



Git स्थापित करें

GitHub डेस्कटॉप क्लाइंट प्रदान करता है, जिसमें सबसे आम रिपॉजिटरी कार्यों के लिए एक ग्राफिकल यूजर इंटरफेस (GUI) है और उन्नत परिदृश्य के लिए Git का एक खुद ब खुद सामयिक बनने वाली कमांड लाइन संस्करण.

GitHub Desktop

<https://desktop.github.com/>

LINUX या POSIX सिस्टम के लिए Git वितरण आधिकारिक Git SCM वेबसाइट से उपलब्ध हैं।

Git सभी प्लेटफ़ॉर्म के लिए

<http://git-scm.com>

उपकरण सेटिंग्स

सभी स्थानीय रिपॉजिटरी के लिए उपयोगकर्ता जानकारी कॉन्फ़िगर करें

```
$ git config --global user.name "[name]"
```

वह नाम सेट करें जिसे आप अपने कमिटेड लेनदेन से जोड़ना चाहते हैं

```
$ git config --global user.email "[email address]"
```

वह ईमेल सेट करें जिसे आप अपने कमिटेड लेन-देन से जोड़ना चाहते हैं

रिपॉजिटरी बनाएं

एक नया रिपॉजिटरी शुरू करें या किसी मौजूदा URL से प्राप्त करें

```
$ git init [project-name]
```

निर्दिष्ट नाम के साथ एक स्थानीय भंडार बनाएँ

```
$ git clone [url]
```

परियोजना और सभी संस्करण इतिहास डाउनलोड करें

परिवर्तन करें

संपादन की समीक्षा करें और कमिटेड लेनदेन शिल्प करें

```
$ git status
```

कमिटेड होने वाली नयी एवं बदली गयी फाइलों की सूची देखे

```
$ git diff
```

उन फाइलों के डिफरेंसेस दिखाएं, जो अभी तक स्टेज नहीं हुई है

```
$ git add [file]
```

संस्करण नियंत्रण के लिए फ़ाइल स्नैपशॉट बनाएँ

```
$ git diff --staged
```

स्टेजिंग और अंतिम फ़ाइल संस्करण के बीच अंतर दिखाएं

```
$ git reset [file]
```

फ़ाइल को स्टेजिंग से निकालें, लेकिन इसकी सामग्री को रखें

```
$ git commit -m"[descriptive message]"
```

संस्करण इतिहास में फ़ाइल स्नैपशॉट को स्थायी रूप से रिकॉर्ड करें

समूह परिवर्तन

पूर्ण किए गए पर्यासों की एक श्रृंखला का नाम दें

```
$ git branch
```

वर्तमान रिपॉजिटरी पर सभी स्थानीय शाखाओं को सूचीबद्ध करें

```
$ git branch [branch-name]
```

एक नई शाखा बनाएँ

```
$ git checkout [branch-name]
```

निर्दिष्ट शाखा पर स्विच करें और कार्य निर्देशिका को अपडेट करें

```
$ git merge [branch-name]
```

वर्तमान शाखा में निर्दिष्ट शाखा के इतिहास को मर्ज करें

```
$ git branch -d [branch-name]
```

निर्दिष्ट शाखा को हटा दें

फ़ाइल नाम संगठन

संस्करणित फ़ाइलों को स्थानांतरित करें और हटाएं

```
$ git rm [file]
```

फ़ाइल को कार्य निर्देशिका और चरण हटाने से हटाएं

```
$ git rm --cached [file]
```

संस्करण नियंत्रण से फ़ाइलें निकालें और स्थानीय फ़ाइलें रखें

```
$ git mv [file-original] [file-renamed]
```

फ़ाइल का नाम बदलें और प्रतिबद्ध कमिट के लिए तैयार करें

ट्रैकिंग प्रतिबंध

अस्थायी फ़ाइलों और रास्तों को छोड़ दें

```
*.log  
build/  
temp-*
```

.gitignore आपको निर्दिष्ट पैटर्न से मेल खाने वाली फ़ाइलों और रास्तों से रोकता है

```
$ git ls-files --others --ignored --exclude-standard
```

प्रोजेक्ट शामिल ना की गई फ़ाइलों को सूचीबद्ध करें

टुकड़े सहेजें

अपूर्ण परिवर्तनों को आश्रय देना और पुनर्स्थापित करना

```
$ git stash
```

सभी परिवर्तनों के साथ ट्रैक की गई फ़ाइलों को अस्थायी रूप से सहेजें

```
$ git stash pop
```

सबसे हाल ही में सहेजी गई फ़ाइल पुनर्प्राप्त करें

```
$ git stash list
```

सभी अस्थायी रूप से सहेजे गए परिवर्तन सेटों को सूचीबद्ध करें

```
$ git stash drop
```

सबसे हाल ही में सहेजे गए परिवर्तन को त्याग देता है

इतिहास की जाँच

प्रोजेक्ट फ़ाइल की प्रगति की जाँच करें

```
$ git log
```

संस्करण इतिहास की सूची देखे वर्तमान शाखा के लिए

```
$ git log --follow [file]
```

फ़ाइल के लिए संस्करण इतिहास सूचीबद्ध करता है, जिसमें नाम शामिल है

```
$ git diff [first-branch]...[second-branch]
```

दो शाखाओं के बीच सामग्री अंतर दिखाता है

```
$ git show [commit]
```

आउटपुट मेटा जानकारी और निर्दिष्ट प्रतिबद्ध के परिवर्तन

फिर कमिट से करना

गलतियों को हटाएं और इतिहास को बदल दें

```
$ git reset [commit]
```

[commit] के बाद सभी कमिट हटाता हैं, स्थानीय स्तर पर परिवर्तन रखते हुए

```
$ git reset --hard [commit]
```

निर्दिष्ट कमिट पर लौटें और बाद के सभी परिवर्तनों को छोड़ दें

सिंक्रनाइज़ेशन बदलें

रजिस्टर रिपॉजिटरी ब्रूकमार्क(URL) और एक्सचेंज वर्जन हिस्ट्री

```
$ git fetch [remote]
```

सभी इतिहास को रिपॉजिटरी बुकमार्क से डाउनलोड करें

```
$ git merge [remote]/[branch]
```

वर्तमान स्थानीय शाखा में बुकमार्क शाखा को मर्ज करें

```
$ git push [remote] [branch]
```

सभी स्थानीय शाखा GitHub पर अपलोड करें

```
$ git pull
```

बुकमार्क इतिहास डाउनलोड करें और परिवर्तनों को समेकित करें

Made with ❤️ by 🐱s and friends

  **COMMUNITY**

© 2019 GitHub, Inc. Jekyll & Primer.