




Guía paso a paso: Constantes en Python

Introducción

En esta lección aprenderemos cómo trabajar con **constantes en Python**, entendiendo que aunque el lenguaje no las implementa de forma estricta como en otros lenguajes, existe una convención ampliamente utilizada: definir las constantes en **mayúsculas** y, si tienen varias palabras, separarlas por guiones bajos. A través de esta guía conocerás cómo declarar y utilizar constantes, además de cómo acceder a constantes predefinidas en Python mediante módulos como `math`. 

Paso 1: Crear el archivo del proyecto

 Nombre y ruta del archivo:

`Variables/constantes.py`

👉 **Descripción:** Vamos a crear un archivo Python llamado `02-14-00-Constantes-UP.py` en nuestro proyecto, donde declararemos las constantes y realizaremos las impresiones en pantalla.

✅ **Código:**

```
# Definición de constantes
PI = 3.14159
MENSAJE_ERROR = 'Se ha producido un error'
NOMBRE_USUARIO_VALIDO = 'admin'
NOMBRE_BASE_DATOS = 'Clientes_DB'

# Imprimir constantes definidas
print(f'El valor de PI es: {PI}')
print(f'Mensaje de error: {MENSAJE_ERROR}')
print(f'Nombre usuario válido: {NOMBRE_USUARIO_VALIDO}')
print(f'Nombre de la base de datos: {NOMBRE_BASE_DATOS}')

# Intento de modificación (no recomendado)
NOMBRE_BASE_DATOS = 'Listado_Clientes_DB'
print('No cambiar el valor de una constante.')
print(f'Nuevo valor de la constante (no recomendado): {NOMBRE_BASE_DATOS}')

# Usar la constante pi del módulo math
print(f'Constante PI del módulo math: {math.pi}')
```

📄 **Explicación:**

- Importamos el módulo `math` para utilizar su constante `pi`.
- Definimos nuestras propias constantes (`PI`, `MENSAJE_ERROR`, `NOMBRE_USUARIO_VALIDO`, `NOMBRE_BASE_DATOS`) siguiendo la convención de mayúsculas y guiones bajos.
- Imprimimos cada una de estas constantes.
- Luego modificamos intencionalmente el valor de `NOMBRE_BASE_DATOS` para mostrar que **Python no impide cambiar una constante**, aunque por buenas prácticas **no deberíamos hacerlo**.
- Finalmente mostramos cómo acceder a la constante `pi` directamente del módulo `math`.

12 Paso 2: Ejecutar el código

👉 **Descripción:** Ahora que ya escribimos el código, procedemos a ejecutarlo para observar los resultados.

📄 **Acción:**

- Haz clic derecho sobre el archivo `constantes.py`.

- Selecciona **Run** o **Ejecutar** para ver los resultados en la consola.

🤖 Resultado esperado:

```
*** Constantes en Python ***  
El valor de PI es: 3.14159  
Mensaje de error: Se ha producido un error  
Nombre usuario válido: admin  
Nombre de la base de datos: Clientes_DB  
No cambiar el valor de una constante.  
Nuevo valor de la constante (no recomendado): Listado_Clientes_DB  
Constante PI del módulo math: 3.141592653589793
```

✅ Observa cómo al imprimir las constantes definidas manualmente se respeta su valor inicial, pero también cómo Python permite (aunque no recomienda) modificar una constante. Además, se imprime la constante `pi` directamente del módulo `math`, que a diferencia de la nuestra, tiene una precisión mayor y está definida en minúscula.

🎉 Conclusión

🇲🇽 En esta lección aprendiste que **Python no implementa las constantes de forma estricta**, pero es una **convención y buena práctica** escribirlas en mayúsculas para indicar que su valor no debería cambiar. También conociste cómo acceder a constantes predefinidas del módulo `math`, como `pi`, que viene incorporada en Python. Entender y aplicar estas convenciones ayuda a escribir código más legible y profesional. ¡Sigue practicando para fortalecer tus habilidades en Python! 🚀 🌱

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)