## Abandonar cambios

Veamos ahora cómo abandonar un cambio y volver al estado de trabajo anterior. Primero, añade una nueva línea a *hello\_git.py*:

**hello\_git.py**

print("Hello Git world!")  
print("Hello everyone.")  
  
print("Oh no, I broke the project!")

Guarda y ejecuta este archivo.

Comprobamos el estado y vemos que Git se da cuenta de este cambio:

git\_practice$ git status  
On branch main  
Changes not staged for commit:  
 (use "git add <file>..." to update what will be committed)  
 (use "git restore <file>..." to discard changes in working directory)  
  
❶ modified: hello\_git.py  
  
no changes added to commit (use "git add" and/or "git commit -a")  
git\_practice$

Git ve que hemos modificado *hello\_git.py* ❶, y podemos confirmar el cambio si queremos. Pero esta vez, en lugar de confirmar el cambio, volveremos a la última confirmación, cuando sabíamos que nuestro proyecto funcionaba. No haremos nada en *hello\_git.py*: no borraremos la línea ni utilizaremos la función Deshacer del editor de texto. En su lugar, introduce los siguientes comandos en tu sesión de terminal:

git\_practice$ git restore .  
git\_practice$ git status  
On branch main  
nothing to commit, working tree clean  
git\_practice$

El comando git restore nombrearchivo te permite abandonar todos los cambios desde la última confirmación en un archivo específico. El comando git restore . abandona todos los cambios realizados en todos los archivos desde la última confirmación; esta acción restaura el proyecto al último estado confirmado.

Cuando vuelvas a tu editor de texto, verás que *hello\_git.py* ha vuelto a este estado:

print("Hello Git world!")  
print("Hello everyone.")

Aunque volver a un estado anterior pueda parecer trivial en este sencillo proyecto, si estuviéramos trabajando en un proyecto grande con docenas de archivos modificados, se restaurarían todos los archivos que hubieran cambiado desde el último commit. Esta función es increíblemente útil: puedes hacer tantos cambios como quieras al implementar una nueva función, y si no funcionan, puedes descartarlos sin afectar al proyecto. No tienes que recordar esos cambios y deshacerlos manualmente. Git hace todo eso por ti.

[anterior](b04_12.html)[Subtema 13 de 16: (Ver todo)](b04.html)[siguiente](b04_14.html)