



# ورقة مساعدة للعمل على GIT

مقدمة من TOWER: حلول متكاملة لإدارة الإصدارات باستخدام Git

## الدمج وإعادة التأصيل (REBASE)

دمج فرع محدد مع الفرع الحالي

```
$ git merge <branch>
```

إعادة تأصيل الفرع الحالي من فرع محدد  
لا تقم بإعادة تأصيل الإيداعات المنشورة

```
$ git rebase <branch>
```

إلغاء إعادة التأصيل

```
$ git rebase --abort
```

متابعة إعادة التأصيل بعد معالجة التعارضات

```
$ git rebase --continue
```

استخدام الأداة المحددة ضمن الإعدادات لمعالجة التعارضات

```
$ git mergetool
```

استخدام المحرر لحل التعارضات بشكل يدوي  
وتحديد الملف على أنه خال من التعارضات  
(بعد إتمام معالجة التعارضات)

```
$ git add <resolved-file>
```

```
$ git rm <resolved-file>
```

## التراجع

تجاهل كافة التعديلات المحلية ضمن دليل العمل

```
$ git reset --hard HEAD
```

تجاهل التعديلات المحلية لملف محدد

```
$ git checkout HEAD <file>
```

التراجع عن إيداع (عبر إنشاء إيداع معاكس)

```
$ git revert <commit>
```

نقل رأس القراءة إلى إيداع سابق

... وإهمال التعديلات التي تلت ذلك الإيداع

```
$ git reset --hard <commit>
```

... والاحتفاظ بالتعديلات التي تلت ذلك الإيداع على أنها تعديلات غير مضافة ضمن منطقة الإدراج

```
$ git reset <commit>
```

... والاحتفاظ بالتعديلات المحلية التي لم يتم إيداعها

```
$ git reset --keep <commit>
```

## التفريعات والوسوم

عرض جميع الفروع المتوفرة

```
$ git branch -av
```

نقل رأس القراءة إلى الفرع المحدد للعمل عليه

```
$ git checkout <branch>
```

إنشاء فرع جديد انطلاقاً من الحالي

```
$ git branch <new-branch>
```

إنشاء فرع مرتبط بفرع على مستودع بعيد

```
$ git checkout --track <remote/branch>
```

حذف فرع محلي

```
$ git branch -d <branch>
```

تعليم الإيداع الحالي بوسم

```
$ git tag <tag-name>
```

## التحديث والنشر

عرض جميع المستودعات البعيدة

```
$ git remote -v
```

عرض تفاصيل مستودع بعيد

```
$ git remote show <remote>
```

إضافة مستودع بعيد والربط معه لمتابعة تعديلاته

```
$ git remote add <shortname> <url>
```

تحميل جميع التعديلات من المستودع البعيد دون إدراجها ضمن الفرع الحالي

```
$ git fetch <remote>
```

تحميل جميع التعديلات من المستودع البعيد ودمجها مباشرة ضمن الفرع الحالي

```
$ git pull <remote> <branch>
```

نشر التعديلات المحلية إلى المستودع البعيد

```
$ git push <remote> <branch>
```

حذف فرع من المستودع البعيد

```
$ git branch -dr <remote/branch>
```

نشر الوسوم التي تم إنشاؤها

```
$ git push --tags
```

## إنشاء المستودعات

استنساخ مستودع موجود مسبقاً

```
$ git clone ssh://user@domain.com/repo.git
```

خلق مستودع جديد

```
$ git init
```

## التعديلات المحلية

عرض الملفات المعدلة ضمن مجلد العمل

```
$ git status
```

مقارنة التعديلات ضمن الملفات التي يتم تعقبها

```
$ git diff
```

إضافة كافة التعديلات الحالية إلى الإيداع التالي

```
$ git add .
```

إضافة بعض التعديلات من الملف المحدد إلى الإيداع التالي

```
$ git add -p <file>
```

إيداع جميع التعديلات الحاصلة على الملفات التي يتم تعقبها

```
$ git commit -a
```

إيداع التعديلات التي قام المستخدم بإضافتها إلى منطقة الإدراج

```
$ git commit
```

تعديل الإيداع الأخير

تجنب تعديل الإيداعات التي تم نشرها!

```
$ git commit --amend
```

## سجل الإيداعات

عرض جميع الإيداعات انطلاقاً من الأحدث

```
$ git log
```

عرض التعديلات الطارئة على ملف محدد منذ إضافته للمستودع

```
$ git log -p <file>
```

من قام بتعديل الملف ومتى؟

```
$ git blame <file>
```





# إدارة الإصدارات

نصائح لنتائج أفضل

## وظف ميزة التفرع

يعتبر التفرع من أقوى ميزات Git، وذلك لم يكن بالصدفة: فالتفرع السريع والسهل كان مطلباً أساسياً منذ اليوم الأول. الفروع هي الحل الأمثل لتجنب خلط مسارات التطوير. عليك استعمال الفروع بكثرة خلال سير عملك: للميزات الجديدة، وإصلاح الأخطاء، واختبار الأفكار...

## اتفق مع فريقك على خطة عمل

يقدم لك Git عدة طرق لتسيير أعمال التطوير: الفروع طويلة الأمد، الفروع بحسب الموضوع، الدمج وإعادة التأميل، `git-flow`... اعتماد إحدى طرق تسيير العمل يتم بناء على مجموعة من العوامل: نوع وحجم المشروع، الطرق المعتمدة للتنفيذ ووضع المشاريع في طور الإنتاج، والتفضيلات الشخصية لك ولفريقك (قد يكون هذا العامل الأهم). المهم في الأمر أن يتم اعتماد طريقة عمل يلتزم بها الجميع.

## المساعدة والتوثيق

للحصول على شروحات تفصيلية لأوامر Git يمكنك تنفيذ الأمر التالي في سطر الأوامر

```
$ git help <command>
```

## مراجع مجانية على الإنترنت

<http://www.git-tower.com/learn>

<http://rogerdudler.github.io/git-guide/>

<http://www.git-scm.org/>

## اختبر الشيفرة قبل إيداعها

تجنب إيداع عمل "تظن" أنه مكتمل. اختبره بعناية لتتأكد من أنه مكتمل بالفعل ولا يحمل آثاراً جانبية للأجزاء الأخرى من المشروع. عندما تقوم بإيداع عمل ناقص في مستودعك المحلي فهذا يتطلب منك أن تسامح نفسك فقط، لكن الاختبار قبل الإيداع يصبح مطلباً أساسياً عند دفع/مشاركة عملك مع الآخرين.

## اكتب رسائل إيداع معبرة

ابدأ رسالتك بخلاصة قصيرة حول التعديلات التي قمت بها (حوالي 50 حرفاً). افصلها عن العرض بسطر فارغ. عرض الرسالة يقدم إجابات مفصلة عن الأسئلة التالية:

« ما هو الدافع لإجراء التعديلات؟ »

« ما أوجه الاختلاف بين الإصدار السابق والحالي؟ »

يستحسن أن تصاغ العبارات بصيغة الأمر وليس بصيغة الماضي (على سبيل المثال `change` وليس `changed`) للتوافق مع الرسائل التي تولدها أوامر مثل `git merge`.

## نظام إدارة الإصدارات ليس نظاماً للنسخ الاحتياطي

إن انتقال ملفاتك بشكل تلقائي إلى مستودع بعيد يعتبر تأثير جانبياً لطيفاً لاستخدام أنظمة إدارة الإصدارات، لكن لا يجوز اعتبارها أنظمة نسخ احتياطي.

عند التعامل مع أنظمة إدارة الإصدارات عليك الانتباه لإيداع التعديلات المتعلقة بموضوع واحد، وليس إدراج جميع الملفات التي تخشى ضياعها كما في النسخ الاحتياطي.

## قم بإيداع تعديلات مترابطة

يجب أن تغلف عملية الإيداع مجموعة من التعديلات المتعلقة بموضوع واحد. على سبيل المثال، إصلاح مشكلتين برمجيتين يجب أن ينتج عنه إيداعان منفصلان. إيداع تعديلات قليلة يبشر للمطورين فهمها ويسمح لهم بالتراجع عنها عند الحاجة.

باستخدام أدوات مثل منطقة الإدراج وقابلية إدراج أجزاء منفصلة من ملف واحد، يساعدنا Git بإنشاء إيداعات دقيقة للغاية.

## قم بالإيداع عند أقرب فرصة

الإيداع بتواتر عالٍ يجعل إيداعاتك صغيرة ويساعدك في حصر الإيداع بالتعديلات المترابطة فقط كما ذكرنا سابقاً. بالإضافة إلى أنه يسمح لك بمشاركة الشيفرة مع الآخرين بتواتر أعلى، وهذا يبشر لهم دمج تعديلاتك بسهولة دون حصول تضاربات.

وعلى النقيض من ذلك، فإن تنفيذ إيداعات كبيرة بفترات متباعدة يصعب دمج تعديلاتك ويزيد التضاربات في عملية الدمج.

## لا تقم بإيداع أعمال غير منتهية

يجب أن تقوم بالإيداع فقط عندما تتأكد من اكتمال الشيفرة المصدرة للموضوع الذي تعمل عليه. هذا لا يعني أن عليك أن تنتهي ميزة ضخمة قبل الإيداع. بل على العكس: قم بتقسيم كل جزء من عملك إلى أجزاء أصغر ذات حجم عمل مناسب ولا تنسى أن تقوم بالإيداع كلما سنحت لك الفرصة.

لا تقم بالإيداع لمجرد وضع شيء ما في المستودع قبل الخروج من المكتب. إذا أحسست أنك يجب أن تقوم بالإيداع من أجل الحصول على نسخة نظيفة للعمل عليها (الانتقال لفرع آخر، أو سحب التعديلات... إلخ) فالأفضل لك أن توظف ميزة التخينة (Stash).