



http://antkh.com/

# មេរៀន Git & GitHub

រៀបរៀងដោយ៖ ហ៊ុន ស្រីឡាន



What is Git?

Git Download

Basic Git  
Commands

Web-Based  
Platforms

What is GitHub?

Basic GitHub  
Workflow

What is GitLab?

Different between  
GitHub & GitLab



# What is Git?



- » **Git** គឺជាប្រព័ន្ធគ្រប់គ្រងនៃការកែប្រែ ដែលត្រូវបានគេបង្កើតឡើងដើម្បីកំណត់ទៅលើការផ្លាស់ប្តូរ ការកែប្រែ Source Code នៅក្នុងការអភិវឌ្ឍកម្មវិធី។
- » នៅក្នុងការគ្រប់គ្រងគម្រោងធំៗ គេប្រើប្រាស់ **Git** ព្រោះវាមាន ប្រសិទ្ធភាព លឿន និងសមត្ថភាពគ្រប់គ្រាន់ក្នុងការបំពេញការងារ។





# Git Download



➤ ដើម្បីប្រើប្រាស់ Command ផ្សេងៗដែលមាននៅក្នុង Git ត្រូវ [Download Git](#) ជាមុនសិន:

- △ ជំហានទី១៖ [Git Download](#)
- △ ជំហានទី២៖ Run the Installer
- △ ជំហានទី៣៖ Verify the installation

ប្រើប្រាស់នៅក្នុង Command Prompt or  
Git Bash ដើម្បីដឹងពី Version របស់ Git



```
1 git --version
```





# Basic Git Commands



## Initialize a New Repository

សម្រាប់បង្កើតនៅ Git Repository ថ្មីនៅក្នុង Current Repository របស់យើង។



```
1 git init
```



## Clone an Existing Repository

This command creates a copy of an existing repository



```
1 git clone [repository URL]
```



## Add Files to the Staging Area

- `git add [File]` ជ្រើសរើស File ណាមួយ ដែលចង់ Add ចូល
- `git add .` វានឹង Add Files ចូលទាំងអស់នៅពេលដែល មានFile ណាមួយមានការប្រែប្រួល។



```
1 git add [File]
2 git add .
```





# Basic Git Commands



## Commit Changes

សម្រាប់កំណត់ នៅអ្វីដែលយើងបានផ្លាស់ប្តូរ រឺ បង្កើតថ្មីសម្រាប់បញ្ជាក់



```
1 git commit -m "Commit message"
```



## Push Changes to a Remote Repository

សម្រាប់ Push នៅ {Files} ផ្សេងៗដែលបានបង្កើត រឺ កែ



```
1 git push origin [branch]
```



## Pull Changes from a Remote Repository

សម្រាប់ ទាញយក នៅ {Files} ផ្សេងៗដើម្បីអោយ Up to Date



```
1 git pull origin [branch]
```





# Basic Git Commands



## Create a New Branch



```
1 git branch [branch name]
```

សម្រាប់បង្កើតនូវឈ្មោះ Branch មួយថ្មីចេញពី Branch ដែលមានស្រាប់។



## Switch to a Branch



```
1 git checkout -b [branch name]
2 or
3 git switch -c [branch name]
```

សម្រាប់បង្កើតនូវឈ្មោះ Branch មួយថ្មីចេញពី Branch ដែលមានស្រាប់ និង Switch ទៅកាន់ឈ្មោះ Branch ដែលបានបង្កើត។





# Basic Git Commands

## List Branches



```
1 git branch
```

សម្រាប់បង្ហាញ នូវឈ្មោះ Branch ផ្សេងៗដែលមាននៅក្នុង Repository

## Delete a Branch



```
1 git delete -d [branch-name]
```

សម្រាប់លុបឈ្មោះ Branch ចេញពី Repository

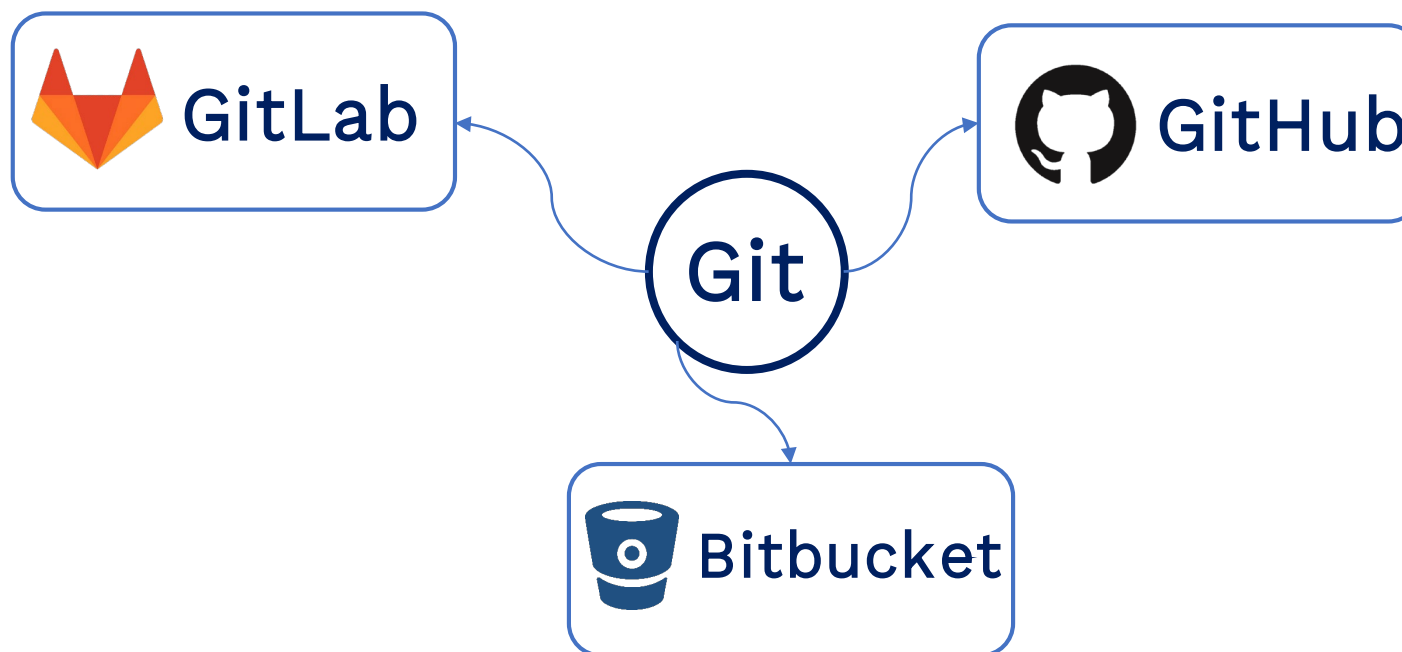




# Web-Based Platforms Compatible with Git



➤ ចំពោះ Web-Based Platforms ដែលប្រើប្រាស់ជាមួយ និង Git:







## What is GitHub?



➤ **GitHub** គឺជា Web-Based Collaborative Platform សម្រាប់ការអភិវឌ្ឍប្រព័ន្ធ ដែលអាចអោយ Git Repository ធ្វើការ Hosting ពិនិត្យកូដ ការតាមដានបញ្ហា និងគ្រប់គ្រងគម្រោងផ្សេងៗ។ វាអនុញ្ញាតអោយអ្នកប្រើប្រាស់ធ្វើការរួមគ្នាដើម្បីកំណត់ ទៅលើគម្រោងនីមួយៗ។

➤ **GitHub** មានមុខងារដូចជា៖

- Collaboration
- Bug Tracking
- Branches
- Git Repositories
- Project Management
- Team Management
- Code Hosting
- Track and Assign Tasks





# Basic GitHub Workflow



## 1 Create a Repository

- 🏠 ជំហានទី១៖ Sign In to GitHub
- 🏠 ជំហានទី២៖ Navigate to New Repository
- 🏠 ជំហានទី៣៖ Fill in Repository Details

Repository Name: ដាក់ឈ្មោះអោយ Repository.

Description (Optional): បន្ថែមការពិពណ៌នាសង្ខេបអំពីគម្រោងរបស់អ្នក។

Public/Private: ជ្រើសរើស Visibility សម្រាប់ Repository របស់អ្នក។

Initialize: Optionally add a README file, .gitignore, and license.





http://antkh.com/

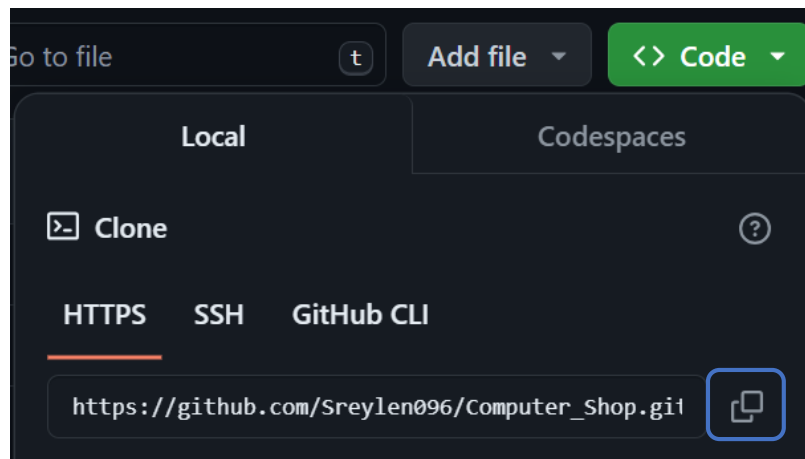


# Basic GitHub Workflow



## 2 Clone the Repository

🏠 ជំហានទី១៖ ចូលទៅកាន់ Repository ដែលបានបង្កើតនៅក្នុង GitHub



Copy Link URL



What is Git?

Git Download

Basic Git  
Commands

Web-Based  
Platforms

What is GitHub?

Basic GitHub  
Workflow

What is GitLab?

Different between  
GitHub & GitLab



## Basic GitHub Workflow



🏠 នៅពេលដែលយើងទើបបង្កើត Account GitHub បើយើងចង់ Add | Commit | Push Code ចូលទៅកាន់ GitHub យើងត្រូវប្រើប្រាស់ Command ដូចខាងក្រោម ដើម្បីអោយ VS Code ស្គាល់ Account GitHub របស់យើងជាមុនសិន ដោយប្រើប្រាស់ Command Prompt or Git Bash៖



```
1 git config --global user.name "Your Name"
2 git config --global user.email "your_email@example.com"
```

➤ “Your Name” : ឈ្មោះ GitHub របស់អ្នក

➤ “your\_email@example.com” : Email របស់អ្នក ដែលបានភ្ជាប់ជាមួយ GitHub





## Basic GitHub Workflow

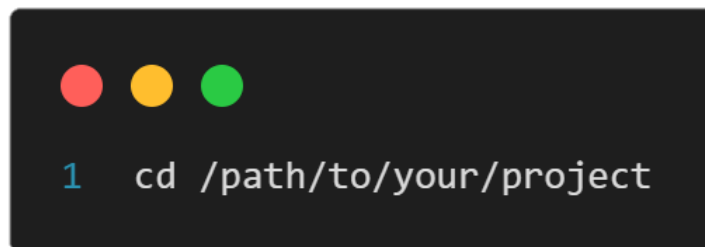


△ នៅពេលដែលយើងបានប្រើប្រាស់ Command ដើម្បី Config ទៅលើ UserName and Email រួចហើយ យើងអាចប្រើប្រាស់នូវ Command ដូចខាងក្រោមដើម្បីបង្កើតនៅ Git Repository ថ្មីនៅក្នុង Current Repository របស់យើង។



```
1 git init
```

△ **ចំណាំ៖** នៅពេលអ្នកចង់ចូលទៅកាន់ Project Repository ណាមួយ ត្រូវប្រើប្រាស់ '**cd**' រួចជ្រើសរើស Path Project ដែលអ្នកចង់ចូលទៅកាន់។



```
1 cd /path/to/your/project
```





# Basic GitHub Workflow



## 2 Clone the Repository

🏠 ជំហានទី២៖

Open your terminal or Git Bash, ជ្រើសរើស Directory ដែលចង់ Clone Repository ដាក់



```
1 git clone [repository URL]
```

Repository URL ដែលបានមកពីជំហានទី១





# Basic GitHub Workflow

## 3 Make Changes

```
1 git checkout -b [branch name]
2 or
3 git switch -c [branch name]
```

```
1 git add [File]
2 git add .
```

```
1 git commit -m "Commit message"
```

## 4 Push Changes

```
1 git push origin [branch]
```





# Basic GitHub Workflow



## 5 Create a Pull Request

បង្កើត Pull Request ដើម្បី Merge នូវការងារដែលស្ថិតនៅក្នុង Branch របស់យើង ចូលទៅកាន់ Branch main។

## 6 Review and Merge

Collaborate with team members to review the pull request. Once approved, merge the changes.







# Basic GitHub Workflow



5

## Merge Conflict

នៅពេលដែល Developer ជាច្រើននាក់ បានកែ រឺ ផ្លាស់ប្តូរកូដនៅ បន្ទាត់ដូចគ្នា ប៉ុន្តែស្ថិតនៅ Branch ផ្សេងគ្នា នោះនឹងធ្វើអោយកើតមាននូវ Conflict កើតឡើង។



```
1 <<<<<< HEAD
2 print("Hello from main")
3 =====
4 print("Hello from feature")
5 >>>>>> feature-branch
```

តាមរយៈរូបភាពខាងស្តាំ កូដកើតមាន Conflict ដែលតម្រូវអោយយើង ត្រូវជ្រើសរើស នូវជម្រើសដូចខាងក្រោម៖

Accept Current Change | Accept Incoming Change

Accept Both Changes | Compare Changes





## Basic GitHub Workflow



5

### Merge Conflict

Accept Current Change

មានន័យថា រក្សានូវការផ្លាស់ប្តូរ នៅលើ Branch ដែលយើងស្ថិតនៅ។  
**Ex:** `print("Hello From Main");`

Accept Incoming Change

មានន័យថា រងទៅយកការផ្លាស់ប្តូរ នៅ Branch ផ្សេងដែលបានធ្វើការកែប្រែ  
**Ex:** `print("Hello From Feature");`

Accept Both Changes

មានន័យថា រងទៅយកទាំងកូដដែលស្ថិតនៅ Branch យើងនិង Branch ផ្សេង។  
**Ex:** `print("Hello From Main"); print("Hello From Feature");`

Compare Changes

មានន័យថា រងទៅពិនិត្យទៅលើ Branch ទាំងពីរ ដើម្បីធ្វើការកែប្រែ។ (Manual Edit)  
**Ex:** `print("Hello from both Main and feature");`





## What is GitLab?



➤ **GitLab** គឺជា Web-Based Collaborative Platform សម្រាប់ការអភិវឌ្ឍប្រព័ន្ធ ។ វាគឺជា DevOps ពេញលេញដែលអាចបំពេញកិច្ចការទាំងអស់នៅក្នុងគម្រោងមួយ ចាប់ពីការធ្វើផែនការគម្រោង និងការគ្រប់គ្រង កូដ រហូតដល់ការត្រួតពិនិត្យ និងសុវត្ថិភាព។

➤ **GitLab** មានមុខងារដូចជា៖

- Version Control System (VCS)
- Collaboration and Code Review
- Project Management
- Security and Compliance
- Collaboration
- Deployment Options
- Monitoring and Analysis





http://antkh.com/



## Basic GitLab Workflow



➤ **GitLab** គឺជាមានដំណើរការដូច ទៅនឹង GitHub ដែលទាំងការ Create Repository, Clone Repository, Make Changes and Push Changes ជាដើម។



What is Git?

Git Download

Basic Git  
Commands

Web-Based  
Platforms

What is GitHub?

Basic GitHub  
Workflow

What is GitLab?

Different between  
GitHub & GitLab