Git\_Labs



By: Nourah Almutlg

# Git-Lab1

## Local[Add, Commit, Log]

في ھذا التمرین سنقوم باستخدام Git للتعامل مع الملفات وادارتھا بشكل محليّ بحیث سنقوم بانشاء ملفات نصّیة وتتبعھا واحد تلوىالآخر كالتالي.

### الخطوات

قم بفتح terminal او git bash على جھازك.قم بالانتقال الى مجلدّ Desktop.

Cd Desktop

بداخل مجلدّ Desktop قم بانشاء مجلدّ جدید بالاسم git-test.

mkdir git-test

قم بالدخول على مجلدّ git-test من ثمّ انشاء مجلدّ جدید بالاسم test01. قم

cd git-test

mkdir test01

بالدخول على مجلدّ test01 من خلال terminal او git bash. الآن قم بانشاء repository جدید باستخدام الأمر git init.

git init

بھذا الشكل اصبح git قادر على تتبع المشروع، بالتالي قم بانشاء ملف first.txt من ثمّ حفظة على repository.

cd> first.txt

بعد عمل commit قم بانشاء ملف آخر بالاسم second.txt من ثمّ حفظة أیضاً على repository.

git commit -m "ABDULAZIZ"

cd> second.txt

بعد الانتھاء من حفظ ملف first.txt و second.txt قم بانشاء ملفین آخرین بالاسم third.txt و fourth.txt.

cd> third.txt

cd> fourth.txt

قم بحفظھم على repository من خلال استخدام الأمر . add وعمل commit لھم.

git add first.txt

git second.txt

git add third.txt

git add fourth.txt

الآن قم باستعراض جمیع commits التي قمت بعملھا من خلال الأمر git log.

git commit -m "ABDULAZIZ"

git log

# Git-Lab2

## Local[Branches and Merging]

من خلال ھذا التمرین سنقوم باستخدام مفھوم Branches وعملیّّة دمجھم.

### الخطوات

قم بفتح terminal او git bash الخاص بجھازك من ثمّ الدخول على مجلدّ test01 الذي قمنا بانشاؤه في التمرین السابق.

Cd test01

قم باستعراض جمیع Branches المتواجدة في المشروع.

git branch

الآن قم بانشاء فرع جدید بالاسم new-feature من ثمّ الانتقال الیھ. في فرع new-feature قم بانشاء ملف

بالاسم lab02.txt من ثمّ عمل add و commit لھذا الملف. الآن قم باستعراض الملفات خلال وجودك في الفرع

new-feature.

git branch new-feature

git checkout new-feature

cd> lab02.txt

git add lab02.txt

git commit -m "Abdulaziz""

ls

قم بالعودة الى الفرع master او main من ثمّ استعراض الملفات مره أخرى. ستلاحظ عدم وجود lab02.txt في ھذاالفرع، بالتالي سنقوم بعملیّّة دمج للفرعین حتى یكون الملف متواجد في الفرق master او main. قم بدمج الفروع.

git checkout master

git marge new-feature.

الآن قم باستعراض الملفات مره أخرى، ستجد أن ملف lab02.txt متواجد في الفرع master او main.

ls

# Git-Lab3

## GitHub Remotely

في ھذا التمرین سنقوم برفع المجلدّ test01 الذي قمنا باستخدامھ في التمارین السابقة على Github.

### الخطوات

قم بالذھاب الى مسار مجلدّ test01.

Cd test01

قم باستعراض remote المتواجدة فیھا

git remote -v

نلاحظ عدم وجود remote سابق والسبب ھو اننا كنا نعمل بشكل محليّ لكن الآن سنقوم برفع المشروع والعمل علیھ

.remotely

قم بالذھاب الى Github وانشاء repository جدید من ثمّ نسخ الرابط الخاص بھ.

قم بالعودة الى terminal او git bash من ثمّ اضافة remote جدید على الرابط الذي تمّ نسخة.

git remote add origin https://github.com/abdulaziz12121/tu-test.git

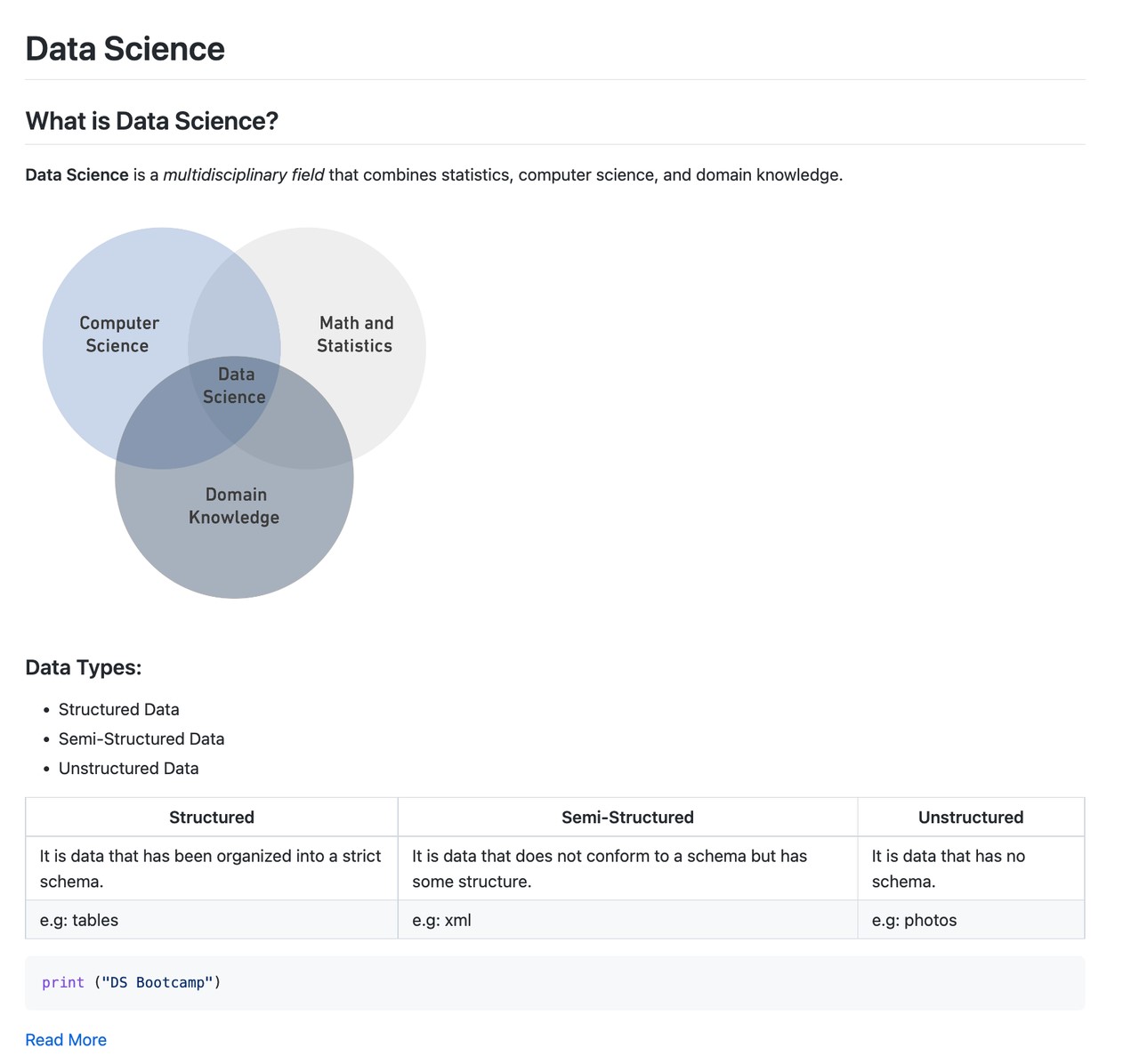
الآن قم برفع جمیع الملفات والتعدیلات الى repository المتواجد في Github من خلال استخدام الأمر git push. قم بالانتقال الى صفحة Github من ثمّ اعادة تحمیلھا refresh وستظھر لك جمیع الملفات التي قمت بانشائھا في التمارینالسابقة. في حالة أردت،

قم بانشاء ملف جدید محلیّاً من ثمّ عمل git push مرّة أخرى لتلاحظ التحدیثات الحاصلة بشكل أكبر.

git push -u origin master

# Markdown-Lab1 (Bonus)

Create Markdown file (Readme) for your repository



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

