

هناك العديد من البرامج المختلفة التي تستخدم لتطبيق تماقا كالعلاقة بين Word Microsoft والبراصج الأخرى ما هو control Version؟ أداة تدير التغييرات المتعلقة بالأكواد والملفات داخل المشروع ه أداة تتيح لك تخزين نسخة من الملفات علي جهازك الشخصي أو على الاداة المستخدمة لتطبيق ال version أداة تجعلك تتبع جميع التعديلات بالتوقيتات الزمنية لكل تعديل control مثل Git هو نظام تحكم في الإصدارات يستخدم على نطاق واسع في تطوير البرمجيات. في مشروع البرمجيات وإدارتها بكفاءة، بالإضافة إلى القدرة على العمل بشكل تعاوني مع فرق التطوير version control يوجد العديد من البرامج المختلفة التي تستخدم لتطبيق control version ال مثل: CVS - SVN - Perforce - Git ما هو الفرق بين Github vs. Git؟ ال Git هو عبارة عن control version يدير التغييرات المتعلقة بالأكواد والملفات داخل المشروع ال Github برنامج لإدارة وتنظيم الملفات (Repository) علي . Cloud . يسمح بمشاركة المشاريع وتعديلها من جهاز ما ال هي ال init git ال هي أمر يستخدم في Git لإنشاء مستودع (Repository) جديد. يتم استخدام هذا الأمر عندما تحتاج إلى تخزين جميع بيانات Git المتعلقة بمستودعك (Repository) مثل تاريخ المشروع وسجلات الإصدارات وغيرها من المعلومات يتم استخدام هذا الأمر عندما تحتاج إلى إنشاء مستودع (Repository) جديد لمشروعك الحالي أو لبدء مشروع جديد، عندما تقوم بتشغيل init git يقوم بإنشاء Repository مثل تاريخ المشروع وسجلات الإصدارات وغيرها من المعلومات ال add git والتي. تسمح لك بتحديد الملفات التي سيتم تضمينها في ال commit القادم م لإضافة ملفات جديدة يمكنك استخدام add git متبوعاً بأسماء الملفات على سبيل المثال git add commit يستخدم هذا الأمر لحفظ التغييرات التي تمت إضافتها إلى منطقة الانتظار (staging) area وتطبيقها على إنشاء commit جديد يمكنك استخدام git commit متبوعاً برسالة commit وصفا للتغييرات التي قمت بها. مما يسمح لك بالعودة إليها في أي وقت لتحديث المشروع أو استعادة نسخة سابقة منه log . هي أمر في git يستخدم لعرض سجل التزامات (Commits) المستودع (Repository) يعرض هذا الأمر قائمة بكل الالتزامات (Commits) الموجودة في الفرع الحالي للمستودع (Repository) ويشمل كل التزام (Commit) معلومات مثل الشفرة التعريفية للالتزام (commit) hash والمؤلف والتاريخ والرسالة التوضيحية للالتزام (Commit) بشكل عام git log يمكن استخدامه لفهم تاريخ المستودع وتحليل تطوره بشكل أفضل، لتحديد مكان الأخطاء ومراجعتها (gitignore) تجاهلها عند تتبع تغييرات المشروع باستخدام git 9 عندما تنشئ مستودعاً جديداً، ستضع ملفاتك ومجلداتك داخل مجلد العمل ، وستقوم بتتبع تغييرات هذه 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 استخدام ملف . هذه الملفات ، حتى لديهم تتبعها ضمن سجلات التاريخ والإصدارات Branches في ال Git تُعرف ال Branches بأنها تسمح لك بتطوير مسارات مختلفة للتغييرات على مستودع git الخاص بك عند إنشاء فرع (Branch) جديد يمكنك تطوير أجزاء أو إجراء التغييرات دون التأثير على الأجزاء الأخرى التي تعمل عليها فرق العمل الأخرى في نفس المشروع أي أنك سوف تأخذ الرئيسي بأي تعديل تقوم به وبعد أن تتأكد أن ما قمت بإضافته أو تعديله مذبوط تستطيع أن تقوم بإضافته للمشروع الرئيسي Git Branch هو أصراً في Git يستخدم لإنشاء فروع (Branches) جديدة في مستودع Git الخاص بك عندما تنشئ فرعاً (Branch) جديداً يتم إنشاء نسخة من كل شيء موجود في الفرع الحالي، بما في ذلك جميع الملفات وسجل التاريخ الخاص بالمستودع، لإنشاء فرع جديد يمكنك استخدام git branch نام متبوعاً باسم الفرع الجديد على سبيل المثال: git checkout new-feature Git عند تشغيل هذا الأمر مع اسم الفرع، يتم التحويل إلى هذا الفرع ويصبح الفرع النشط الذي يتم العمل عليه للتبديل إلى فرع معين، git checkout new-feature دون التأثير على الفرع الرئيسي أو أي فروع أخرى ال Revert Git لا يحذف أي تعديلات سابقة بل ينشئ الالتزام (commit) الجديد الذي يعكس حالة المستودع قبل إجراء التعديلات التي تم إجراؤها في الالتزام (commit) المحدد بمعنى آخر يتم إنشاء الالتزام الجديد الذي يحتوي على تعديلات تعكس الحالة السابقة للمستودع Git Switch يقوم بوظيفة كلد من git checkout و git branch معا عند تشغيل هذا الأمر مع اسم الفرع أو معرف الالتزام يتم التحويل إلى هذا الفرع أو الالتزام (commit) ويصبح الفرع أو الالتزام (commit) النشط الذي يتم العمل عليه للتبديل إلى فرع أو التزام (commit) معين، git switch -c new-feature git branch -d branchname "git switch -c new-feature git branch -d branchname" لحذف فرع Git من مستودعك ال استخدام git merge في ذلك الفرع git merge من فرع إلى آخر التغييرات من الفرع الأول والفرع الثاني يتم استخدام الأمر "git merge" بالشكل التالي حيث أن "branchname" هو اسم الفرع الذي تريد دمجه بالفرع الحالي، ومن المهم التأكد من التحديث الانتقال (إلى الفرع الرئيسي قبل استخدام هذا الأمر وذلك باستخدام

الأمر git checkout main-branch fast-forward merge الرئيسي دون إنشاء نقطة تفج جديدة. يتم استخدام هذا النوع من الدمج عندما يتم العمل على فرع فرعي بشكل منفرد وبدون أي تداخلات مع الفرع الرئيسي the 3-way merge تغييرات في ال main في هذ الحالة يقوم cit بإنشاء نقطة تفرع جديدة (merge) (commit) ويحاول دمج الفيان ال+. الفرعين Squash Merge في Merge في ات فيجب عليك التأكد من الحفاظ على تاريخ المشروع عرتنا ومنظفا وذلك باستخدام رسائل الالتزامات قالبت الصحيحة ووصف دقيق للتعديلات المدمجة يعا i Merge conflicts ودمج التعارضات جزءا من عملية الدمج (merge) في Git ج الجزء في فرعين مختلفين بطريقة عختلفة من وتحديث عملية conflicts Merge عندما نتم تعديل نفس الملف قبل أكثر من مستخدم في نفس الوقت، وهذا يؤدي إلى وجود تعارضات بين التعديلات المختلفة التي يجب حلها يمكن حل التعارضات بطرق مختلفة مثل دمج التعديلات المختلفة بشكل يدوي، أو اختيار إحدى التعديلات على أو استخدام أدوات تلقائية لدمج التعديلات ويعتمد الخيار المناسب على طبيعة التعارض وحجم وعند الانتهاء من حل التعارضات، يتم حفظ النسخة الجديدة Merge cit يحافظ على تاريخ المشروع الخطي (Linear) مما يسهل تتبع التغييرات وفهم عملية التطوير لايقوم بتغيير الفرع (Branch) الأصلي أو تاريخ الالتزامات (Commits) الأصلي العيوب يمكن أن يؤدي إلى تاريخ التزامات (Commits) فوضوي ومربك عند دمج فروع متعددة ه قد يتطلب وقتا وجهذا إضافيا لحل تضاربات الدمج Cit Rebase المميزات يوفر تاريخ التزامات (Commits) أكثر نظافة وتنظيكا عن طريق دمج التغييرات من فرع واحد في آخر بدون إنشاء علامات دمج غير ضرورية . يتيح عملية تطوير أكثر تيسيرا وخطية . يقلل من عدد تضاربات الدمج التي يتعين حلها حيث يتم دمج التغييرات في تاريخ خطي