

Exercise 1: Basic Local Git Cycle

Overview

In this exercise we will go through one basic local Git version control cycle. You will make use of some of the most frequently used Git commands:

```
git status
git init
git add <file_with_new_changes>
git commit -m 'my commit message'
git log / git reflog
git diff <name_of_changed_file>
```

Step-by-Step instructions

We start in our workshop directory. Let's make a new folder just for this exercise.

```
>> cd ~/git_workshop2020
>> mkdir my_first_repo
>> cd my_first_repo
```

At this point the new folder is not under Git version control yet. You can see this by the output of

```
>> git status
fatal: Not a git repository (or any of the parent directories): .git
```

Once you initiate Git version control for this folder, it will become a repository or "repo" and a new file `.git` will be created. Next, check the status of your repo. It should be empty like the one below.

```
>> git init
>> git status
On branch main

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

Can you see the hidden `.git` file when typing the command `ls -al` ?

Let's create our first file now! Open your favourite text editor. Write a short sentence about yourself (can be something silly like your favourite food). Then save the file in your local repo under the name `name_surname.md` (.md for [Markdown](#)). If you now check the status again, you will see that the file appears as an untracked file.

```
>> git status
On branch main

No commits yet

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

    John_Doe.md
```

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

As you can see in the output, Git already suggests the next useful command `git add`. Let's add the file and check its status again!

```
>> git add John_Doe.md
>> git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   John_Doe.md
```

Congratulations, Git is now tracking your file and will note any new changes automatically.

Questions

Go ahead and make a change in your file.

How does the output of `git status` change?

Before you run `git add` on your modified file, try `git diff`. What does it do?

Now that you have added new files to your folder and made some changes you want to create a snapshot in Git, so that you can always return to this specific point in your project progress. Such a snapshot is called a 'commit' in Git language. Make sure all changes you want to include in this commit have been added with `git add`. With every commit you should also define a commit message that describes what changes this commit includes. For this you can use the `-m` flag.

```
>> git commit -m 'adding a new file'
[main (root-commit) 9770d9c] adding a new file
1 file changed, 1 insertion(+)
create mode 100644 John_Doe.md
```

A new commit has now been added to your Git repo. You can check the history of your commits with the command `git log` or `git reflog`. Try both commands and see how they compare!