



GETOPT

`getopt` nos permite manejar opciones cortas (`-o`) y largas (`--option`) en los argumentos de un script.

📌 ¿Cómo funciona?

1. Se importa el módulo `getopt` y `sys` .
2. Se define qué opciones espera el script (cortas y largas).
3. Se analiza (`parsea`) los argumentos de la terminal.
4. Se actúa según las opciones recibidas.

Ejemplo Básico:

```
import sys
import getopt

def main():
    try:
        # Definimos opciones
        opts, args = getopt.getopt(sys.argv[1:], "ho:v", ["help", "output="])

        for opt, arg in opts:
            if opt in ("-h", "--help"):
```

```

        print("Uso: script.py -o <archivo_salida> -v")
        sys.exit()
    elif opt in ("-o", "--output"):
        print(f"Guardando salida en: {arg}")
    elif opt == "-v":
        print("Modo verbo activado")

except getopt.GetoptError as err:
    print(f"Error: {err}")
    sys.exit(2)

if __name__ == "__main__":
    main()

```

✓ ¿Para qué sirve `getopt.getopt()` en el código?

- Se usa para analizar (`parsear`) los argumentos de la terminal y separarlos en opciones (`o` , `-output`) y valores asociados.

✗ ¿Cómo se diferencian las opciones cortas (`-o`) de las largas (`--output`)?

- La **opción corta** usa **un guion ()** y suele ser de una sola letra (`o`).
- La **opción larga** usa **dos guiones (-)** y suele ser una palabra completa (`-output`).
- Ambas pueden aceptar valores, pero no hay diferencia en la cantidad de palabras.

✓ ¿Qué hace el script cuando se usa `-h` o `--help` ?

- Muestra un mensaje de ayuda (`"Uso: script.py -o <archivo_salida> -v"`) y luego se cierra con `sys.exit()` .