

GIT को आधारभूत ज्ञान

यदि तपाईंले Git सिक्नको लागि मात्र एउटा अध्याय पढ्न सक्नुहुन्छ भने, यो नै हो। यो अध्यायले तपाईंलाई Git को आधारभूत आदेशहरू बुझाउन मद्दत गर्छ, जसले गर्दा तपाईंले समयको ठूलो भागमा गर्नुपर्ने कामहरू गर्न सक्नुहुनेछ।

यो अध्यायको अन्त्यसम्म, तपाईंले एउटा repository कन्फिगर र initialize गर्ने, फाइलहरू ड्र्याक गर्न सुरु र रोक्ने, र परिवर्तनहरू stage तथा commit गर्ने तरिकाहरू सिक्नुहुनेछ। हामी तपाईंलाई के-कसरी Git लाई निश्चित फाइलहरू र फाइल ढाँचाहरूलाई ignore गर्न सेटअप गर्ने, गल्तीहरू छिटो र सजिलै ठीक गर्ने, आफ्नो परियोजनाको इतिहास हेर्ने र commit हरूबीचका परिवर्तनहरू जाँच्ने तरिका, साथै remote repository बाट push र pull गर्ने प्रक्रिया पनि देखाउनेछौं।

GIT REPOSITORY प्राप्त गर्ने तरिका

सामान्यतया, तपाईंले Git repository दुई तरिकाबाट प्राप्त गर्न सक्नुहुन्छ:

1. तपाईंले हाल version control मा नरहेको कुनै स्थानीय directory लाई Git repository मा परिवर्तन गर्न सक्नुहुन्छ, वा
2. तपाईंले कुनै अर्को स्थानबाट अवस्थित Git repository लाई clone गर्न सक्नुहुन्छ।

यी दुवै अवस्थामा, तपाईंको स्थानीय मेसिनमा Git repository तयार हुन्छ, र काम सुरु गर्न सक्नुहुन्छ।

अवस्थित DIRECTORY मा REPOSITORY INITIALIZE गर्ने तरिका

यदि तपाईंसँग एउटा project directory छ जुन हाल version control मा छैन र त्यसलाई Git बाट नियन्त्रण गर्न चाहनुहुन्छ भने, सुरुमा तपाईंले उक्त project को directory मा जानुपर्ने हुन्छ। तपाईंले यो कहिल्यै गर्नुभएको छैन भने, यो प्रक्रिया तपाईंले प्रयोग गरिरहेको सिस्टममा निर्भर गर्दै फरक देखिन्छ:

त्यसपछि टाइप गर्नुहोस्:

```
$ git init
```

यसले एउटा नयाँ subdirectory **.git** बनाउँछ, जसमा तपाईंको repository का सबै आवश्यक फाइलहरू हुन्छन् — यो Git repository को skeleton हो। यो बिन्दुमा, तपाईंको project मा कुनै पनि फाइल ड्र्याक भएको हुँदैन।

अवस्थित फाइलहरूलाई VERSION CONTROL मा राख्ने तरिका

यदि तपाईंले खाली directory को सट्टा अवस्थित फाइलहरूलाई version control मा राख्न चाहनुहुन्छ भने, तपाईंले ती फाइलहरूलाई ड्र्याक गर्न सुरु गर्नु पर्छ र एक प्रारम्भिक commit गर्नु पर्छ। यो गर्नका लागि, तपाईंले केही **git add** आदेशहरू प्रयोग गरेर ड्र्याक गर्न चाहने फाइलहरू निर्दिष्ट गर्नुहोस्, त्यसपछि **git commit** गर्नुहोस्:

```
$ git add *.c  
$ git add LICENSE  
$ git commit -m 'Initial project version'
```

अवस्थित REPOSITORY CLONE गर्ने तरिका

यदि तपाईंले कुनै अवस्थित Git repository को प्रतिलिपि प्राप्त गर्न चाहनुहुन्छ — जस्तै, तपाईं योगदान दिन

चाहनुभएको कुनै project – तपाईंलाई चाहिने आदेश हो `git clone`। यदि तपाईं Subversion जस्ता अन्य VCSs संग परिचित हुनुहुन्छ भने, तपाईंले देख्नुहुनेछ कि यसमा आदेश "clone" हो, "checkout" होइन। यो एउटा महत्वपूर्ण भिन्नता हो – केवल working copy प्राप्त गर्ने सट्टा, Git ले server मा भएको लगभग सबै डेटा को पूर्ण प्रतिलिपि प्राप्त गर्छ। जब तपाईं `git clone` चलाउनुहुन्छ, तपाईंले project को इतिहासको हरेक संस्करणसहितका सबै फाइलहरूको पूर्ण प्रतिलिपि प्राप्त गर्नुहुन्छ। वास्तवमा, यदि तपाईंको server को disk बिग्रन्छ भने, तपाईं client को कुनै पनि clone प्रयोग गरेर server लाई clone गरेको अवस्थामा फिर्ता ल्याउन सक्नुहुन्छ (तपाईं server-side hooks जस्ता केही कुरा गुमाउन सक्नुहुन्छ, तर सबै versioned डेटा सुरक्षित हुनेछ)।

REPOSITORY CLONE गर्ने आदेश

तपाईंले कुनै repository लाई clone गर्न `git clone <url>` आदेश प्रयोग गर्नुहुन्छ। उदाहरणका लागि, यदि तपाईंले Git को linkable library जसलाई libgit2 भनिन्छ, clone गर्न चाहनुहुन्छ भने, तपाईं यसरी गर्न सक्नुहुन्छ:

```
bash
$ git clone https://github.com/libgit2/libgit2
```

यसले libgit2 नामको एक directory बनाउँछ, त्यसको भित्र .git directory initialize गर्छ, त्यस repository का सबै डेटा pulls गर्छ, र पछिल्लो संस्करणको working copy checkout गर्छ। यदि तपाईं नयाँ रूपमा बनाइएको libgit2 directory भित्र जानुहुन्छ भने, त्यहाँ project का फाइलहरू हेर्न सक्नुहुनेछ, जसलाई काम गर्न वा प्रयोग गर्न तयार गरिएको हुन्छ।

यदि तपाईंले repository लाई libgit2 सट्टा अरु कुनै नामको directory मा clone गर्न चाहनुहुन्छ भने, तपाईं नयाँ directory को नाम अतिरिक्त argument को रूपमा दिन सक्नुहुन्छ:

```
bash
$ git clone https://github.com/libgit2/libgit2 mylibgit
```

यो आदेशले अघिल्लो आदेशसँगै एउटै काम गर्दछ, तर target directory को नाम mylibgit हुन्छ।

Git ले डेटा स्थानान्तरण गर्नका लागि विभिन्न प्रकारका transfer protocol हरूलाई समर्थन गर्छ। माथिको उदाहरणले `https://` protocol प्रयोग गरेको छ, तर तपाईंले `git://` वा

`user@server:path/to/repo.git` जस्ता SSH transfer protocol पनि देख्न सक्नुहुन्छ।