Git & Github

Made By Nabil Hany

ده ملف تدوين لمذاكرة كورس تعلم Git & Github من منصة المدرسة لتعليم البرمجة

الملف من غير مشاهدتك للدورة ملهوش أي قيمة لأن الي شاف مش زي الي سمع حاول تجمع ما بين الاتنين والدورة مجانية و شرحها كويس جدا وهيضيفلك كتير

https://almdrasa.com/tracks/programming-foundations/courses/git-github/

الملف ده عبارة عن كتابة للملخصات الكتابية داخل الدورة واضافات من عندي اثناء المذاكرة لحاجات كانت بالنسبالي صعبة في الفهم ومشاكل ممكن تقابلك فمتنسنيش من دعائك ولو حابب تتابعني علي لينكد ان هسبلك لينك البروفايل بتاعي

LinkedIn

الموضوع: _____ التاريخ: ١٨/ ٥/ ٢٠٢٥

* تعلم Git و Github *

مقدمة الدورة

ماهو Version control ؟

- أداة تدير التغيرات المتعلقة بالأكواد والملفات داخل المشروع.
- أداة تجعلك تتبع جميع التعديلات بالتوقيتات الزمنية لكل تعديل.
- أداة تتيح لك تخزين نسخة من الملفات على جهازك الشخصي أو على الأداة المستخدمة لتطبيقه ال Version Control مثل Git

ماهو Git ؟

- هو نظام تحكم في الإصدارات يستخدم على نطاق واسع في تطوير البرمجيات وإدارتها بكفاءة، بالإضافة إلى القدرة على العمل بشكل تعاوني مع فرق التطوير.
- Git يسمح للمستخدمين إنشاء نسخ من المشروع بسهولة (Branches) ودمج التغيرات بين هذه النسخ بشكل آمن ومنظم.
- يعتمد Git على نموذج توزيعي حيث يمكن للمطورين العمل على المشروع بشكل مستقل ودمج تغيراتهم فيما بعد.
- هذا يسهل العمل على مشاريع كبيرة ويسمح بتعاون فعال بين مختلف أعضاء الفريق.. وهو من أشهر أنواع ال Version Control.
- يوجد العديد من البرامج المختلفة التي تستخدم لتطبيقه الـ Version Control مثل :

- CVS - SVN - PerForce - Bazzare

التاريخ: / /

الموضوع:

ماهو الفرق بين Git و Github ؟

- الـ Git هو عبارة عن Version Control يدير التغيرات المتعلقة بالأكواد والملفات داخل المشروع.
- الـ Github برنامج لإدارة وتنظيم الملفات (Git Repos) على الـ Cloud. ويسمح بمشاركة المشاريع وتعديلها من أى جهاز.

أساسيات Git

تسطيب git - install Git

1- اعمل سرش عن Git ثم Git For Windows /X64 setup
2- بعد إختيار مكان التسطيب نقوم بفتح الملف وإتمام التسطيب
3- ثم نقوم بفتح الـ Git Bash

* كتابة أمر git-version يظهر لك الـ Version الخاص بك
  git --version
* عند كتابة git فقط سيظهر لك كل أوامر git

عمل أكونت على github ☑

رفع المشروع على Git

- يمكن كتابة الأكواد أو الأوامر فى الـ cmd ويمكن كتابتها أيضا فى الـ Git Bash ولكن يفضل فتح فولدر المشروع على فيجوال إستوديو كود ومن ثم فتح الـ Terminal الخاص بالبرنامج وكتابة الأوامر مثل أمر init

الموضوع: _____ التاريخ: / /

كما أن بيئة العمل تسمى Working directory
لها مُسمى كدا معناها ال Folder اللي إحنا شغالين فيه

الأمر git init ⇐ يقوم بإنشاء مستودع git فارغ أو إعادة تهيئة مستودع موجود.
create an empty Git repository or re initialize an existing one

عند كتابة الأمر git init -h ⇐ تظهر قائمة help

ما هي ال git init ؟

- ال git init هي أمر يستخدم في Git لإنشاء مستودع (RePository) جديد. يتم إستخدام هذا الأمر عندما تحتاج إلى إنشاء مستودع جديد لمشروعك الحالي أو لبدء مشروع جديد.
- عندما تقوم بتشغيل git init يقوم Git بإنشاء مجلد جديد في المسار الحالي يسمى ".git". يتم إستخدام هذا المجلد لتخزين جميع بيانات Git المتعلقة بمستودعك، مثل تاريخ المشروع وسجلات الإصدارات وغيرها من المعلومات.

git add

- يستخدم هذا الأمر لإضافة الملفات المحددة إلى منطقة الإنتظار (staging area)، والتي تسمح لك بتحديد الملفات التي سيتم تضمينها في ال Commit القادم.
- لإضافة ملفات جديدة يمكنك إستخدام git add متبوعاً بأسماء الملفات

git add File1.txt File2.txt

الموضوع: _____ التاريخ: / /

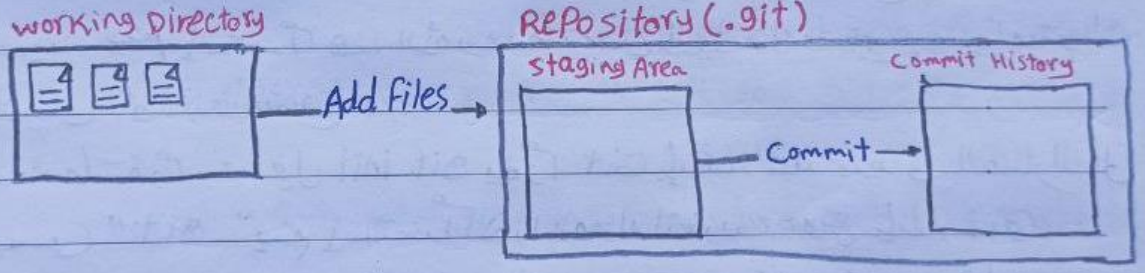
git commit

- يستخدم هذا الأمر لحفظ التغييرات التي تمت إضافتها إلى منطقة الإنتظار (staging area) وتطبيقها على المستودع (Repository) الخاص بك
- لإنشاء commit جديد يمكنك إستخدام git commit متبوعاً برسالة commit وصفاً للتغييرات التي قمت بها. على سبيل المثال:-

git commit -m "أضفت ملفات جديدة"

- يتم تخزين التغييرات التي تم إجراؤها في commit بشكل دائم في المستودع الخاص بك، مما يسمح لك بالعودة إليها في أي وقت لتحديث المشروع أو إستعادة نسخة سابقة منه.

Git Work Flow



1- الأمر git init يقوم بإنشاء مستودع فارغ
2- الأمر git add يقوم بإضافة الملفات إلى منطقة الإنتظار
3- الأمر git commit يستخدم في حفظ التغييرات التي تمت إضافتها إلى منطقة الإنتظار وتطبيقها على المستودع الخاص بك

الموضوع: _____ التاريخ: / /

- لإزالة التراك من على ملف نكتب الأمر git rm --cached اسم الملف
- لمعرفة الحالة نكتب الأمر git status
- ال commit هو snap shot كل commit عبارة عن حالة يكون عليها الكود في ذلك الوقت
- الأمر cd معناة Change Directory وهو من أوامر ال cmd يقوم بتغيير مكان ال Folder الحالي ويساعدك على الإنتقال من فولدر لآخر
- في حاجة هتحصلك مع أول commit هتضيف user لنفسك :-

* please tell me who you are : **

Run

git config --global user.email "nhany474@gmail.com"

git config --global user.name "Nabil-Hany22"

الحتة دي من أول الدقيقة 5 ← درس ال Git commit

git Log

- هي أمر في Git يستخدم لعرض سجل التزامات commits المستودع.
- يعرض هذا الأمر قائمة بكل الإلتزامات commits الموجودة في الفرع الحالي للمستودع ويشمل كل إلتزام معلومات مثل الشفرة التعريفية للالتزام (commit hash) والمؤلف Author والتاريخ والرسالة التوضيحية للإلتزام.
- بشكل عام Git Log يمكن إستخدامه لفهم تاريخ المستودع وتحليل تطوراته بشكل أفضل، كما يمكن إستخدامه كأداة لتحديد مكان الأخطاء ومراجعتها.

الموضوع: _____ التاريخ: / /

.gitignore

. هو ملف يتم إنشاؤه داخل مجلد العمل الخاص بمشروع Git ويحتوى على قائمة بالملفات والمجلدات التى يجب تجاهلها عند تتبع تغييرات المشروع بإستخدام Git

. عندما تنشأ مستودع Git جديد، ستضع ملفاتك ومجلداتك داخل مجلد العمل، وستقوم بتتبع تغييرات هذه الملفات بإستخدام Git ومع ذلك قد يكون هناك ملفات أو مجلدات غير ضرورية أو غير مرغوب فيها، مثل ملفات التكوين أو ملفات تضم معلومات حساسة أو ملفات مؤقتة. يمكن إستخدام ملف (.gitignore) لتعليم Git تجاهل هذه الملفات، حتى لايتم تتبعها ضمن سجلات التاريخ والإصدارات.

لو ملف الـ (.history) مش موجود :-

> ذلك لأنه يجب تثبيت إكستنشن Local History

تطبيق على gitignore. لإلغاء تتبع ملف الـ history . نقوم بإنشاء ملف الـ .gitignore. ثم نكتب بداخله (.history) أو (.history/) حينها لن يحدث تتبع (Tracking) لملف الـ history

الفروع (Branches)

Branches

فى الـ Git, تعرف الـ Branches بأنها تسمح لك بتطوير مسارات مختلفة للتغييرات على مستودع Git الخاص بك. عند إنشاء فرع جديد يمكنك تطوير أجزاء أو إجراء التغييرات دون التأثير على الأجزاء

الموضوع: _____ التاريخ: / /

الأخرى التى تعمل عليها فرقة العمل الأخرى فى نفس المشروع
أى أنك سوف تأخذ نسخة من المشروع وتقوم بالتعديل عليه ولن يتأثر
المشروع الرئيسى بأى تعديل تقوم به، وبعد أن تتأكد أن ماقمت بإضافته
أو تعديله مضبوط تستطيع أن تقوم بإضافته للمشروع الرئيسى.

موقع Learn Git Branching لتعلم الـ Branch Visualization

Git Branch

هو أمر فى Git يستخدم لإنشاء فروع جديدة فى مستودع Git الخاص بك
عندما تنشأ فرعاً جديداً يتم إنشاء نسخة من كل شئ موجود فى الفرع الحالى
بما فى ذلك جميع الملفات وسجل التاريخ الخاص بالمستودع، وتستخدم
هذه الفروع لتجربة وتطوير ميزات جديدة بدون التأثير على الفرع الرئيسى.

لإنشاء فرع جديد، يمكنك إستخدام الأمر git branch متبوعاً بـ اسم الفرع
الجديد

```
git branch new-feture
```

Git checkout

هو أمر فى Git يستخدم للتبديل بين الفروع المختلفة فى مستودع
Git الخاص بك. عند تشغيل هذا الأمر مع إسم الفرع يتم التحويل إلى
هذا الفرع ويصبح الفرع النشط الذى يتم العمل عليه.

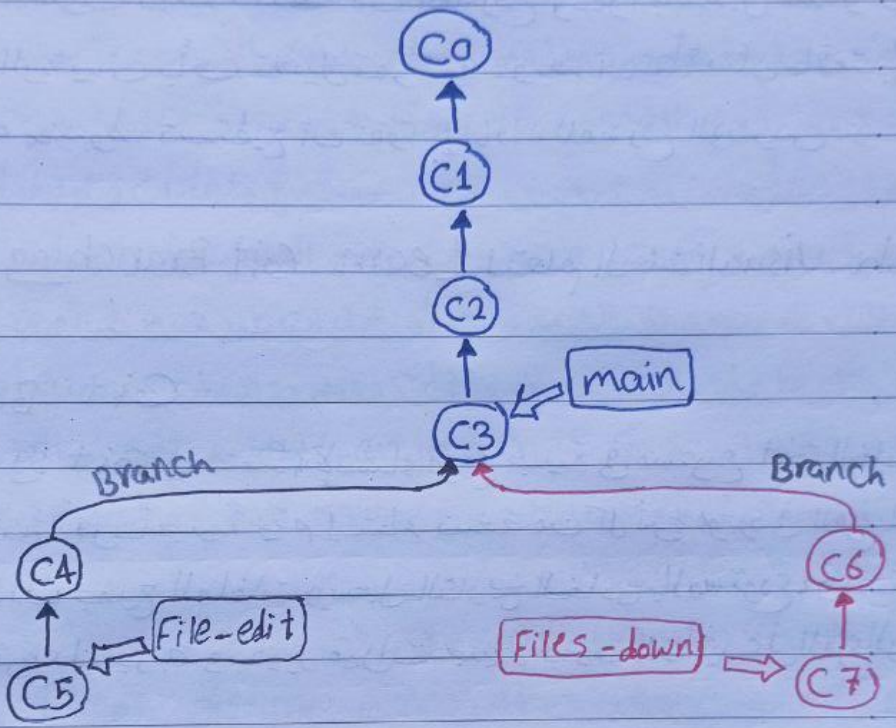
للتبديل إلى فرع معين، يمكن إستخدام الأمر git checkout ومتبوعا باسم الفرع

```
git checkout new-feature
```

بمجرد التبديل إلى الفرع الجديد، يمكنك العمل على الملفات وإجراء التغييرات كما تشاء
دون التأثير على الفرع الرئيسى أو أى فروع أخرى

الموضوع: _____ التاريخ: / /

شرح الـ Branch Visualization



Git Revert

الـ Git Revert لا يحذف أى تعديلات سابقة بل ينشئ إلتزام commit الجديد الذى يعكس حالة المستودع قبل إجراء التعديلات التى تم إجراؤها فى إلتزام commit المحدد. بمعنى آخر يتم إنشاء الإلتزام الجديد الذى يحتوى على تعديلات تعكس الحالة السابقة للمستودع.

لمعرفة المزيد عن كيفية إستخدام Git Revert دعنا نلقى نظرة على المثال التالى

```
git revert abc123
```

الموضوع: Git Revert من Chat-gpt التاريخ: / /

Git Revert هو أمر Command في Git وظيفته التراجع عن commit معين، بس بطريقة آمنة.
لكن مش زى git rest اللى بيرجعك لورا وبيغير التاريخ.
& git revert بيعمل commit جديد بيعكس التغييرات اللى حصلت فى commit قديم، فابالتالى ما بيكسرش الـ history بتاع المشروع.

إمتى أستخدم git revert
لما تكون:
- إشتغلت commit وطلع فيه غلط
- والمشروع إتشارك بالفعل (على Git Hub مثلاً)
- وعايز ترجع اللى حصل، بس من غير ما تمسح أو تغير فى تاريخ المشروع

مثال عملى
- تخيل عندك التاريخ ده A --- B --- C --- D ← Head
- إنت اكتشفت إن commit C فيه مشكلة وعايز ترجعه
- هتكتب git revert C
- اللى هيحصل إن Git هيعمل commit جديد تسميه مثلاً E يعكس تغيرات C
A --- B --- C --- D --- E
- كده إنت رجعت التغييرات، بس من غير ما تمسح أو تلخبط الـ history

الأمر git Log --oneLine
عشان تشوف قائمة الـ commits وتختار الـ commit اللى عايز ترجعه

الموضوع: _____ التاريخ: / /

Git switch

هو أمر في Git يستخدم للتبديل بين الفروع أو الإلتزامات المختلفة في مستودع Git الخاص بك. وهذا الأمر يقوم بوظيفة كلا من git branch و git checkout معا. عند تشغيل هذا الأمر مع إسم الفرع أو معرف الإلتزام (commit) يتم التحويل إلى هذا الفرع أو الإلتزام ويصبح الفرع أو الإلتزام النشط الذي يتم العمل عليه.

للتبديل إلى فرع أو إلتزام معين، يمكن إستخدام الأمر git switch متبوعاً بإسم الفرع

```
git switch -c new-feature
```

الدمج (Merge)

```
git branch -d branch name
```

يستخدم أمر "git branch-d branch name" لحذف فرع Git من مستودع Git.

لذلك عند إستخدام الأمر "git branch-d branch name" ستقوم Git بحذف الفرع المحدد (branch name) من مستودع Git الحالي وتحتاج إلى التأكد من أنك تستخدم هذا الأمر بحذر لأنه إذا قمت بحذف الفرع الذي تعمل عليه دون قصد فقد تفقد العمل الذي تم إنجازه في ذلك الفرع.

git merge

يتم إستخدام الأمر "git merge" لدمج فرع Git محدد بفرع آخر في مستودع Git. ويسمح لك الدمج بإضافة التغييرات من فرع لآخر.

الموضوع: _____ التاريخ: / /

عندما تقوم بدمج فرعين في Git باستخدام الأمر "git merge" يتم إنشاء commit جديد يحتوي على جميع التغييرات من الفرع الأول والفرع الثاني.

يتم إستخدام الأمر "git merge" بالشكل التالي

```
git merge branchname
```

حيث أن "branch name" هو الفرع الذي تريد دمجه بالفرع الحالي. ومن المهم التأكد من التحديث (الإنتقال) إلى الفرع الرئيسي قبل استخدام هذا الأمر، وذلك باستخدام الأمر

```
git Checkout main-branch
```

Fast-Forward merge

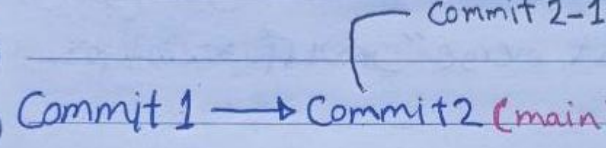
يشير المصطلح إلى نوع من الدمج في Git يتم فيه دمج فرع branch إلى الفرع الرئيسي في حالة عدم وجود تغييرات في الـ main. وبمعنى أخر يتم دمج التغييرات في الفرع الفرعي مباشرة مع الفرع الرئيسي دون إنشاء نقطة تفرع جديدة. يتم إستخدام هذا النوع من الدمج عندما يتم العمل على فرع فرعي بشكل منفرد وبدون أي تداخلات مع الفرع الرئيسي

The 3-way merge

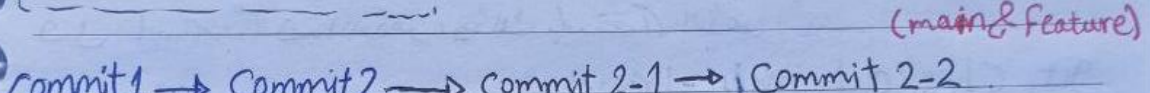
هو نوع آخر من دمج الفروع في Git ويتم استخدامه في حالة وجود تغييرات في الـ main. في هذه الحالة يقوم Git بإنشاء نقطة تفرع جديدة (merge commit) ويحاول دمج التغييرات من الفرعين

تخيل بصري لـ الـ Fast-forward merge

before merge

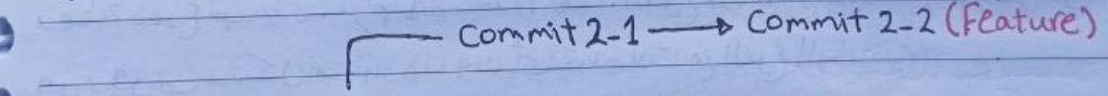


Fast-forward merge

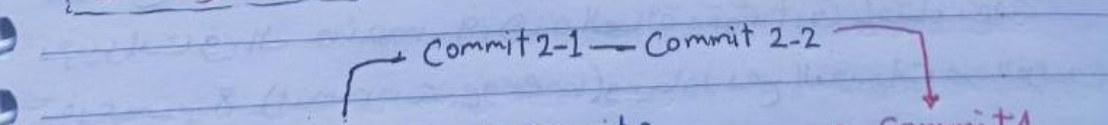


تخيل بصري لـ الـ 3-way merge

before merge



3-way merge

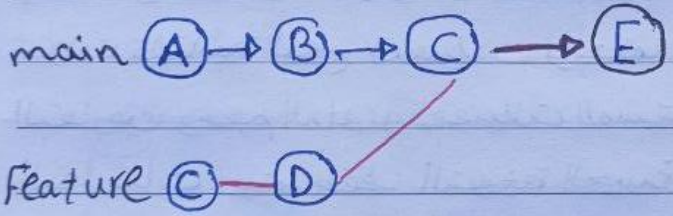


الموضوع: _____ التاريخ: / /

squash Merge

- يعد squash merge أحد الطرق التي يتم بها دمج الفروع في Git يستخدم هذا الأسلوب عندما تريد دمج التعديلات التي تم إجراؤها في فرع معين إلى فرع آخر.

- يعد squash Merge طريقة جيدة لتقليل عدد الإلتزامات commits الزائدة في تاريخ History مشروع Git وللمحافظة على التاريخ الخاص بالمشروع أنيقاً ومنظماً. ومن المهم ملاحظة أنه إذا كنت تستخدم squash merge في Git فيجب عليك التأكد من الحفاظ على تاريخ المشروع مرتباً ومنظماً، وذلك بإستخدام رسائل الإلتزامات الصحيحة ووصف دقيق للتعديلات المدمجة.



الـ E أخدت معلومات من الـ C ومن الـ D بس ماشية في نفس الـ Branch بتاع الـ main، وهو ده معنى كلمة إسكواش = يجمع

إمتى تستخدم squash ؟

- لو إشتغلت على فرع feature وعملت 10 كوميتات صغيرة (تجريب، تصحيح، تعيل ....)
- وعايز تدمج في main بدون ما تبهدل التاريخ وقتها squash بيساعدك تخلي كل الشغل ده في كوميت واحد.

الموضوع: _____ التاريخ: / /

Merge conflits

- يعد Merge conflits أو دمج التعارضات جزءًا من عملية الدمج (merge) في Git
- وتحدث عملية Merge conflits عندما يتم تعديل نفس الملف أو الجزء في فرعين مختلفين بطريقة مختلفة من قبل أكثر من مستخدم في نفس الوقت وهذا يؤدي إلى وجود تعارضات بين التعديلات المختلفة التي يجب حلها.
- عند حدوث التعارضات يقوم Git بإبلاغ المستخدم بالتعارضات ويطلب منه إجراء اللازم لحل هذه التعارضات
- يمكن حل التعارضات بطرق مختلفة مثل دمج التعديلات المختلفة بشكل يدوي أو اختيار إحدى التعديلات على حساب الأخرى أو استخدام أدوات تلقائية لدمج التعديلات ويعتمد الخيار المناسب على طبيعة التعارض وحجم الملف وتفضيلات المستخدم. وعند الإنتهاء من حل التعارضات، يتم حفظ النسخة الجديدة

تبسيط للفكرة :-

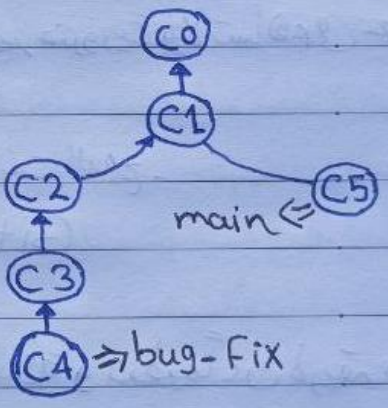
يعني لو عندي فرع رئيسي main وعندي تفرع إسمه First branch وعندي تفرع تاني إسمه second branch وجيت في السطر التاني مثلاً وعملت تغيير وعملت commit في التفرع الأول وكمان في التفرع التاني ثم روحت عملت دمج merge للتفرع الأول ودمجته مع الفرع الرئيسي main وقتها هيحصل Forward merge ومش هيحصل مشاكل. لكن لما تيجي تدمج التفرع التاني second branch مع الفرع الرئيسي main محرر الـ VS code هيعملك إيرور ويظهرلك في الـ UI إنك لازم تختار ما بين التغييرين

تمج أنهى واحد فيهم أو تدمجهم الإتنين مع بعض وده معناه إن
في الحالات اللى زى دى محرر الأكواد فيجوال إستوديو كود بيرجعلك
عشان تقرر إنه يعمل إيه في حالة التعارض دى.

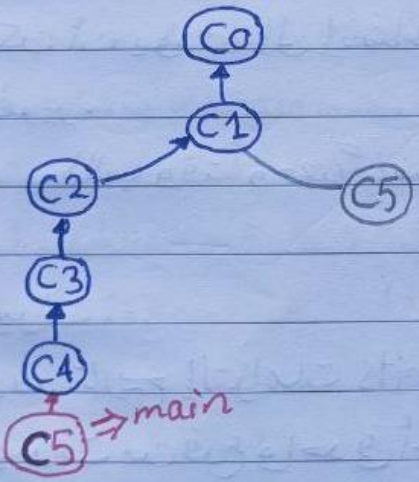
Git Rebase

هو أمر بنستخدمه علشان ننقل التعديلات من فرع على فرع تانى،
بس بشكل مرتب ونضيف في التاريخ.
يعنى بدل ماندمج فرعين ونسيب أثر "الدمج" بنخلى شكل الكوميتات
كأنها كانت على الفرع الأصلى من البداية

قبل الـ Rebase



بعد الـ Rebase



دلوقتى أنا عايز أدمج فرع bug-Fix مع الفرع الرئيسى main يعنى
هدمج C5 مع C4 فهكتب الأمر git rbase bug-Fix اللى
بيحصل إن git هيشيل C5 وياخدها يحطها فوق C4 ويضيف ليها
التغييرات وده هيحصل برضو لو عندك C5 و C6 هيشيلهم ويحطهم
فوق كل الـ commits ويبقا عندك الفرع كله فى Line واحد

الموضوع: _____ التاريخ: / /

مقارنة بين مميزات وعيوب Git rebase و Git merge

Git Merge

المميزات :-

- يحافظ على تاريخ المشروع الخطي (Linear)، مما يسهل تتبع التغييرات وفهم عملية التطوير.
- لا يقوم بتغيير الفرع (Branch) الأصلي أو تاريخ الإلتزامات Commits الأصلي.
- هو طريقة بسيطة ومباشرة لدمج التغييرات من فرع واحد إلى آخر.

العيوب :-

- يمكن أن يؤدي إلى تاريخ التزامات Commits فوضوي ومربك عند دمج فروع متعددة
- يمكن أن يؤدي إلى إنشاء علامات دمج غير ضرورية ولا تساهم في عملية التطوير.
- قد يتطلب وقتاً وجهداً إضافياً لحل تضاربات الدمج.

Git Rebase

المميزات :-

- يوفر تاريخ إلتزامات Commits أكثر نظافة وتنظيماً عن طريقة دمج التغييرات من فرع واحد في أخر بدون إنشاء علامات دمج غير ضرورية.
- يتيح عملية تطوير أكثر تيسيراً وخطية.
- يقلل من عدد تضاربات الدمج التي يتعين حلها، حيث يتم دمج التغييرات في تاريخ خطي

العيوب :-

- يقوم بتغيير تاريخ الإلتزام Commit الأصلي، مما يمكن أن يجعل من الصعب تتبع التغييرات وفهم عملية التطوير.
- يمكن أن يؤدي إلى تاريخ أكثر تعقيداً، خاصة عند إعادة ترتيب فروع كبيرة أو طويلة الأمد.

الموضوع: _____ التاريخ: / /

• يمكن أن يؤدي إلى مشكلات في الفروع المشتركة، حيث يتم إعادة كتابة التاريخ ويمكن أن يسبب تضاربات للمطورين الآخرين.
• بشكل عام، git merge هو خيار جيد عندما تريد الحفاظ على تاريخ المشروع بشكل واضح ومباشر.

(*) نقطة مهمة بين git switch و git checkout
في إصدار Git 2.23 أضافو أمر جديد إسمه git switch هدفه إنه يستبدل الأمر القديم git checkout لأن git checkout "محشو زيادة" (يعني بيعمل وظائف كتير جداً حسب السياق) وده ممكن يسبب لخبطة للمستخدم

الفرق بين git checkout و git switch
git checkout ده أمر قديم وكبير في Git بيعمل حاجات كتير:-
1- تتنقل لفرع معين — git checkout main
2- تعمل فرع جديد وتتنقل له — git checkout -b feature
3- ترجع ملف لأخر نسخة — git checkout -- index.html

git switch في إصدار Git 2.23 قرروا يفصلو الموضوع علشان يسهلوا الفهم:
- لو عايز تتنقل بين الفروع إستخدم git switch
- لو عايز تعمل تغييرات على الملفات إستخدم git restore

* إستخدام git switch
1- التبديل لفرع موجود — git switch main
2- إنشاء فرع جديد والتبديل له — git switch -c feature

الموضوع: _____ التاريخ: / /

العمل مع الريبو عن بعد working with remote Repo

تحدثنا سابقاً على أن Git hub يقوم بتوفير بيئة عمل مرنة يمكن للمطورين إستخدامها للتحكم في إدارة مشاريعهم، ومشاركة الأعمال و التعاون مع الأخرين في مشاريع مشتركة، وإدارة الإصدارات والتعديلات على الكود المصدري وإدارة الأخطاء و العيوب وتتبع الأعمال.

الكود الذي تقوم بكتابته يكون على local لا أحد يمكنه رؤيته غيرك ولكن أحياناً تريد مشاركته مع الأخرين وهنا يأتي دور Git hub الذي يجعلك تقوم بإنشاء مستودع Repository عليه وتحميل الكود المحلي إليه، ويمكنك مشاركة المستودع Repo مع الأخرين عن طريق إرسال رابط المستودع Repo لهم. وبمجرد أن يتمتع أي شخص بصلاحيات الوصول إلى المستودع، يمكنه القيام بعمليات التحرير و التعديل والتحديثات على الكود المصدري وإرسالها مرة أخرى للمستودع Repo

إتبع الخطوات التالية لرفع مشروعك على Git hub

```
git init
git add Readme.md
git commit -m "first commit"
git branch -m main
git remote add origin https://github.com/userName/RepoName.git
git push -u origin main
```

git init

هذا الأمر يبدأ مشروع Git جديد. ينشئ Git مجلداً جديداً بإسم ".git" والذي يحتوي على جميع الأدوات التي يحتاجها Git لتتبع التغييرات في المشروع

git add Readme.md

هذا الأمر يقوم بإضافة ملف Readme.md

git commit -m "first commit"

يستخدم هذا الأمر لتأكيد التغييرات في مشروع Git. يتم تضمين الملفات التي تم إضافتها إلى منطقة الإنتظار في عملية التأكيد (commit)، ويتم وصف التغييرات في رسالة التأكيد.

git branch -m main

يقوم هذا الأمر بإعادة تسمية الفرع الإفتراضي الخاص بالمستودع من "master" إلى "main"

git remote add origin https://github.com/userName/repoName.git

يستخدم هذا الأمر لإضافة الخادم البعيد (remoterepo) الذي تم ربط المشروع به إلى Git.

يتم إستخدام "origin" كـ الخادم البعيد (remote repo) في هذا المثال.

git push -u origin main

يقوم هذا الأمر بإرسال التغييرات المؤكدة إلى الخادم البعيد remote Repository

أي أنه يرفع المشروع لـ Remote لأول مرة

الموضوع: _____ التاريخ: / /

Git Fetch و Git Pull كلاهما يُستخدمان لجلب تحديثات من مستودع Git البعيد (remote repository) إلى مستودع Git المحلي (local repo)، ولكن هناك اختلافات بينهما.

Git Fetch

يستخدم لجلب التحديثات الجديدة من المستودع البعيد Remote Repo إلى المستودع المحلي Local repo، لكنه لا يقوم بدمج هذه التحديثات مع النسخة المحلية، ولذلك يجب أن تقوم بتنفيذ الأمر git merge

Git Pull

يستخدم لجلب التحديثات الجديدة من المستودع البعيد Remote Repo ودمجها مع النسخة المحلية. يقوم git Pull بالقيام بنفس العمل الذي يقوم به Git Fetch، ولكن يضيف خطوة إضافية وهي تنفيذ الأمر git merge لدمج التحديثات الجديدة مع النسخة المحلية بشكل تلقائي

يمكن إستخدام Git Fetch عندما تريد فقط الحصول على تحديثات جديدة دون دمجها مع النسخة المحلية، بينما يمكن إستخدام Git Pull عندما تريد جلب التحديثات الجديدة ودمجها مع النسخة المحلية

يعني إيه private Repository في Github ؟

هو مستودع مخفي يعني :-

- محدش يقدر يشوفه غيرك إنت وصاحب الصلاحيات (اللي بتديهم access)
- مش ظاهر لأي زائر يدخل على صفحتك في Git Hub
- حتى لو حد عنده لينك للريبو مش هيقدر يفتحه إلا لو أنت أضفته بنفسك كـ collaborator

الموضوع: _____ التاريخ: / /

الفرق بين Private و Puplic

العنصر	Puplic Repo	Private Repo
الرؤية	الكل يقدر يشوفه	انت والأشخاص المصرح لهم
مناسب لـ	مشاريع مفتوحة المصدر open source	مشاريع خاصة أو لسة بتشتغل عليها
الظهور في البروفايل	ظاهر ضمن المشاريع	مش ظاهر إلا ليك

إمتى تستخدم Private Repository ؟

- لما تكون بتطور مشروع ولسة مش جاهز للنشر.
- لو بتشتغل على فكرة startup وعايز تحتفظ بيها لنفسك.
- بتجهز شغل Freelance أو تدريبات خاصة.
- بتبني تطبيقه ولسة مش حابب الناس تشوف الكود

إزاى أشارك الريبو الخاص مع حد ؟

1- ادخل على الريبو
2- روح على Collaborators > settings
3- وتضيف أى username على GitHub علشان تشاركه المشروع

مين يقدر يعمل Private Repos ؟

- لو عندك حساب GitHub عادى (Free) تقدر تعمل Private repos (بحدود كويسة)
- ولو مشترك فى Git Hub Pro أو طالب، عندك مزايا كتير.

الموضوع: _____ التاريخ: / /

Fork

يعني إنشاء نسخة من مستودع Git الحالي وتخزينة في حساب Git آخر. عادة ما يتم إستخدام هذا الأمر عند الرغبة في إضافة تحسينات أو تعديلات لمشروع Git موجود على الإنترنت، حيث يتم إنشاء نسخة من المستودع الأصلي وتعديلها دون التأثير على المستودع الأصلي.

يمكن لأي شخص إنشاء نسخة من المستودع وإجراء التعديلات التي يريدها وإرسالها إلى المستودع الأصلي للمشروع من خلال إرسال طلب دمج Pull Requst

Clone

يعني إنشاء نسخة من مستودع Git الحالي وتخزينها على جهاز الكمبيوتر الخاص بك لتتمكن من العمل عليها بشكل محلي.

يتم إستخدام هذا الأمر عادة عند الرغبة في الحصول على نسخة من مستودع Git موجود على الإنترنت.

يمكن إستخدام الأمر Clone عندما ترغب في الحصول على إصدار محدث من المستودع بعد إجراء تغييرات من قبل فريق التطوير

useful plugins <<<

1. Git History
2. Git Graph
3. ★ Git Lens

تم بحمد الله

في النهاية يارب يكون الملف فادك ولو
بمعلومة صغيرة أسال الله أن ينفعك به
ويجعل النفع فينا وفيك
وبعتذرلك لو معرفتش تقرأ حاجة من خطي
 فلو رشحتلي كورسات تحسين خط أكون
 شاكر ليك
أو اكتبلي في الكومنتات ممكن اعمل
الملخص ده ديجتال مين عارف 

 والسلام عليكم ورحمة الله وبركاتة