

1. Create a new folder named `data\_project` on your local machine. Initialise it as a git repository and verify its status.

```
mkdir data_project
```

```
cd data_project
```

```
git init
```

```
git status
```

```
● (base) cindy@wangxinyideMacBook-Pro data_project % mkdir data_project
cd data_project
git init
git status

hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /Users/cindy/qt350-quiz01/data_project/data_project/data_project/.git/
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
○ (base) cindy@wangxinyideMacBook-Pro data_project %
```

2. Inside the `data\_project` directory, create three Python files: `data\_cleaning.py`, `data\_analysis.py`, and `visualisation.py`. Add all these files to the staging area.

```
touch data_cleaning.py data_analysis.py visualisation.py
```

```
git add data_cleaning.py data_analysis.py visualisation.py
```

```
git status
```

```
nothing to commit (create/copy files and use "git add" to track)
● (base) cindy@wangxinyideMacBook-Pro data_project % touch data_cleaning.py data_analysis.py visualisation.py
git add data_cleaning.py data_analysis.py visualisation.py
git status

On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   data_analysis.py
    new file:   data_cleaning.py
    new file:   visualisation.py
❖ (base) cindy@wangxinyideMacBook-Pro data_project %
```

3. Commit the added Python files with the message "Initial commit with data processing scripts".

`git commit -m "Initial commit with data processing scripts"`

```
● (base) cindy@wangxinyideMacBook-Pro data_project % git commit -m "Initial commit with data processing scripts"

[master (root-commit) 85b9a5d] Initial commit with data processing scripts
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 data_analysis.py
create mode 100644 data_cleaning.py
create mode 100644 visualisation.py
○ (base) cindy@wangxinyideMacBook-Pro data_project %
```

4. Display the commit history of your repository.

`git log -- oneline`

```
● (base) cindy@wangxinyideMacBook-Pro data_project % git log --oneline

85b9a5d (HEAD -> master) Initial commit with data processing scripts
○ (base) cindy@wangxinyideMacBook-Pro data_project %
```

5. Add a new function `clean_data()` in `data_cleaning.py`. After modifying the file, stage and commit the changes with the message "Add clean\_data function to data\_cleaning.py".

```
● (base) cindy@wangxinyideMBP data_project % echo "def clean_data():\n    pass" >> data_cleaning.py
git add data_cleaning.py
git commit -m "Add clean_data function to data_cleaning.py"

[master eb04750] Add clean_data function to data_cleaning.py
1 file changed, 2 insertions(+)
○ (base) cindy@wangxinyideMBP data_project %
```

6. Create a new branch named `feature-exploration` and switch to it.

`git branch feature-exploration`

`git checkout feature-exploration`

```
● (base) cindy@wangxinyideMBP data_project % git branch feature-exploration
git checkout feature-exploration

Switched to branch 'feature-exploration'
● (base) cindy@wangxinyideMBP data_project % git branch feature-exploration
git checkout feature-exploration

fatal: a branch named 'feature-exploration' already exists
Already on 'feature-exploration'
○ (base) cindy@wangxinyideMBP data_project %
```

7. On the `feature-exploration` branch, add a new file `exploratory\_analysis.py` and commit it with the message "Add exploratory\_analysis.py for initial data exploration"

```
touch exploratory_analysis.py
```

```
git add exploratory_analysis.py
```

```
git commit -m "Add exploratory_analysis.py for initial data exploration"
```

```
● (base) cindy@wangxinyideMBP data_project % touch exploratory_analysis.py
● (base) cindy@wangxinyideMBP data_project % git add exploratory_analysis.py
● (base) cindy@wangxinyideMBP data_project % git commit -m "Add exploratory_analysis.py for initial data exploration"
[feature-exploration f177ab3] Add exploratory_analysis.py for initial data exploration
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 exploratory_analysis.py
○ (base) cindy@wangxinyideMBP data_project %
```

8. Switch back to the `main` (or `master`) branch from `feature-exploration`.

```
git checkout main
```

```
● (base) cindy@wangxinyideMBP data_project % git branch
    feature-exploration
* main
● (base) cindy@wangxinyideMBP data_project % git checkout main
Already on 'main'
○ (base) cindy@wangxinyideMBP data_project %
```

9. Merge the `feature-exploration` branch into the `main` (or `master`) branch.

```
git merge feature-exploration
```

```
● (base) cindy@wangxinyideMBP data_project % git merge feature-exploration
Updating eb04750..f177ab3
Fast-forward
 exploratory_analysis.py | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 exploratory_analysis.py
○ (base) cindy@wangxinyideMBP data_project %
```

10. Commit the merge and list all the branches in your repository.

```
● (base) cindy@wangxinyideMBP data_project % git branch  
\  
> git branch  
  
    feature-exploration  
* main  
    feature-exploration  
* main  
○ (base) cindy@wangxinyideMBP data_project %
```