

RAPPORT

LO43

XIA Xiaotong

1. git config

Git contient un outil appelé **git config** pour vous permettre de voir et modifier les variables de configuration qui contrôlent tous les aspects de l'apparence et du comportement de Git.

```
[xiaxiaotongdeMacBook-Pro:~ xiaxiaotong$ git config
usage: git config [<options>]

Config file location
  --global          use global config file
  --system          use system config file
  --local           use repository config file
  -f, --file <file> use given config file
  --blob <blob-id>  read config from given blob object

Action
  --get             get value: name [value-regex]
  --get-all        get all values: key [value-regex]
  --get-regexp      get values for regexp: name-regex [value-regex]
  --get-urlmatch    get value specific for the URL: section[.var] URL
  --replace-all    replace all matching variables: name value [value_regex]
  --add             add a new variable: name value
  --unset           remove a variable: name [value-regex]
  --unset-all      remove all matches: name [value-regex]
  --rename-section  rename section: old-name new-name
  --remove-section  remove a section: name
  -l, --list        list all
  -e, --edit        open an editor
  --get-color       find the color configured: slot [default]
  --get-colorbool   find the color setting: slot [stdout-is-tty]

Type
  --bool            value is "true" or "false"
  --int             value is decimal number
  --bool-or-int     value is --bool or --int
  --path            value is a path (file or directory name)

Other
  -z, --null        terminate values with NUL byte
  --name-only       show variable names only
  --includes        respect include directives on lookup
  --show-origin     show origin of config (file, standard input, blob, command line)
```

(la liste de options de git config)

2. git init

Crée un dépôt Git vide ou réinitialise un dépôt existant

```
[xiaxiaotongdeMacBook-Pro:~ xiaxiaotong$ git init
Initialized empty Git repository in /Users/xiaxiaotong/.git/
```

3 . git status

Montre les chemins qui ont des différences entre le fichier d'index et le commit HEAD actuel, les chemins qui ont des différences entre l'arbre de travail et de fichier d'index, et les chemins dans l'arbre de travail qui ne sont pas suivis par Git.

```
[xiaxiaotongdeMacBook-Pro:~ xiaxiaotong$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
```

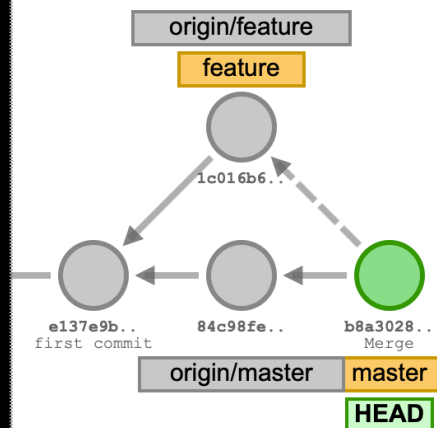
```
.CFUserTextEncoding
.DS_Store
.IdentityService/
.ServiceHub/
.Trash/
.bash_history
.bash_sessions/
.config/
.mono/
.packettracer
.runoob.txt.swp
.subversion/
.templateengine/
.vim/
.viminfo
Applications/
Cisco Packet Tracer 7.2.2/
Desktop/
Documents/
Downloads/
Library/
Movies/
Music/
PacketTracer7/
Pictures/
Public/
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

4. git merge

Intègre les modifications des commits nommés (à partir du moment où leurs historiques ont divergé de la branche actuelle) dans la branche actuelle. Cette commande est utilisée par git pull pour incorporer les modifications d'un autre référentiel et peut être utilisée manuellement pour fusionner les modifications d'une branche à une autre.

```
$ git checkout master  
$ git merge feature
```



5. git diff

Affiche les modifications entre les commits, un commit et l'arbre de travail, etc.

```
[xiaxiaotongdeMacBook-Pro:~ xiaxiaotong$ echo "HELLO WORLD" > hello.txt ]
[xiaxiaotongdeMacBook-Pro:~ xiaxiaotong$ git add hello.txt ]
[xiaxiaotongdeMacBook-Pro:~ xiaxiaotong$ echo "HELLO GIT" > hello.txt ]
[xiaxiaotongdeMacBook-Pro:~ xiaxiaotong$ git diff ]
diff --git a/hello.txt b/hello.txt
index 4e3dffe..7b7691f 100644
--- a/hello.txt
+++ b/hello.txt
@@ -1,1 @@
-HELLO WORLD
+HELLO GIT
```

6. git blame

Annote chaque ligne du fichier donné avec les informations de la révision qui a modifié la dernière ligne. Vous pouvez éventuellement commencer à annoter à partir de la révision donnée.

Lorsque spécifié une ou plusieurs fois, -l limite les annotations aux lignes demandées.

L'origine des lignes est automatiquement suivie lors du renommage de tout le fichier (il n'existe actuellement aucune option pour désactiver le renommage après). Suivre des lignes déplacées d'un fichier à un autre ou suivre des lignes copiées et collées à partir d'un autre fichier, etc.

```
[xiaxiaotongdeMacBook-Pro:~ xiaxiaotong$ git blame
usage: git blame [<options>] [<rev-opts>] [<rev>] [--] <file>
```

<rev-opts> are documented in git-rev-list(1)

--incremental	Show blame entries as we find them, incrementally
-b	Show blank SHA-1 for boundary commits (Default: off)
--root	Do not treat root commits as boundaries (Default: off)
--show-stats	Show work cost statistics
--progress	Force progress reporting
--score-debug	Show output score for blame entries
-f, --show-name	Show original filename (Default: auto)
-n, --show-number	Show original linenumber (Default: off)
-p, --porcelain	Show in a format designed for machine consumption
--line-porcelain	Show porcelain format with per-line commit information
-c	Use the same output mode as git-annotate (Default: off)
-t	Show raw timestamp (Default: off)
-l	Show long commit SHA1 (Default: off)
-s	Suppress author name and timestamp (Default: off)
-e, --show-email	Show author email instead of name (Default: off)
-w	Ignore whitespace differences
--indent-heuristic	Use an experimental heuristic to improve diffs
--minimal	Spend extra cycles to find better match
-S <file>	Use revisions from <file> instead of calling git-rev-list
--contents <file>	Use <file>'s contents as the final image
-C[<score>]	Find line copies within and across files
-M[<score>]	Find line movements within and across files
-L <n,m>	Process only line range n,m, counting from 1
--abbrev[=<n>]	use <n> digits to display SHA-1s