Contents

Git **1**

- Setup 1
- Creating a Repository 1
- Tracking Changes 1
- Exploring History 2
- Ignoring things 2
- Remote repositories 3

Git

See also:

- https://swcarpentry.github.io/git-novice
- http://rogerdudler.github.io/git-guide/

Setup

Set global configuration

```
git config --global user.name "Vlad Dracula"
git config --global user.email "vlad@tran.sylvan.ia"
git config --global color.ui "auto"
```

List all configured values

```
git config --list
```

Creating a Repository

First we create a new directory, then we initialize it as a repository

```
mkdir planets
cd planets
git init
```

Verify that it is indeed a repository

```
git status
```

Tracking Changes

Create a file called `mars.txt`

```
nano mars.txt
```

and enter the following text

```
Cold and dry, but everything is my favorite color
```

Verify that git noticed some changes with

```
git status
```

Add the changes to the staging area

```
git add mars.txt
```

Again check with

```
git status
```

Now commit the changes to the repository and provide a message

```
git commit -m "Start notes on Mars as a base"
```

Re-check `git status`. Have a look in the log

```
git log
```

We will now introduce some changes. Open the file again with `nano mars.txt` and add a second line

```
The two moons may be a problem for Wolfman
```

Use `git status` again. To see the introduced changes use

```
git diff
```

Add and commit the new content

```
git add mars.txt
git commit -m "Add concerns about effects of Mars' moons on Wolfman"
```

Repeat the above steps as often as you want. You can check your commits with `git log` at any time.

Exploring History

Introduce some changes with `nano mars.txt` and do not add or commit them. Explore your changes against the current and previous versions

```
git diff HEAD mars.txt
git diff HEAD~1 mars.txt
git diff HEAD~2 mars.txt
```

Explore all changes in comparison to a specific version, first find the hash sum for your desired commit using `git log` and then

```
git diff <hash-sum> mars.txt
# for example (this will probably not work for you)
git diff 373e6b6e9d86dab310b95642660e7b4c07054c1e mars.txt
# this can be shortened as long as the beginning of the hash sum is unique
git diff 373e6b6 mars.txt
```

Revert the changes you have not yet added or committed

```
git checkout HEAD mars.txt
# or just
git checkout -- mars.txt
```

Ignoring things

Create some files you don't want to track

```
mkdir results
touch a.dat b.dat c.dat results/a.out results/b.out
```

Check `git status` Ignore the files by creating a special file `nano .gitignore` with the content

```
*.dat
results/
```

Check again with `git status`. Add `.gitignore` and commit your changes.

Remote repositories

To work with remote repositories, e.g. on GitHub important commands are

```
# Clone a repository to your local machine
git clone https://github.com/user/repo.git
# Get changes from the server
git pull
# Publish your local changes on the server
git push
```