

# 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 بانشاء ملفات نصية وتتبعها واحد تلو الآخر كالتالي.

#### Code used

1- C:\Users\fisai>cd desktop

2- C:\Users\fisai\Desktop>mkdir git-test

3- C:\Users\fisai\Desktop>cd git-test

4-

#### الخطوات

- قم بفتح terminal او git bash على جهازك
- قم بالانتقال الى مجلد Desktop.
- بداخل مجلد Desktop قم بانشاء 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 بانشاء repository جديد باستخدام الأمر git init.
- بهذا الشكل اصبح git قادر على تتبع المشروع، بالتالي قم بانشاء ملف first.txt من ثم حفظه على repository.
- بعد عمل commit قم بانشاء ملف آخر بالاسم second.txt من ثم حفظه أيضاً على repository.
- بعد الانتهاء من حفظ ملف first.txt و second.txt قم بانشاء 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-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 بانشاؤه في التمرين السابق.
- قم باستعراض جميع Branches المتواجدة في المشروع.
- الآن قم بانشاء فرع جديد بالاسم new-feature من ثمّ الانتقال اليه.
- في فرع new-feature قم بانشاء 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.

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

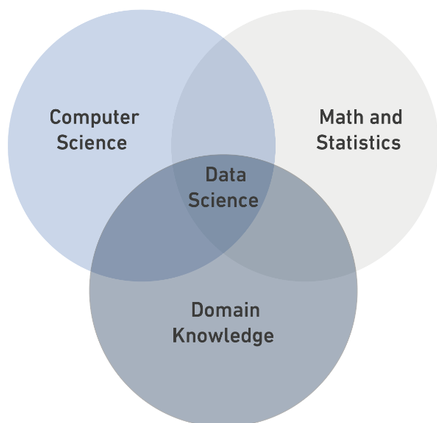
## Markdown-Lab1 (Bonus)

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")
```

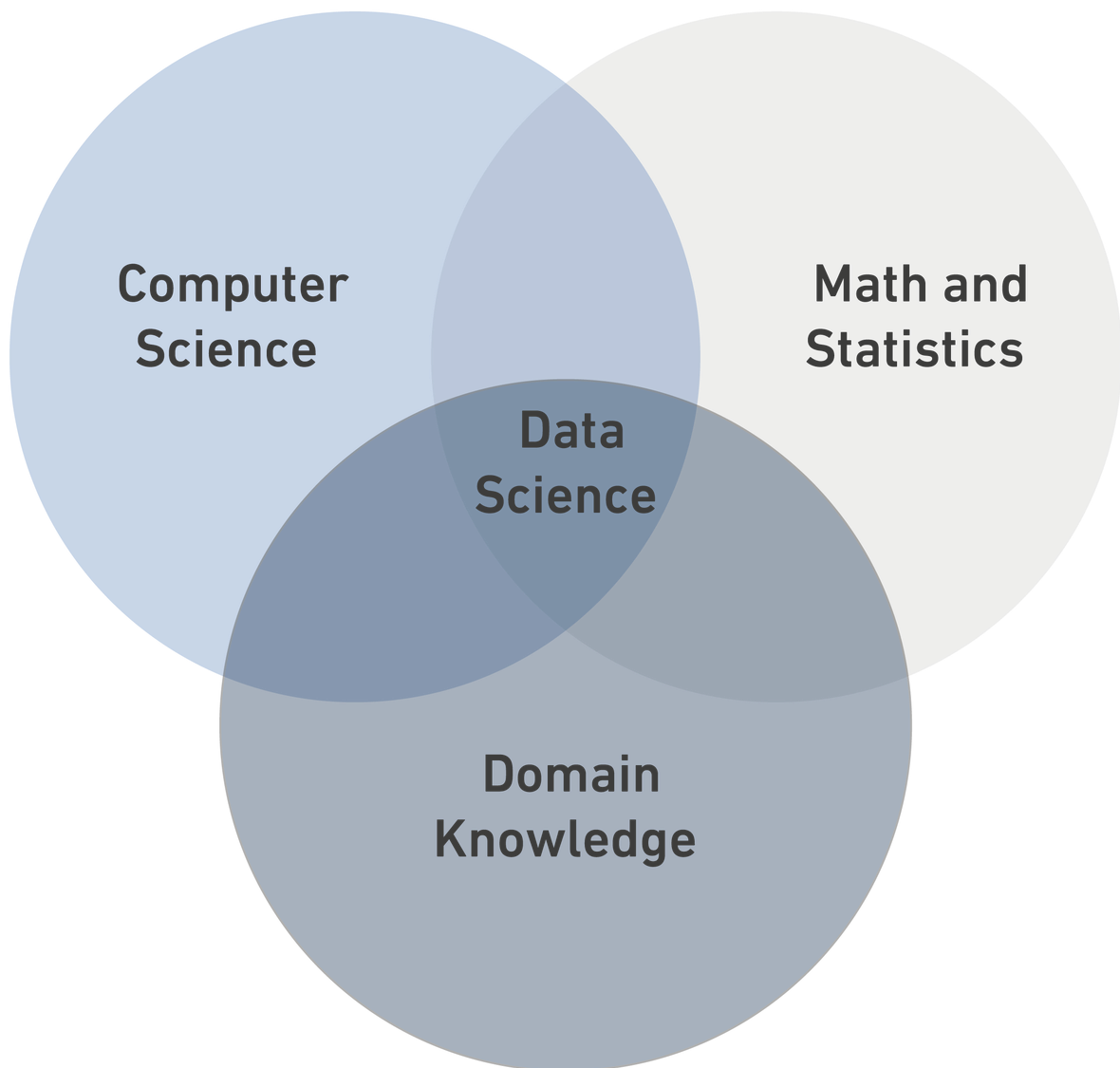
[Read More](#)

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 Bootcamp".



## Code used Lab1

- 1- C:\Users\fisai>cd desktop
- 2- C:\Users\fisai\Desktop>mkdir git-test
- 3- C:\Users\fisai\Desktop>cd git-test
- 4- C:\Users\fisai\Desktop\git-test>mkdir test01
- 5- C:\Users\fisai\Desktop\git-test>cd test01
- 6- C:\Users\fisai\Desktop\git-test\test01>git init
- 7- C:\Users\fisai\Desktop\git-test\test01>cd > first.txt
- 8- C:\Users\fisai\Desktop\git-test\test01>git add .
- 9- C:\Users\fisai\Desktop\git-test\test01>git commit -m "first1"
- 10- C:\Users\fisai\Desktop\git-test\test01>cd > second.txt
- 11- C:\Users\fisai\Desktop\git-test\test01>git add .
- 12- C:\Users\fisai\Desktop\git-test\test01>git commit -m "file2"
- 13- C:\Users\fisai\Desktop\git-test\test01>cd > third.txt
- 14- C:\Users\fisai\Desktop\git-test\test01>cd > fourth.txt
- 15- C:\Users\fisai\Desktop\git-test\test01>git add .
- 16- C:\Users\fisai\Desktop\git-test\test01>git commit -m "file3-4"
- 17- C:\Users\fisai\Desktop\git-test\test01>git log

## Code used Lab2

- 1- C:\Users\fisai\Desktop\git-test\test01>git branch
- 2- C:\Users\fisai\Desktop\git-test\test01>git branch new-feature
- 3- C:\Users\fisai\Desktop\git-test\test01>git checkout new-feature
- 4- C:\Users\fisai\Desktop\git-test\test01>cd > lab02.txt
- 5- C:\Users\fisai\Desktop\git-test\test01>git add .
- 6- C:\Users\fisai\Desktop\git-test\test01>git commit -m "featureFaisal"
- 7- C:\Users\fisai\Desktop\git-test\test01>git checkout master
- 8- C:\Users\fisai\Desktop\git-test\test01>dir
- 9- C:\Users\fisai\Desktop\git-test\test01>git merge new-feature
- 10- C:\Users\fisai\Desktop\git-test\test01>dir

## Code used Lab3

- 1- C:\Users\fisai\Desktop\git-test\test01>git remote
- 2- C:\Users\fisai\Desktop\git-test\test01>git remote add origin [https://github.com/FaisalAlossaimi/Lab1\\_3.git](https://github.com/FaisalAlossaimi/Lab1_3.git)
- 3- C:\Users\fisai\Desktop\git-test\test01>git branch -M master
- 4- C:\Users\fisai\Desktop\git-test\test01>git push -u origin master
- 5-

