



By: Nourah Almutlg

Solved By: Salman Gassem

Git-Lab1

Local[Add, Commit, Log]

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

الخطوات

- قم بفتح terminal او git bash على جهازك. قم بالانتقال الى مجلد Desktop.
- بداخل مجلد Desktop قم بإنشاء مجلد جديد بالاسم git-test.
- قم بالدخول على مجلد git-test من ثم انشاء مجلد جديد بالاسم test01. قم بالدخول على مجلد test01 من خلال terminal او git bash.
- الآن قم بإنشاء repository جديد باستخدام الأمر git init.
- بهذا الشكل اصبح git قادر على تتبع المشروع، بالتالي قم بإنشاء ملف first.txt من ثم حفظه على repository. بعد عمل commit قم بإنشاء ملف آخر بالاسم second.txt من ثم حفظه 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.

Solution:

```

(base) salman@Salmans-MacBook-Air ~ % cd Desktop
(base) salman@Salmans-MacBook-Air Desktop % mkdir git-test
(base) salman@Salmans-MacBook-Air Desktop % cd git-test
(base) salman@Salmans-MacBook-Air git-test % mkdir test01
(base) salman@Salmans-MacBook-Air git-test % cd test01
(base) salman@Salmans-MacBook-Air test01 % git init
Initialized empty Git repository in /Users/salman/Desktop/git-test/test01/.git/
(base) salman@Salmans-MacBook-Air test01 % touch first.txt
(base) salman@Salmans-MacBook-Air test01 % ls
first.txt
(base) salman@Salmans-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) salman@Salmans-MacBook-Air test01 % git add .
(base) salman@Salmans-MacBook-Air test01 % git commit -m "First Git lab commit"
[main (root-commit) d908ec4] First Git lab commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 first.txt
(base) salman@Salmans-MacBook-Air test01 % ls
first.txt
(base) salman@Salmans-MacBook-Air test01 % touch second.txt
(base) salman@Salmans-MacBook-Air test01 % git status
On branch main

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

nothing added to commit but untracked files present (use "git add" to track)
(base) salman@Salmans-MacBook-Air test01 % git add .
(base) salman@Salmans-MacBook-Air test01 % git commit -m "Second Git lab commit"
[main 20dde9] Second Git lab commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 second.txt
(base) salman@Salmans-MacBook-Air test01 % git status
On branch main
nothing to commit, working tree clean
(base) salman@Salmans-MacBook-Air test01 % ls
first.txt      second.txt
(base) salman@Salmans-MacBook-Air test01 % git log
commit 20dde92e15e7c472de0f88ec751ca659f8cd8 (HEAD -> main)
Author: salman <s.z.gassem@gmail.com>
Date:   Tue Jul 23 17:01:28 2024 +0300

    Second Git lab commit

commit d908ec404e121947af5db0cd7a915e19adb1c7b7
Author: salman <s.z.gassem@gmail.com>
Date:   Tue Jul 23 17:00:53 2024 +0300

    First Git lab commit
(base) salman@Salmans-MacBook-Air test01 % touch third.txt
(base) salman@Salmans-MacBook-Air test01 % touch fourth.txt
(base) salman@Salmans-MacBook-Air test01 % git add .
(base) salman@Salmans-MacBook-Air test01 % git status
On branch main

```

```
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   fourth.txt
        new file:   third.txt

[(base) salman@Salmans-MacBook-Air test01 % git commit -m "Third and Fourth Git lab commit"
[main 9b1009d] Third and Fourth Git lab commit
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 fourth.txt
 create mode 100644 third.txt
[(base) salman@Salmans-MacBook-Air test01 % git log
zsh: command not found: git
[(base) salman@Salmans-MacBook-Air test01 % git log
commit 9b1009d691cad607f66a895e5628d7a4db08c46d (HEAD -> main)
Author: salman <s.z.gassem@gmail.com>
Date:   Tue Jul 23 17:02:24 2024 +0300

    Third and Fourth Git lab commit

commit 20ddee92e15e7cba472de0f88ec751ca659f8cd8
Author: salman <s.z.gassem@gmail.com>
Date:   Tue Jul 23 17:01:28 2024 +0300

    Second Git lab commit

commit d908ec404e121947af5db0cd7a915e19adb1c7b7
Author: salman <s.z.gassem@gmail.com>
Date:   Tue Jul 23 17:00:53 2024 +0300

    First Git lab commit
(base) salman@Salmans-MacBook-Air test01 %
```

Git-Lab2

Local[Branches and Merging]

من خلال هذا التمرين سنقوم باستخدام مفهوم Branches وعملية دمجه.

الخطوات

- قم بفتح terminal او git bash الخاص بجهازك من ثمّ الدخول على مجلد test01 الذي قمنا بإنشائه في التمرين السابق.
- قم باستعراض جميع Branches المتواجدة في المشروع.
- الآن قم بإنشاء فرع جديد بالاسم new-feature من ثمّ الانتقال اليه. في فرع new-feature قم بإنشاء ملف بالاسم lab02.txt من ثمّ عمل add و commit لهذا الملف. الآن قم باستعراض الملفات خلال وجودك في الفرع new-feature.
- قم بالعودة الى الفرع master او main من ثمّ استعراض الملفات مره أخرى. ستلاحظ عدم وجود lab02.txt في هذا الفرع، بالتالي سنقوم بعملية دمج للفرعين حتى يكون الملف متواجد في الفرق master او main. قم بدمج الفروع. الآن قم باستعراض الملفات مره أخرى، ستجد أن ملف lab02.txt متواجد في الفرع master او main.

Solution:

```
[(base) salman@Salmans-MacBook-Air test01 % git branch
* main
[(base) salman@Salmans-MacBook-Air test01 % git branch new-feature
[(base) salman@Salmans-MacBook-Air test01 % git branch
* main
  new-feature
[(base) salman@Salmans-MacBook-Air test01 % git checkout new-feature
Switched to branch 'new-feature'
[(base) salman@Salmans-MacBook-Air test01 % touch lab02.txt
[(base) salman@Salmans-MacBook-Air test01 % git add .
[(base) salman@Salmans-MacBook-Air test01 % git commit -m "Adding file to new-feature branch"
[new-feature 95fd067] Adding file to new-feature branch
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 lab02.txt
[(base) salman@Salmans-MacBook-Air test01 % ls
first.txt      fourth.txt     lab02.txt      second.txt     third.txt
[(base) salman@Salmans-MacBook-Air test01 % git checkout main
Switched to branch 'main'
[(base) salman@Salmans-MacBook-Air test01 % ls
first.txt      fourth.txt     second.txt     third.txt
[(base) salman@Salmans-MacBook-Air test01 % git merge new-feature
Updating 9b1009d..95fd067
Fast-forward
 lab02.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 lab02.txt
[(base) salman@Salmans-MacBook-Air test01 % ls
first.txt      fourth.txt     lab02.txt      second.txt     third.txt
[(base) salman@Salmans-MacBook-Air test01 % git branch
* main
  new-feature
[(base) salman@Salmans-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.

Solution:

```
[(base) salman@Salmans-MacBook-Air test01 % git remote -v
[(base) salman@Salmans-MacBook-Air test01 % git remote add GitLabHW https://github.com/SalmanGassem/Git-Lab.git
[(base) salman@Salmans-MacBook-Air test01 % git remote -v
GitLabHW          https://github.com/SalmanGassem/Git-Lab.git (fetch)
GitLabHW          https://github.com/SalmanGassem/Git-Lab.git (push)
[(base) salman@Salmans-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 GitLabHW main

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

[(base) salman@Salmans-MacBook-Air test01 % git status
On branch main
nothing to commit, working tree clean
[(base) salman@Salmans-MacBook-Air test01 % ls
first.txt      fourth.txt    lab02.txt     second.txt    third.txt
[(base) salman@Salmans-MacBook-Air test01 % git push GitLabHW
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 GitLabHW main

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

[(base) salman@Salmans-MacBook-Air test01 % git push --set-upstream GitLabHW main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (9/9), 826 bytes | 826.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/SalmanGassem/Git-Lab.git
 * [new branch]      main -> main
branch 'main' set up to track 'GitLabHW/main'.
[(base) salman@Salmans-MacBook-Air test01 % git status
On branch main
Your branch is up to date with 'GitLabHW/main'.

nothing to commit, working tree clean
(base) salman@Salmans-MacBook-Air test01 %
```


Markdown-Lab1 (Bonus)

Create Markdown file (Readme) for your repository

Solution:

```
# Data Science
## What is Data Science?
**Data Science** is a multidisciplinary field that combines statistics, computer science, and domain knowledge.

### Data Types:
- Structured Data
- Semi-Structured Data
- Unstructured Data

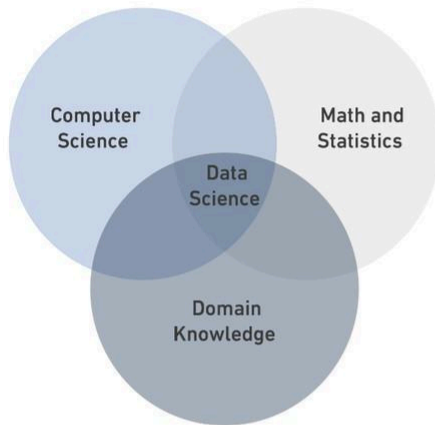
| Structured | Semi-Structured | Unstructured |
|--|--|--|
| It is data that does not conform to a schema but has some structure. | It is data that does not conform to a schema but has some structure. | It is data that has no schema. |
| e.g: tables | e.g: xml | e.g: photos |
```python
print("DS Bootcamp")
```

[Read More](https://en.wikipedia.org/wiki/Data_science)
```


Data Science

What is Data Science?

Data Science is a *multidisciplinary field* that combines statistics, computer science, and domain knowledge.



Data Types:

- Structured Data
- Semi-Structured Data
- Unstructured Data

| Structured | Semi-Structured | Unstructured |
|--|--|--------------------------------|
| It is data that has been organized into a strict schema. | It is data that does not conform to a schema but has some structure. | It is data that has no schema. |
| e.g: tables | e.g: xml | e.g: photos |

```
print ("DS Bootcamp")
```

[Read More](#)

Hint: for more information, go to: <https://www.markdownguide.org/cheat-sheet/> Online editor: <https://stackedit.io/app#>

Requirements:

Heading style.

H1 for "Data Science".

H2 for "What is Data Science?".

- -
 - ◻
 - H3 for “Data Types”.
- A bold style for the “Data Science” definition.
- An italic style for “multidisciplinary field “.
- Image style for “DS.png” and resize the image to width="400" and height="400".
- Unordered List for Data Types.
- Link style for the following link “https://en.wikipedia.org/wiki/Data_science”.
- Table style for Data Types.
- Code style for Python code prints “DS Bootcamp”.

