

Git_Labs

أكاديمية طويق
Tuwaiq Academy



By: Nourah Almutlg

Sarraa Almudayfir

Git-Lab1

Local[Add, Commit, Log]

في هذا التمرين سنقوم باستخدام Git للتعامل مع الملفات وإدارتها بشكل محلي بحيث سنقوم بإنشاء ملفات نصية وتتبعها واحد تلو الآخر كالتالي.

الخطوات

- قم بفتح terminal أو git bash على جهازك.
- قم بالانتقال إلى مجلد Desktop.
- بداخل مجلد Desktop قم بإنشاء مجلد جديد بالاسم git-test.
- قم بالدخول على مجلد git-test من ثم إنشاء مجلد جديد بالاسم test01.
- قم بالدخول على مجلد test01 من خلال terminal أو git bash.
- الآن قم بإنشاء repository جديد باستخدام الأمر git init.
- بهذا الشكل أصبح git قادر على تتبع المشروع، بالتالي قم بإنشاء ملف first.txt من ثم حفظه على repository.

```
test01 — zsh — 102x39
(base) sarraa@sras-MacBook-Air Desktop % mkdir git-test
(base) sarraa@sras-MacBook-Air Desktop % cd git-test
(base) sarraa@sras-MacBook-Air git-test % mkdir test01
(base) sarraa@sras-MacBook-Air git-test % cd test01
(base) sarraa@sras-MacBook-Air test01 % git init
Initialized empty Git repository in /Users/sarraa/Desktop/git-test/test01/.git/
(base) sarraa@sras-MacBook-Air test01 % touch first.txt
(base) sarraa@sras-MacBook-Air test01 % git status
On branch main

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
(base) sarraa@sras-MacBook-Air test01 % git add .
(base) sarraa@sras-MacBook-Air test01 % git status
On branch main

No commits yet

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

(base) sarraa@sras-MacBook-Air test01 % git commit -m "first commit"
[main (root-commit) 6e12bcb] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 first.txt
(base) sarraa@sras-MacBook-Air test01 % git status
On branch main
nothing to commit, working tree clean
(base) sarraa@sras-MacBook-Air test01 %
```

- بعد عمل commit قم بإنشاء ملف آخر بالاسم second.txt من ثم حفظة أيضاً على repository.
 - بعد الانتهاء من حفظ ملف first.txt و second.txt قم بإنشاء ملفين آخرين بالاسم third.txt و fourth.txt.
-

- قم بحفظهم على repository من خلال استخدام الأمر . add وعمل commit لهم.
- الآن قم باستعراض جميع commits التي قمت بعملها من خلال الأمر git log.

```
test01 — -zsh — 92x41

nothing added to commit but untracked files present (use "git add" to track)
[(base) sarraa@sras-MacBook-Air test01 % git add .
[(base) sarraa@sras-MacBook-Air test01 % git commit -m "Sec commit"
[main 608b335] Sec commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 second.txt
[(base) sarraa@sras-MacBook-Air test01 % touch third.txt
[(base) sarraa@sras-MacBook-Air test01 % touch fourth.txt
[(base) sarraa@sras-MacBook-Air test01 % git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        fourth.txt
        third.txt

nothing added to commit but untracked files present (use "git add" to track)
[(base) sarraa@sras-MacBook-Air test01 % git add .
[(base) sarraa@sras-MacBook-Air test01 % git commit -m "third commit"
[main 3655e89] third commit
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 fourth.txt
 create mode 100644 third.txt
[(base) sarraa@sras-MacBook-Air test01 % git log
commit 3655e89cb6e579da0e7dff4d7dfed4f720bf0a9f (HEAD -> main)
Author: Sarraakhalil <kw6.sasm@gmail.com>
Date: Tue Jul 23 17:24:45 2024 +0300

    third commit

commit 608b335753b34e18ff92785de4f9a1bb827af059
Author: Sarraakhalil <kw6.sasm@gmail.com>
Date: Tue Jul 23 17:23:55 2024 +0300

    Sec commit

commit 6e12bcbdbbbb91a9aa31d5f2932f1ab68a79e1f54
Author: Sarraakhalil <kw6.sasm@gmail.com>
Date: Tue Jul 23 17:21:45 2024 +0300

    first commit
(base) sarraa@sras-MacBook-Air test01 %
```

Git-Lab2

Local[Branches and Merging]

من خلال هذا التمرين سنقوم باستخدام مفهوم Branches و عملية دمجهم.

الخطوات

- قم بفتح terminal او git bash الخاص بجهازك من ثم الدخول على مجلد test01 الذي قمنا بإنشاؤه في التمرين السابق.
- قم باستعراض جميع Branches المتواجدة في المشروع.
- الآن قم بإنشاء فرع جديد بالاسم new-feature من ثم الانتقال اليه.
- في فرع new-feature قم بإنشاء ملف بالاسم lab02.txt من ثم عمل add و commit لهذا الملف.
- الآن قم باستعراض الملفات خلال وجودك في الفرع new-feature.

```

test01 — zsh — 92x43
[(base) sarraa@sras-MacBook-Air git-test % cd test01
[(base) sarraa@sras-MacBook-Air test01 % git branch
* main
[(base) sarraa@sras-MacBook-Air test01 % git branch <new-feature>
zsh: parse error near `\\n'
[(base) sarraa@sras-MacBook-Air test01 % git branch new-feature
[(base) sarraa@sras-MacBook-Air test01 % git branch
* main
  new-feature
[(base) sarraa@sras-MacBook-Air test01 % git checkout new-feature
Switched to branch 'new-feature'
[(base) sarraa@sras-MacBook-Air test01 % git branch
  main
* new-feature
[(base) sarraa@sras-MacBook-Air test01 % touch lab02.txt
[(base) sarraa@sras-MacBook-Air test01 % git add .
[(base) sarraa@sras-MacBook-Air test01 % git status
On branch new-feature
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   lab02.txt

[(base) sarraa@sras-MacBook-Air test01 % git command -m "commit"
git: 'command' is not a git command. See 'git --help'.
[(base) sarraa@sras-MacBook-Air test01 % git commit -m "commit"
[new-feature 0912862] commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 lab02.txt
[(base) sarraa@sras-MacBook-Air test01 % ls
first.txt      fourth.txt     lab02.txt      second.txt     third.txt
[(base) sarraa@sras-MacBook-Air test01 % git checkour main
git: 'checkour' is not a git command. See 'git --help'.

The most similar command is
  checkout
[(base) sarraa@sras-MacBook-Air test01 % git checkout main
Switched to branch 'main'
[(base) sarraa@sras-MacBook-Air test01 % git branch
* main
  new-feature
[(base) sarraa@sras-MacBook-Air test01 % ls
first.txt      fourth.txt     second.txt     third.txt
[(base) sarraa@sras-MacBook-Air test01 % 

```

- قم بالعودة الى الفرع master او main من ثم استعراض الملفات مره أخرى. ستلاحظ عدم وجود lab02.txt في هذا الفرع، بالتالي سنقوم بعملية دمج للفرعين حتى يكون الملف متواجد في الفرع master او main.
- قم بدمج الفروع.
- الآن قم باستعراض الملفات مره أخرى، ستجد أن ملف lab02.txt متواجد في الفرع master او main.

```

((base) sarraa@sras-MacBook-Air test01 % git branch
* main
  new-feature
((base) sarraa@sras-MacBook-Air test01 % ls
first.txt      fourth.txt      second.txt      third.txt
((base) sarraa@sras-MacBook-Air test01 % git merge new-feature
Updating 3655e89..0912862
Fast-forward
 lab02.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 lab02.txt
((base) sarraa@sras-MacBook-Air test01 % git branch
* main
  new-feature
((base) sarraa@sras-MacBook-Air test01 % ls
first.txt      fourth.txt      lab02.txt      second.txt      third.txt
((base) sarraa@sras-MacBook-Air test01 % █

```


Git-Lab3

GitHub Remotely

في هذا التمرين سنقوم برفع المجلد test01 الذي قمنا باستخدامه في التمارين السابقة على Github.

الخطوات

- قم بالذهاب الى مسار مجلد test01.
- قم باستعراض remote المتواجدة فيه.
- نلاحظ عدم وجود remote سابق والسبب هو اننا كنا نعمل بشكل محلي لكن الآن سنقوم برفع المشروع والعمل عليه remotely.
- قم بالذهاب الى Github وانشاء repository جديد من ثم نسخ الرابط الخاص به.

-
- قم بالعودة الى terminal او git bash من ثم اضافة remote جديد على الرابط الذي تم نسخه.
 - الآن قم برفع جميع الملفات والتعديلات الى repository المتواجد في Github من خلال استخدام الأمر git push.
 - قم بالانتقال الى صفحة Github من ثم اعادة تحميلها refresh وستظهر لك جميع الملفات التي قمت بانشائها في التمارين السابقة.
 - في حالة أردت، قم بانشاء ملف جديد محلياً من ثم عمل git push مرة أخرى لتلاحظ التحديثات الحاصلة بشكل أكبر.

```

[(base) sarraa@sras-MacBook-Air test01 % git remote
[(base) sarraa@sras-MacBook-Air test01 % git remote -v
[(base) sarraa@sras-MacBook-Air test01 % git remote add origin https://github.com/Sarraakhalil/DS_Test.git
[(base) sarraa@sras-MacBook-Air test01 % git remote
origin
[(base) sarraa@sras-MacBook-Air test01 % git push
fatal: The current branch main has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin main

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.

[(base) sarraa@sras-MacBook-Air test01 % git push push origin main
error: src refspec origin does not match any
error: failed to push some refs to 'push'
[(base) sarraa@sras-MacBook-Air test01 % git push origin name
error: src refspec name does not match any
error: failed to push some refs to 'https://github.com/Sarraakhalil/DS_Test.git'
[(base) sarraa@sras-MacBook-Air test01 % git push -u origin main
To https://github.com/Sarraakhalil/DS_Test.git
! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/Sarraakhalil/DS_Test.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
[(base) sarraa@sras-MacBook-Air test01 % git branch
* main
  new-feature
(base) sarraa@sras-MacBook-Air test01 % 

```