Start What's git

نظام Git

هو نظام يساعد على الاحتفاظ بتسلسل التغييرات التي تطرأ على مشروع ما خلال فترة زمنية معينة. أفضل طريقة للتعرف على Git هي بإستخدام Git نفسه. دعنا نتخيل أنك بيتر كريستين اسبيورنسن في العام ١٨٤١ بالنرويج تجلس على طاولة مكتبك لتكتب قصة للأطفال.

تقوم بفتح جهاز الحاسب الخاص بك وتقوم بإنشاء مجلد جديد، ثم تقوم بإنشاء ملف لتكتب فيه قصتك. هل كان بيتر يملك حاسوبا في ١٨٤١؟ بالطبع فهذا من وحي خيالنا فما المانع في ذلك؟ المهم انك تضع عنوانا للقصة: الجديان الثلاثة.

ولأنك كاتب متمرس، فإن أول ما تود أن تفعله هو ان تفتح برنامج Terminal وتكتب هذا الأمر حتى تسمح ل Git من متابعة التغييرات التي ستقوم بها في هذا الملف:

|  |
| --- |
| git init |

يعتمد Git على ثلاثة بيئات للعمل:

1. Working Directory: المكان الذي تقوم فيه بإنشاء وحذف وتعديل ملفات المشروع.
2. Staging Area: المكان الذي نقوم فيه بإدراج فيه الملفات التي نرغب ل Git أن يحفظها.
3. Repository: المكان الذي يستخدم للتخزين النهائي للملفات.

الآن وبعد أن قمت بتهيئة المشروع للعمل مع Git، فقد تود الاطلاع على الوضع الحالي للملفات لتتمكن من رؤية جميع الملفات تحت إدارة Git وهل تمت إضافتها لل Staging area أم لا تزال في الـ working area:

|  |
| --- |
| git status |

ثم تقوم بإضافة الملفات لل Staging area تمهيدا لحفظها نهائيا:

|  |
| --- |
| git add fileName |

ولكن قبل الحفظ النهائي قد تود في رؤية جميع التغييرات في الملف:

|  |
| --- |
| git diff |

اخيرا انت راض عن عنوان القصة فتقوم بحفظ هذه التغييرات مع إضافة رسالة ليسهل عليك التعرف على كل تغيير قمت به من دون الدخول للملف:

|  |
| --- |
| git commit -m “Add story title” |

بيتر (انت بيتر، الكاتب تذكر!)، لقد كتبت قصتك الآن على عدة مراحل متبعا نفس النظام السابق:

* تعديل الملف
* إضافته لـ staging area
* ثم حفظه في ال repo

يمكنك الاطلاع على كل التعديلات التي قمت بها الآن (ومن على [Github](https://github.com/bashmoanas/TheThreeGoats/commits/main/TheThreeGoats.md)):

|  |
| --- |
| git log |

لقد كتبت قصة جميلة حقا ولكنك لست متأكدا من النهاية التي كتبتها بشكل كاف، تفكر انه يجب ان تكون طريقة ما لكتابة نهاية مختلفة دون المساس بالنهاية التي كتبتها مسبقا. بالطبع أنت محق وإلا فلم نحن هنا. يمكنك من خلال Git ان تقوم بعمل branch (فرع) جديد من قصتك تكتب فيه النهاية الجديدة. (ومن على [Github](https://github.com/bashmoanas/TheThreeGoats/commits/anotherVersion)):

|  |
| --- |
| git branch branchName // Create new branch  git checkout branch // Go to the new branch |

لقد كنت محقا فالنهاية الثانية تعجبك اكثر وتقرر الاحتفاظ بها. حينها يمكنك التخلص من ال branch الذي قمت بإنشائه سابقا فقد أتم مهمته على أكمل وجه.

|  |
| --- |
| git checkout main // Go to the main branch  git merge branchName // merge the new End to main  git branch -d branchName // Delete the new branch |

بيتر، لقد كتبت كتابا رائعا ولكن لا يمكن عمل كل شيء بمفردك. تقوم باستدعاء المراجع اللغوي ودار النشر لمساعدتك على إنهاء الكتاب. بإمكانهم عمل نسخة من ال repo على جهازهم الخاص للعمل عليها دون المساس بنسختك.

|  |
| --- |
| git clone repoLocation cloneNameOfChoice //clone repo |

بهذه الطريقة يمكن لكل شخص العمل على نسخته الخاصة بشكل منفصل عن النسخة المشتركة بينكم جميعا والتي يتم تسميتها عادة origin. عادة تكون طريقة العمل كالتالي:

* يقوم الشخص بعمل fetch لاية تعديلات من ال origin
* ثم يقوم بعمل دمج merge لهذه التعديلات في نسخته الخاصة على الحاسب
* ثم يقوم بعمل branch جديد لعمل التعديلات التي يريدها
* تجنبا أن يكون قد تم تحديثات على النسخة المشتركة، يقوم بعمل fetch مرة اخرى وmerge.
* في النهاية يقوم بإضافة تعديلاته على النسخة المشتركة من خلال push.

|  |
| --- |
| git fetch  git merge origin/branchName  git push branchName origin |

حسنا يمكن لبيتر أن يرتاح الآن. إذا كنت ترغب في معرفة المزيد عن قصة الجديان الثلاثة برجاء قراءة هذه السلسلة من [التغريدات](https://mobile.twitter.com/labanshy/status/1424099093488586761?lang=ar-x-fm). يمكنك الإطلاع على المشروع على [Github](https://github.com/bashmoanas/TheThreeGoats).

قد تتسائل الآن، ولكنني لست كاتبا نرويجيا يكتب قصص أطفال منذ قرون مضت، انا مبرمج، كيف يمكن ل Git أن يساعدني.

إذا كنت تعمل بمفردك، فبإمكانك استخدام ال branching في Git لتطوير خاصية جديدة دون المساس بالكود المكتوب سلفا والذي يعمل بالفعل. في هذه الحالة سوف تتمكن من التركيز على الخاصية التي تود إضافتها وتستطيع ان تقوم بتجربتها قبل إضافتها للكود الذي كتبته بالفعل.

اما اذا كنت تعمل مع فريق عمل، ف Git لا غنى عنه. فبإستخدام ال branching يمكن لكل فرد في الفريق أن يساهم في المشروع دون حدوث مشاكل في الكود الذي تم كتابته مسبقا.

كما يتم إستخدام Github\* بحيث يحتوي على النسخة المشتركة من ال repository ويكون لدى كل فرد نسخته الخاصة ليعمل عليها دون الحاجة حتى لوجود الانترنت.

يسمح Github بعمل Pull request وذلك حتى تتمكنوا كفريق من مناقشة التعديلات التي تمت قبل إضافتها للنسخة المشتركة بينكم.

اذا لم تستعمل Git من قبل فقد يبدو لك معقدا. وهذه صحيح لذا ابدأ بإستخدامه فورا حتى تتأقلم عليه سريعا 🤷🏻‍♂️.

* مواقع Github او GitBucket او GitLab كلها أدوات ل Git ذات واجهة استخدام.

end What's git

start gitflow

End gitflow

Start better way git

End better way git

Start git practices for teams

end git practices for teams

start rebase vs merge

end rebase vs merge