

Git_Labs

أكاديمية طويق
Tuwaiq Academy



By: Nourah Almutlg

Git-Lab1

Local[Add, Commit, Log]

في هذا التمرين سنقوم باستخدام Git للتعامل مع الملفات وإدارتها بشكل محلي بحيث سنقوم بإنشاء ملفات ز 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 تم 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 تم حفظه على repository.
- ♦ بعد عمل commit قم بإنشاء ملف آخر بالاسم second.txt من تم 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.

```
labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop
$ mkdir git-test

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop
$ cd git-test

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test
$ mkdir test01

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test
$ cd test01/

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01
$ git init
Initialized empty Git repository in C:/Users/labbo/OneDrive/Desktop/git-test/test01/.git/

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ ls -a
./ ../ .git/

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ touch first.txt

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ git add .

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ git commit -m "first_text"
[master (root-commit) 0b06692] first_text
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 first.txt

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ touch second.txt

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ git add .

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ git commit -m "second_text"
[master 4f1f786] second_text
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 second.txt

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ touch third.txt

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ touch fourth.txt

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ git add .
```

```

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ touch fourth.txt

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ git add .

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ git commit -m "third commit"
[master 8815037] third commit
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 fourth.txt
create mode 100644 third.txt

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ git log
commit 88150371297e7b1b11458a2c1f7bcab99589dcbf (HEAD -> master)
Author: MOUSA-CS <mousa.ub11@gmail.com>
Date: Wed Jul 24 00:32:15 2024 +0300

    third commit

commit 4f1f786c72b9cb0e7bd5bb436f08a52629bca5ce
Author: MOUSA-CS <mousa.ub11@gmail.com>
Date: Wed Jul 24 00:30:46 2024 +0300

    second_text

commit 0b066929d5e6b42f7cc5e5a1123a3d539f9a91fc
Author: MOUSA-CS <mousa.ub11@gmail.com>
Date: Wed Jul 24 00:29:45 2024 +0300

    first_text

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$

```

Git-Lab2

Local[Branches and Merging]

من خلال هذا التمرين سنقوم باستخدام مفهوم Branches وعملية دمجهم.

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 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.

```
labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop
$ cd git-test/test01

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ git branch
* master

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ git branch -a
* master

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ git checkout -b feature-new
Switched to a new branch 'feature-new'

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (feature-new)
$ git branch feature-new
fatal: a branch named 'feature-new' already exists

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (feature-new)
$ touch lab02.txt

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (feature-new)
$ git add .

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (feature-new)
$ git commit -m "commit new branch as lab02"
[feature-new 76e6ca8] commit new branch as lab02
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 lab02.txt

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (feature-new)
$ ls
first.txt  fourth.txt  lab02.txt  second.txt  third.txt

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (feature-new)
$ git checkout main
error: pathspec 'main' did not match any file(s) known to git
```

```

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (feature-new)
$ git checkout master
Switched to branch 'master'

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ ls
first.txt  fourth.txt  second.txt  third.txt

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ git merge feature-new
Updating 8815037..76e6ca8
Fast-forward
 lab02.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 lab02.txt

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ ls
first.txt  fourth.txt  lab02.txt  second.txt  third.txt

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ |

```

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 فيها.
- 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 تم 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 تم اعادة تحميلها refresh وستظهر لك جميع الملفات التي قمت بانشاءها في التمارين السابقة.
- ♦ في حالة أردت، قم بانشاء ملف جديد محلياً من تم عمل git push مرة أخرى لتلاحظ التحديثات الحاصلة بشكل أكبر.

```
labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ git remote -v

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ git remote add day3_gitlab https://github.com/mousa-dsc/day3-.git

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ git push
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream day3_gitlab master

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

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$ git push --set-upstream day3_gitlab master
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 12 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (9/9), 815 bytes | 815.00 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/mousa-dsc/day3-.git
 * [new branch]      master -> master
branch 'master' set up to track 'day3_gitlab/master'.

labbo@DESKTOP-3DDJIGM MINGW64 /c/users/labbo/onedrive/desktop/git-test/test01 (master)
$
```

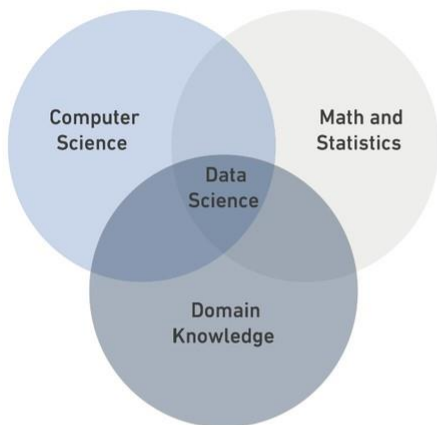
Markdown-Lab1 (Bonus)

Create Markdown file (Readme) for your repository

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".

