

Git Labs

By: Nourah Almutlg

Git-Lab1

Local[Add, Commit, Log]

في هذا التمرين سنقوم باستخدام Git للتعامل مع الملفات وإدارتها بشكل مدوّلي بحيث سنقوم بإنشاء ملفات ز يقبوتتبعها 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 ثمّ 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 ثمّ حفظة على repository.
- بعد عمل commit قم بإنشاء ملف آخر بالاسم second.txt من ثمّ 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.

Steps :

- C:\Users\User\Desktop>cd
C:\Users\User\Desktop
- C:\Users\User\Desktop>mkdir git-test
- C:\Users\User\Desktop>cd git-test
- C:\Users\User\Desktop\git-test>mkdir test01
- C:\Users\User\Desktop\git-test>cd test01
- C:\Users\User\Desktop\git-test\test01>git init
Initialized empty Git repository in C:/Users/User/Desktop/git-test/test01/.git/

- C:\Users\User\Desktop\git-test\test01>git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
- C:\Users\User\Desktop\git-test\test01>echo "this is first file" > first.txt
- C:\Users\User\Desktop\git-test\test01>git add first.txt
- C:\Users\User\Desktop\git-test\test01>git commit -m "Add first.txt file"
[master (root-commit) f532564] Add first.txt file
1 file changed, 1 insertion(+)
create mode 100644 first.txt
- C:\Users\User\Desktop\git-test\test01>echo "this is second file" > second.txt
- C:\Users\User\Desktop\git-test\test01>git add second.txt
- C:\Users\User\Desktop\git-test\test01>git commit -m "Add second.txt file"
[master 0332ae5] Add second.txt file
1 file changed, 1 insertion(+)
create mode 100644 second.txt
- C:\Users\User\Desktop\git-test\test01>echo "this is third file" > third.txt
- C:\Users\User\Desktop\git-test\test01>git add third.txt
- C:\Users\User\Desktop\git-test\test01>git commit -m "Add third.txt file"
[master 4adc186] Add third.txt file
1 file changed, 1 insertion(+)
create mode 100644 third.txt
- C:\Users\User\Desktop\git-test\test01>echo "this is forth file" > forth.txt
- C:\Users\User\Desktop\git-test\test01>git add forth.txt
- C:\Users\User\Desktop\git-test\test01>git commit -m "Add forth.txt file"
[master b22160a] Add forth.txt file
1 file changed, 1 insertion(+)
create mode 100644 forth.txt

- C:\Users\User\Desktop\git-test\test01>git log
commit b22160a070849a56da795c03aa504ffb0c38bf21 (HEAD -> master)
Author: Fahad Almalki <faemalki@gmail.com>
Date: Tue Apr 30 03:39:16 2024 +0300

Add forth.txt file

commit 4adc186f676b00896bd03a6869ff604a9736446a
Author: Fahad Almalki <faemalki@gmail.com>
Date: Tue Apr 30 03:38:07 2024 +0300

Add third.txt file

commit 0332ae5d326f0b16a1d574f269415e0d3f93fd92
Author: Fahad Almalki <faemalki@gmail.com>
Date: Tue Apr 30 03:36:30 2024 +0300

Add second.txt file

commit f5325649cca6a4603d6c8fcbf544f21760e66222
Author: Fahad Almalki <faemalki@gmail.com>
Date: Tue Apr 30 03:34:11 2024 +0300

Add first.txt file

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 بإنشاؤه 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.

- C:\Users\User\Desktop\git-test\test01>
 - C:\Users\User\Desktop\git-test\test01>git branch
 - master
 - C:\Users\User\Desktop\git-test\test01>git branch new-feature
 - C:\Users\User\Desktop\git-test\test01>git checkout new-feature
Switched to branch 'new-feature'
 - C:\Users\User\Desktop\git-test\test01>echo "this is lab02.txt" > lab02.txt
 - C:\Users\User\Desktop\git-test\test01>git commit -m "lab02.txt file"
 - C:\Users\User\Desktop\git-test\test01>dir
- ```
04/30/2024 04:16 AM <DIR> .
04/30/2024 04:16 AM <DIR> ..
04/30/2024 03:31 AM 23 first.txt
04/30/2024 03:38 AM 23 forth.txt
04/30/2024 04:16 AM 22 lab02.txt
```

```
04/30/2024 03:35 AM 24 second.txt
04/30/2024 03:37 AM 23 third.txt
 5 File(s) 115 bytes
 2 Dir(s) 4,354,084,864 bytes free
```

- C:\Users\User\Desktop\git-test\test01>git checkout master  
Switched to branch 'master'
- C:\Users\User\Desktop\git-test\test01>git merge new-feature
- C:\Users\User\Desktop\git-test\test01>dir

Directory of C:\Users\User\Desktop\git-test\test01

```
04/30/2024 04:24 AM <DIR> .
04/30/2024 04:24 AM <DIR> ..
04/30/2024 03:31 AM 23 first.txt
04/30/2024 03:38 AM 23 forth.txt
04/30/2024 04:16 AM 22 lab02.txt
04/30/2024 03:35 AM 24 second.txt
04/30/2024 03:37 AM 23 third.txt
 5 File(s) 115 bytes
 2 Dir(s) 4,354,084,864 bytes free
```

---

---

# 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 عليها 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.

- C:\Users\User\Desktop\git-test\test01>git remote -v
- C:\Users\User\Desktop\git-test\test01>git remote add origin https://github.com/Fahad20m/Git-Lab3.git
- C:\Users\User\Desktop\git-test\test01>git push -u origin master
- C:\Users\User\Desktop\git-test\test01>git push -u origin master

---

---

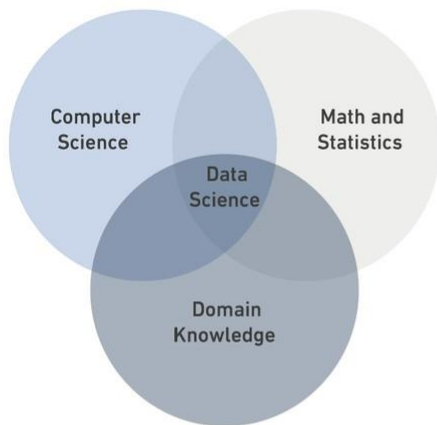
# Markdown-Lab1 (Bonus)

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](https://en.wikipedia.org/wiki/Data_science)".
  - ◆ Table style for Data Types.
  - ◆ Code style for Python code prints "DS Bootcam".

