

A Git™ Cheat Sheet for Agile Writers

BASIC COMMANDS

Use SSH to access a location:

```
ssh <username>@<host>
```

Navigate to a directory:

```
cd <directory>
```

Open a file in Vim:

```
vi <file>
```

List current directory contents:

```
ls
```

GETTING GIT DATA

Clone a New Repository:

```
git clone <URL>
```

Check Out a Branch:

```
git checkout <branch>
```

Search Within The Branch:

```
git grep "search string"
```

Get the list of repo branches:

```
git fetch <URL>
```

Get recent branch changes:

```
git pull
```

MANAGE YOUR BRANCH

Check Branch Status:

```
git status
```

Undo Uncommitted Changes:

```
git reset HEAD
```

New Branch:

```
git branch
```

Delete a Branch:

```
git branch -D <branch>
```

MAKING COMMITS

Add Files to a Commit:

```
git add <file>
```

Package Files to Commit:

```
git commit
```

Push the Commit:

```
git push <URL> <branch>
```

WHO DID THAT?

Line-by-line commit info:

```
git blame <file>
```

COMMIT LOG ENTRIES

All commit log entries:

```
git log
```

Git Log Options:

```
git log [option]
```

Only # entries: `-#`

One commit: `<SHA>`

One file: `-p <file>`

Before a date: `--before="<date>"`

After a date: `--after="<date>"`

Short list of log entry 1st lines:

```
git shortlog
```

WHEN IN DOUBT...

Git's CLI Help Files:

```
git <command> --help
```

Git's Official Website:

```
http://www.git-scm.com
```

Brackets ([]) indicate where you should add options, etc. to your commands.

Carats (<>) show items to replace with your specific information.

File-related commands and options can use relative or absolute file paths.

