edad = 'treinta'
print('Edad:', edad)

# Si queremos acceder a una variable no declarada manda error
telefono = '55443322'
print('Teléfono:', telefono)
```



## Conclusión

Aprendiste cómo modificar el valor de una variable en Python, cómo cambiar su tipo de dato gracias a la naturaleza dinámica del lenguaje, y cómo evitar errores al trabajar con variables no declaradas. Estos fundamentos son esenciales para desarrollar programas interactivos y eficientes 🧠💡

**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)**