

Git Labs

By: Nourah Almutlg

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.
- 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.
- 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.

Git-Lab1 Solution

- cd desktop
- mkdir git-test
- cd git-test
- mkdir test01
- cd test01
- git init
- cd > first.txt

- `git add .`
 - `git commit -m "add first file"`
 - `cd > second.txt`
 - `git add .`
 - `git commit -m "add second file"`
 - `cd > third.txt`
 - `cd > fourth.txt`
 - `git add .`
 - `git commit -m "add files 3 and 4"`
 - `git log`
-
-

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 ثم الدخول على مجلد test01 الذي قمنا بانشاؤه في التمرين السابق.
- قم باستعراض جميع Branches المتواجدة في المشروع.
- الآن قم بانشاء فرع جديد بالاسم new-feature من ثم الانتقال اليه. في فرع new-feature قم بانشاء ملف بالاسم lab02.txt من ثم 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.

Git-Lab2 Solution

- `git branch`
- `git branch new-feature`
`git checkout new-feature`
- `cd > lab02.txt`
`git add .`
`git commit -m "add lab2"`
- `dir`
- `git checkout master`
`dir`
- `git merge new-feature`
- `dir`

Git-Lab3

GitHub Remotely

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

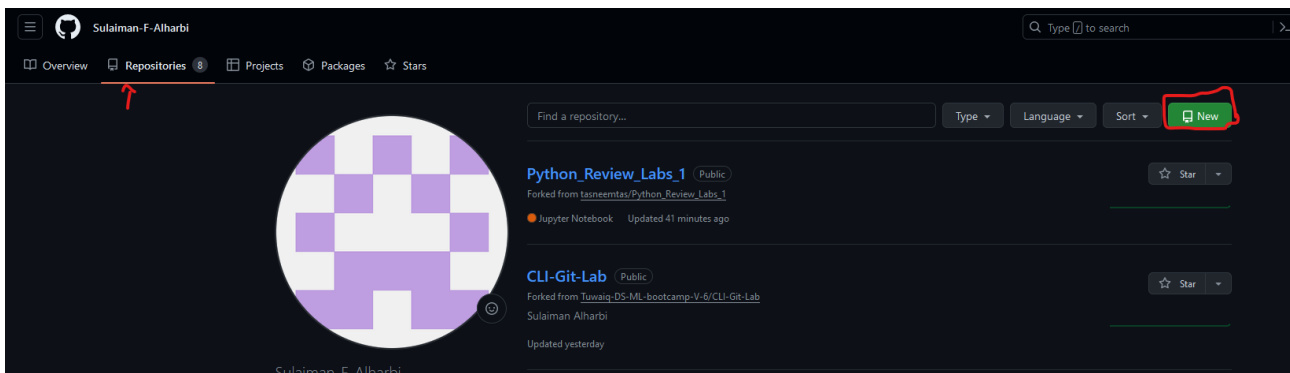
الخطوات

- 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.
- 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.

Git-Lab3 Solution

- git remote
- git remote add origin https://github.com/Sulaiman-F-Alharbi/Git-Lab3.git
- git push -u origin master
- git push

Creating github repository:




Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *

 Sulaiman-F-Alharbi

Repository name *

Git_Lab3


✔ Git_Lab3 is available.

Great repository names are short and memorable. Need inspiration? How about [studious-octo-train](#) ?

Description (optional)

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

Initialize this repository with:

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore


.gitignore template: **None**

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)


Choose a license

License: **None**

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)


 You are creating a public repository in your personal account.





Create repository


 Sulaiman-F-Alharbi / test


Q Type to search

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

 **test** Public

 Pin  Unwatch 1  Fork 0  Star 0

 **Set up GitHub Copilot**
Use GitHub's AI pair programmer to autocomplete suggestions as you code.
[Get started with GitHub Copilot](#)

 **Add collaborators to this repository**
Search for people using their GitHub username or email address.
[Invite collaborators](#)

Quick setup — if you've done this kind of thing before
[Set up in Desktop](#) or [HTTPS](#) [SSH](#) <https://github.com/Sulaiman-F-Alharbi/test.git>
Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# test" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Sulaiman-F-Alharbi/test.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/Sulaiman-F-Alharbi/test.git
git branch -M main
git push -u origin main
```

Sulaiman-F-Alharbi / Git-Lab3

Search Type to search

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Git-Lab3Public

PinUnwatch1Fork0Star0

master1 Branch0 Tags

Go to file

Add fileCode

About

Sulaiman-F-AlharbiSulaiman Alharbi

c28bd91 - yesterday6 Commits

first.txt	add first file	yesterday
fourth.txt	ad files 3 and 4	yesterday
lab02.txt	Sulaiman Alharbi	yesterday
second.txt	add second file	yesterday
third.txt	ad files 3 and 4	yesterday

README

Add a README

Help people interested in this repository understand your project by adding a README.

About

No description, website, or topics provided.

Activity0 stars1 watching0 forks

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

© 2024 GitHub, Inc. TermsPrivacySecurityStatusDocsContactMarketplaceDo not store sensitive information



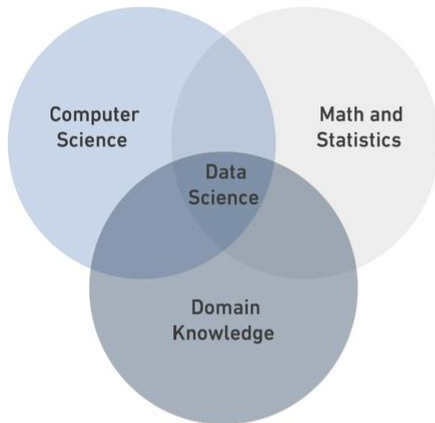
Markdown-Lab1 (Bonus)

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”.

