



ximenacq@ximena-vm: ~/LaboratorioGit

GNU nano 4.8

gatosGit/gatos.py

```
import funcionesTarea

gato1 = funcionesTarea.gato()
<<<<<<< HEAD
gato1.nombre = "Lia"
=====
gato1.nombre = "Campanita"
>>>>>>> feature
gato1.color = "negro"
gato1.edad = 4
gato1.aniadirlista()

gato2 = funcionesTarea.gato()
gato2.nombre = "Pelusa"
gato2.color = "cafe"
gato2.edad = 3
gato2.aniadirlista()

gato3 = funcionesTarea.gato()
gato3.nombre = "Jimmy"
gato3.color = "gris"
gato3.edad = 1
gato3.aniadirlista()

funcionesTarea.printlista()
```

```
ximenacq@ximena-vm:~/LaboratorioGit$ git log --oneline --graph
* 96d1ada (HEAD -> main, origin/main, origin/HEAD) Cambio de nombre gato1 branch main
* d57fc66 (origin/feature, feature) Creacion de objeto gato3
* b7649bd Cambio de nombre del objeto gato1
* 1640128 included gato2 and list printing
* c2b877a corrected mistakes
* d02a6d2 included gato1
* 7ab0adc included functions file
* a13fd08 first commit
```

**--graph**  
Draw a text-based graphical representation of the commit history on the left hand side of the output. This may cause extra lines to be printed in between commits, in order for the graph history to be drawn properly. Cannot be combined with **--no-walk**.  
  
This enables parent rewriting, see [History Simplification](#) above.  
  
This implies the **--topo-order** option by default, but the **--date-order** option may also be specified.

`--oneline`

This is a shorthand for "`--pretty=oneline --abbrev-commit`" used together.

- oneline

`<hash> <title line>`

This is designed to be as compact as possible.

```
ximenacq@ximena-vm:~/LaboratorioGit$ python3 gatosGit/gatos.py  
Lia, 4 annos, color: negro  
Pelusa, 3 annos, color: cafe  
Jimmy, 1 annos, color: gris
```