



git

+ GitHub

اعداد:

م. أحمد حاج محمد سلامة الجامعة الدولية للعلوم والنهضة



ماذا سوف نتعلم في هذا المقرر







GitIgnore ملف التجاهل



Git Checkout السفر في الزمن



Reset الأمر









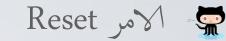


لاشك أنه ستكون أجزاء أو ملفات من المشروع الذي تعمل عليه لا تريد مشاركتها مع غيرك ممن يطلع على شيفرة المشروع، و من أمثلة تلك الملفات:

- ملفات السجلات log files
- الملفات المؤقتة temporary files
 - الملفات المخفية.
 - الملفات الشخصية.
- ملفات الكلمات السرية و مفاتيح المصادقة.
 - غير ذلك من الملفات ذات الحساسية.

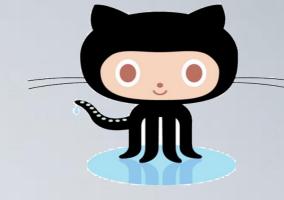








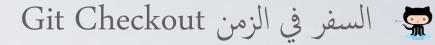


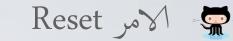


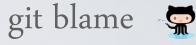
تستطيع أن تطلب من git أن يتجاهل تلك الملفات أو الأجزاء من المشروع التي ينبغي له تجاهلها باستخدام ملف gitignore. لاحظ أن اسم الملف يبدأ بنقطة وبالتالي سيكون مخفياً بشكل افتراضي في أنظمة لينكس و ماك أو إس) و لن يتعقب git الملفات و المجلدات المحددة في gitignore. رغم أنه يتتبع ملف gitignore. نفسه.

يجب أن يتم إنشاء الملف gitignore في أعلى مستوى من مجلدات المشروع (في المجلد الجذر root بتعبير آخر)، ثم و بعد إنشاء الملف نستخدم أي محرر ، نصي، و نضيف القواعد التي تحدد الملفات التي نرغب بتجاهلها، و حين نقول قواعد فنحن نعني ذلك لأننا قد نكون في حالات نحتاج فيها لتجاهل الكثير من الملفات دفعة واحدة كتلك التي لها امتداد محدد مثلاً أو التي تطابق أسهاؤها نمطاً معيناً ... إلخ















فيا يلي القواعد التي نستطيع تعريفها:

Pattern	Explanation/Matches	Examples
	Blank lines are ignored	
# text comment	Lines starting with # are ignored	
name	All name files, name folders, and files and folders in any name folder	/name.log /name/file.txt /lib/name.log
name/	Ending with / specifies the pattern is for a folder. Matches all files and folders in any name folder	/name/file.txt /name/log/name.log
		no match: /name.log
name.file	All files with the <i>name.file</i>	/name.file /lib/name.file

ملف التجاهل GitIgnore	
السفر في الزمن Git Checkout	
Reset الأمر	
استخدام Commit –amend	
git blame	





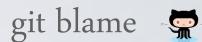
/name.file	Starting with / specifies the pattern matches only files in the root folder	/name.file
		no match: /lib/name.file
lib/name.file	Patterns specifiing files in specific folders are always realative to root (even if you do not start with /)	/lib/name.file no match: name.file /test/lib/name.file
**/lib/name.file	Starting with ** before / specifies that it matches any folder in the repository. Not just on root.	/lib/name.file /test/lib/name.file
**/name	All name folders, and files and folders in any name folder	/name/log.file /lib/name/log.file /name/lib/log.file















name?.file	? matches a single non-specific character	/names.file
		/name1.file
		no match:
		/names1.file
name[a-z].file	[range] matches a single character in the specified range (in this case a character in the range of a-z, and also be numberic.)	/names.file /nameb.file
		no match:
		/name1.file
name[abc].file	[set] matches a single character in the specified set of characters (in this case either a, b, or c)	/namea.file
		/nameb.file
		no match:
		/names.file
name[!abc].file	[!set] matches a single character, except the ones spesified in the set of characters (in this	/names.file /namex.file
	case a, b, or c)	/namex.me
		no match:
		/namesb.file
*.file	All files withe .file extention	/name.file /lib/name.file
		/IID/IIailie.ilie
name/ !name/secret.log	! specifies a negation or exception. Matches all files and folders in any <i>name</i> folder, except name/secret.log	/name/file.txt /name/log/name.log
		no match:
		/name/secret.log

















*.file !name.file	! specifies a negation or exception. All files withe .file extention, except name.file	/log.file /lastname.file
		no match: /name.file
.file !name/.file junk.*	Adding new patterns after a negation will re-ignore a previous negated file All files withe .file extention, except the ones in name folder. Unless the file name is junk	/log.file /name/log.file
		no match: /name/junk.file





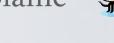
Git Checkout السفر في الزمن



Commit –amend استخدام



git blame









نستطيع باستخدام git أن نعيد عقارب الساعة إلى الوراء مجازياً، بمعنى أن نعيد الملفات الموجودة في المستودع إلى الحالة التي كانت فيها لحظة مساهمة ما تماماً، ذلك من خلال الأمر git و checkout ثم ذكر معرّف المساهمة المطلوب العودة إليها.

لقد أنشأت ملف html وملف js باستخدام الامر

<اسم الملف> Touch

XPRISTO@DESKTOP-STN36A1 MINGW64 ~/Desktop/test (master)

\$ touch index.html

XPRISTO@DESKTOP-STN36A1 MINGW64 ~/Desktop/test (master)
\$ touch main.js

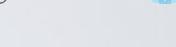
GitIgnore ملف التجاهل



Reset الأمر



git blame 🦃







```
بعد تتبع الملفات السابقة و عمل الإيداع

$ git add *

XPRISTO@DESKTOP-STN36A1 MINGW64 ~/Desktop/test (master)

$ git commit -m "one-v1"

[master (root-commit) 0102ba9] one-v1

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 index.html
```

استعراض عمليات الإيداع

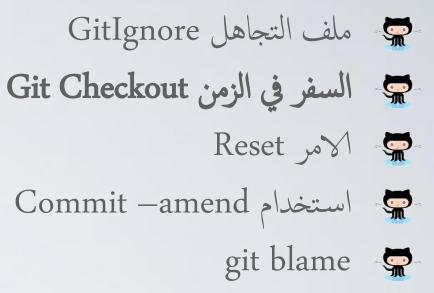
```
Git Checkout السفر في الزمن Git Checkout السفر في الزمن Reset الامر Commit –amend استخدام git blame
```





للرجوع الى الوراء بالزمن عند عملية إيداع

```
PRISTO@DESKTOP-STN36A1 MINGW64 ~/Desktop/test (master)
 git checkout 0102ba9151dc7884682b6be3f3cf80c1ccda21f2
Note: switching to '0102ba9151dc7884682b6be3f3cf80c1ccda21f2'.
You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.
If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:
  git switch -c <new-branch-name>
Or undo this operation with:
 git switch -
Turn off this advice by setting config variable advice.detachedHead to false
HEAD is now at 0102ba9 one-v1
(PRISTO@DESKTOP-STN36A1 MINGW64 ~/Desktop/test ((0102ba9...))
commit 0102ba9151dc7884682b6be3f3cf80c1ccda21f2 (HEAD)
Author: ahmadSalameh1995 <ahmadsalama998877@gmail.com>
Date:
       Fri Aug 25 16:05:29 2023 +0300
    one-v1
 PRISTO@DESKTOP-STN36A1 MINGW64 ~/Desktop/test ((0102ba9...))
```







ذكرنا في الفصل الأول أنه توجد طريقة ثانية متقدّمة للتراجع عن مساهمة أو أكثر قمنا بها، وقلنا أنّنا سنغطيها في الفصل المتقدّم، و الآن قد حان الوقت للحديث عنها، ألا و هي طريقة الأمر reset يُستخدم الأمر reset عند الرغبة في إرجاع المستودع في التاريخ إلى مساهمة سابقة و له ثلاثة أنماط

• النمط الخشن hard—: هو أخطر الأنماط حيث أنّه يعيد المستودع إلى مساهمة سابقة مع إلغاء جميع المساهمات و التغييرات التي حدثت بعد تلك المساهمة.

git reset --hard <commit-id>

• النمط الناعم soft -: هو شبيه بالسابق إلا أنّه لا يلغي المساهمات التالية، بل يبقيها و كأنّها متت و تم عمل add لها إنّها لم يتم عمل commit و لهذا اسمه سوفت).

git reset --soft <commit-id>













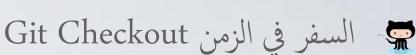


النمط الوسطي mixed شبيه بالنمط الناعم مع فارق أنّه يبقي التغييرات وكأنها حدثت للتو و لم تسجل في git commit ثم git add تاريخ المستودع بعد، و بالتالي إذا أردت الاحتفاظ بها فيجب أن تقوم بعمل git add ثم git commit أ

git reset -mixed <commit-id>

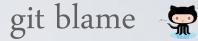
بغضّ النظر عن النمط الذي تريد استخدامه فيجب أولاً أن تقوم بتحديد المساهمة التي ترغب في العودة gitk النظر عن خلال تحديد معرّفها Commit ID من خلال الامر git log او الامر gitk





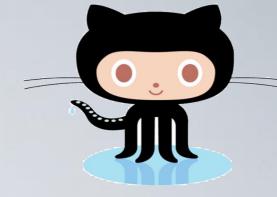












تخيل أنك قمت بعمل commit و من ثم نسيت إضافة أحد الملفات لها، أو قمت بعمل commit أخطأت في كتابة الرسالة وأردت أن تعيد صياغتها مثلاً أو لأي سبب ما، أردت أن تتراجع عن الأمر، عندها نقول أنه في حالة أردت المحاولة مرة أخرى، فقم بتنفيذ commit مع الخيار amend – كالتالي

git commit -amend

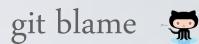
بمثال بسيط، تخيل أنك قمت بعمل commit و بعد ذلك وجدت أنك نسيت أحد الملفات، و لنقل file.cpp و أردت أن تضيفه، عندها يمكننا القيام بالتالي بعد commit التي قمنا بها

git commit -m 'initial commit'
git add file.cpp
git commit --amend















أحياناً نرغب و لسبب ما بمعرفة من هو المساهم الذي تسبب بتغيير سطر ما من ملف ما أو مجموعة أسطر، يتيح لنا git blame ذلك من خلال الأمر git blame و أرى تسميته طريفة لأن blame باللغة الإنجليزية تستعمل في الحالات السلبية لإلقاء اللوم على أحد ما . صيغة الأمر هي:

git blame -L <start>,+<end> -- <file-path>

```
XPRISTO@DESKTOP-STN36A1 MINGW64 ~/Desktop/test (master)
$ git blame -L 2,+4 -- asd.js
3beed2c1 (ahmadSalameh1995 2023-08-25 18:11:36 +0300 2) var t=6-5;
3beed2c1 (ahmadSalameh1995 2023-08-25 18:11:36 +0300 3) var x=5*5;
3beed2c1 (ahmadSalameh1995 2023-08-25 18:11:36 +0300 4) var t=6-5;
3beed2c1 (ahmadSalameh1995 2023-08-25 18:11:36 +0300 5) var x=5*5;
```

GitIgnore ملف التجاهل

