

Git उपनामहरू

अर्को अध्यायमा जानु अघि, हामी तपाईंको Git अनुभवलाई सरल, सजिलो, र बढी परिचित बनाउन सक्छ भन्ने सुविधा प्रस्तुत गर्न चाहन्छौं: उपनामहरू। स्पष्टताका लागि, हामी यो पुस्तकमा अरु कुनै ठाउँमा प्रयोग गर्ने छैनौं, तर यदि तपाईं नियमित रूपमा Git प्रयोग गर्नुहुन्छ भने, उपनामहरू बारे जान्नुपर्छ।

Git ले स्वचालित रूपमा तपाईंको आदेशलाई अनुमान गर्दैन यदि तपाईंले त्यसलाई आंशिक रूपमा टाइप गर्नुभयो भने। यदि तपाईं प्रत्येक Git आदेशको पूरै पाठ टाइप गर्न चाहनुहुन्न भने, तपाईं सजिलैसँग `git config` प्रयोग गरेर प्रत्येक आदेशको लागि उपनाम सेट गर्न सक्नुहुन्छ। यहाँ केही उदाहरणहरू छन् जुन तपाईंले सेट गर्न चाहनुहुन्छ:

```
$ git config --global alias.co checkout
$ git config --global alias.br branch
$ git config --global alias.ci commit
$ git config --global alias.st status
```

यसको अर्थ, उदाहरणका लागि, **git commit** टाइप गर्नको सट्टा तपाईंले केवल **git ci** टाइप गर्नुपर्ने हुन्छ। जब तपाईं Git प्रयोग गर्दै जानुहुन्छ, तपाईंले अन्य आदेशहरू पनि बारम्बार प्रयोग गर्नुहुनेछ; नयाँ उपनामहरू बनाउन हिचकिचाउनु पर्दैन। यो विधि तपाईंले हुनु पर्छ भनेर सोच्ने आदेशहरू बनाउनका लागि पनि धेरै उपयोगी हुन सक्छ। उदाहरणका लागि, कुनै फाइललाई unstaging गर्दा देखिने प्रयोगको समस्या समाधान गर्न, तपाईंले आफ्नो आफ्नै **unstage** उपनाम Git मा थप्न सक्नुहुन्छ:

```
$ git config --global alias.unstage 'reset HEAD --'
```

यसले निम्न दुई आदेशहरूलाई बराबर बनाउँछ:

```
$ git unstage fileA
$ git reset HEAD -- fileA
```

यो अलि बढी स्पष्ट देखिन्छ। यो पनि सामान्य हो कि तपाईंले **last** जस्तै एउटा आदेश थप्नुहोस्, यसरी:

```
$ git config --global alias.last 'log -1 HEAD'
```

यसरी, तपाईंले अन्तिम कमिट सजिलैसँग देख्न सक्नुहुन्छ:

```
$ git last
commit 66938dae3329c7aeb598c2246a8e6af90d04646
Author: Josh Goebel <dreamer3@example.com>
Date: Tue Aug 26 19:48:51 2008 +0800
```

```
Test for current head
```

```
Signed-off-by: Scott Chacon <schacon@example.com>
```

जसरी तपाईं भन्न सक्नुहुन्छ, Git ले नयाँ आदेशलाई तपाईंले जुन उपनाम दिनुभएको छ त्यसमा प्रतिस्थापन गर्दछ। यद्यपि, सायद तपाईं Git को उप-आदेशको सट्टा बाह्य आदेश चलाउन चाहनुहुन्छ। त्यस अवस्थामा, तपाईं आदेश! अक्षरबाट सुरु गर्नुहुन्छ। यो उपयोगी छ यदि तपाईंले आफ्नो Git रिपोजिटरीसँग काम गर्ने आफ्नै उपकरणहरू लेख्नु भएको छ भने। हामी यसलाई **git visual** लाई **gitk** चलाउनका लागि उपनाम दिई प्रदर्शन गर्न सक्छौं:

```
$ git config --global alias.visual '!gitk'
```