

ইন্সটল গিট

গিটহাবের রিপোজিটরির সাধারণ অ্যাকশন ও উন্নত পরিস্থিতিতে জন্য git এর একটি স্বয়ংক্রিয়ভাবে আপডেট করার কমান্ড লাইন সংস্করণসহ ডেস্কটপের জন্য গ্রাফিকাল ইউজার ইন্টারফেস সুবিধায়ুক্ত ক্লায়েন্ট আছে।

উইন্ডোজের জন্য গিটহাব

windows.github.com

ম্যাকের জন্য গিটহাব

mac.github.com

লিনাক্স ও posix সিস্টেমের জন্য গিট অফিশিয়াল git scm ওয়েবসাইটে উপলব্ধ।

সকল প্ল্যাটফর্মের জন্য গিটহাব

git-scm.com

টুল কনফিগার করা

Git কনফিগার করাঃ একবার করার প্রয়োজন হয়।

```
$ git config --global user.name "[name]"
নাম ঠিক করুন যেটা আপনি কমিট ট্রাসেকশনের সাথে সংযুক্ত করতে চান।
$ git config --global user.email "[email address]"
ইমেইল ঠিক করুন যেটা আপনি কমিট ট্রাসেকশনের সাথে সংযুক্ত করতে চান।
```

রিপোজিটরি তৈরি করা

নতুন repository তৈরি বা ডাউনলোড করার জন্য।
\$ git init [project-name]
নির্দিষ্ট নাম দিয়ে একটি নতুন repository তৈরি করুন।
\$ git clone [url]
একটি প্রকল্প এবং তার সম্পূর্ণ সংস্করণ ইতিহাস ডাউনলোড করুন।

পরিবর্তন করা

সম্পাদনা পর্যালোচনা করুন এবং একটি কমিট ট্রাসেকশন ক্রাফট করুন।
\$ git status
সকল ফাইল লিস্ট করুন যা কমিট করা হবে।
\$ git diff
ফাইলগুলোর পার্থক্য দেখুন যেগুলো এখনো staged হয়নি।
\$ git add .
সংস্করণের সকল ফাইল ম্যাপশট করুন।
\$ git add [file]
সংস্করণের জন্য প্রস্তুতি ফাইল ম্যাপশট করুন।
\$ git diff --staged
staging এবং last file version এর মধ্যে পার্থক্য দেখুন।
\$ git reset [file]
ফাইল unstages, কিন্তু তার বিষয়বস্তু অপরিবর্তিত রাখুন।
\$ git commit -m"[descriptive message]"
ফাইলের ম্যাপশট ভার্সনের ইতিহাসে সংরক্ষণ করুন।

গ্রুপ পরিবর্তন

Name a series of commits and combine completed efforts

```
$ git branch
বর্তমান repositoryর মধ্যে সব লোকাল ব্রাঞ্চ তালিকাভুক্ত করুন।
$ git branch [branch-name]
একটি নতুন ব্রাঞ্চ তৈরি করুন।
$ git switch -c [branch-name]
নির্দিষ্ট শাখায় সুইচ এবং আপডেট ওয়াকিং ডিরেক্টরি।
$ git merge [branch-name]
বর্তমান শাখায় নির্দিষ্ট শাখার ইতিহাস একত্রিত করুন
$ git branch -d [branch-name]
নির্দিষ্ট শাখা মুছে ফেলে।
```

ফাইলের নাম পরিবর্তন (refactor) করুন

versioned ফাইল নতুন স্থানে স্থাপন বা মুছে ফেলুন

```
$ echo >> [file name]
নতুন ফাইল যোগ করার জন্য।
ওয়াকিং ডিরেক্টরি থেকে ফাইল মুছে এবং মুছে ফেলার পর্যায়ে।
$ git rm [file]
ওয়াকিং ডিরেক্টরি থেকে ফাইল মুছে এবং মুছে ফেলার পর্যায়ে।
$ git rm --cached [file]
ভার্সন নিয়ন্ত্রণ থেকে ফাইল মুছে ফেলা কিন্তু স্থানীয়ভাবে ফাইল সংরক্ষণ করা।
$ git mv [file-original] [file-renamed]
ফাইলের নাম পরিবর্তন করা এবং কমিট জন্য এটি প্রস্তুত করুন।
```

ট্র্যাকিং দমন

অস্থায়ী ফাইল এবং পাথ বাদ দিন

```
*.log
build/
temp-*
.gitignore নামের একটি পার্শ্ব ফাইল নির্দিষ্ট প্যাটার্নের সাথে মিলে যাওয়া ফাইল
এবং পাথগুলির দুর্ঘটনাজনিত সংস্করণকে দমন করে।
$ git ls-files --others --ignored --exclude-standard
এই প্রকল্পে সব উপেক্ষিত ফাইলের তালিকা তৈরি করুন।
```

ফ্রাগমেন্টস সংরক্ষণ

অসম্পূর্ণ পরিবর্তন পুনরুদ্ধার এবং সরাসরি রাখা।
\$ git stash
সাময়িকভাবে সব পরিবর্তিত ট্র্যাক ফাইল স্টোর করা।
\$ git stash pop
সাম্প্রতিক stashed ফাইল রিস্টোর করা।
\$ git stash list
সব stashed changesets ফাইলের তালিকা তৈরি করুন।
\$ git stash drop
সাম্প্রতিক stashed changeset ফাইল বর্জন করুন।

হিস্টোরি রিভিউ

ব্রাউজ করুন এবং প্রকল্পের ফাইল এর বিবর্তন পরিদর্শন করুন।
\$ git log
বর্তমান শাখার জন্য ভার্সন ইতিহাসের তালিকা তৈরি করুন।
\$ git log --follow [file]
বর্তমান শাখার জন্য ভার্সন ইতিহাসের তালিকা তৈরি করুন, renames সহ।
\$ git diff [first-branch]...[second-branch]
দুই শাখার মধ্যে বিষয়বস্তুর পার্থক্য দেখুন।
\$ git show [commit]
নির্দিষ্ট আউটপুট মেটাডেটা এবং বিষয়বস্তু পরিবর্তন কমিট করুন।

কমিট রিডো করা

ভুল এবংক্রাফট প্রতিস্থাপনের ইতিহাস মুছুন।
\$ git reset [commit]
undoes all commits after [commit], preserving changes locally
\$ git reset --hard [commit]
সব ইতিহাস এবং করা পরিবর্তনগুলি ফিরে নির্দিষ্ট কমিটে অস্বীকার করুন।

সিক্রোনাইজড পরিবর্তনগুলি

একটি দূরবর্তী (url) নিবন্ধন করুন এবং repository ইতিহাস একত্রিত করুন।
\$ git fetch [remote]
দূরবর্তী repository থেকে সব ইতিহাস ডাউনলোড করুন।
\$ git merge [remote]/[branch]
বর্তমান স্থানীয় শাখা ও দূরবর্তী শাখা একত্রিত করুন।
\$ git push [remote] [branch]
সকল লোকাল ব্রাঞ্চ কমিট গিটহাবে আপলোড করুন।
\$ git pull
বুকমার্ক হিস্টোরি ও অন্তর্ভুক্ত পরিবর্তনগুলো ডাউনলোড করুন।