

git - Part II

Branching & Remotes

Stefan Pranger

stefan.pranger@student.tugraz.at

October 24, 2019

Overview

- 1 Remote Repositories
- 2 Branches
 - Working with branches
 - Merging and merge conflicts
- 3 Stash
- 4 Reflog
- 5 Remotes
- 6 Summary

Road Map

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

- 22. 10.: Local Repository and git basics
- *Today*: Branches, Remotes, non-linear workflow and working in a team
- 29. 10.: Miscellaneous: Tagging, Reflog, .gitconfig, Best Practices and Workflows, Hooks, Background, etc. . .

Remote Repositories

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

```
$> git pull origin master
```

Remote Repositories

Remote
Repositories

Branches

Stash

Reflog

Remotes

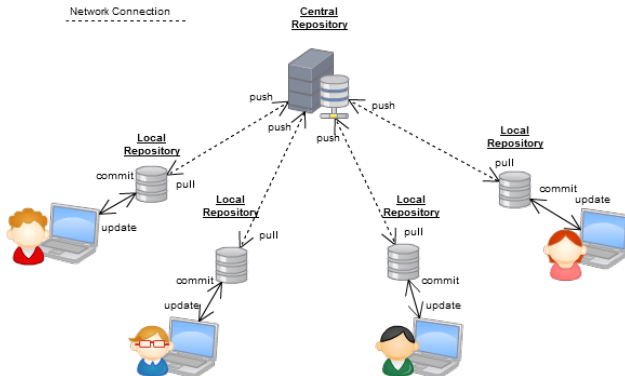
Summary

Remote

```
$> git pull origin master
```

Branch

Remote Repository



```
$> git pull origin master
```

Recap

Remote
Repositories

Branches

Working with
branches

Merging and merge
conflicts

Stash

Reflog

Remotes

Summary

- Local Repository
- Stages
- Commits
i.e. one entry in the history of your project



Untracked



Modified



Staged



Committed

Recap

Remote
Repositories

Branches

Working with
branches

Merging and merge
conflicts

Stash

Reflog

Remotes

Summary

- Local Repository
- Stages
- Commits
i.e. one entry in the *history* of your project



Untracked



Modified



Staged



Committed

Recap

- Local Repository
- Stages
- Commits

i.e. one entry in the *history* of your project

Quoting Wikipedia:

... support for *distributed, non-linear workflows*.



Untracked



Modified



Staged



Committed

What is a branch?

Remote
Repositories

Branches

Working with
branches

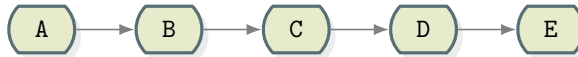
Merging and merge
conflicts

Stash

Reflog

Remotes

Summary



What is a branch?

Remote
Repositories

Branches

Working with
branches

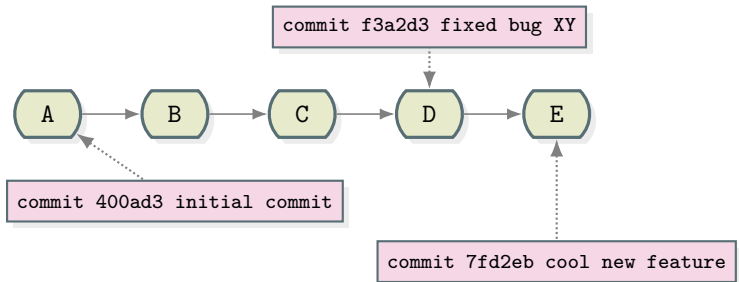
Merging and merge
conflicts

Stash

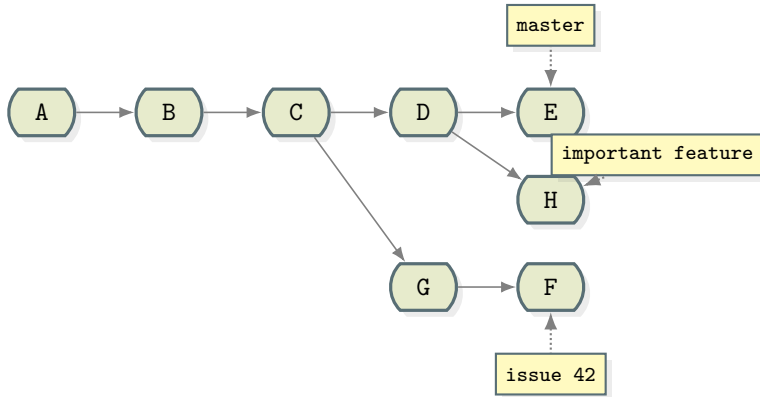
Reflog

Remotes

Summary



What is a branch?



Remote
Repositories

Branches

Working with
branches

Merging and merge
conflicts

Stash

Reflog

Remotes

Summary

What is a branch?

Remote
Repositories

Branches

Working with
branches

Merging and merge
conflicts

Stash

Reflog

Remotes

Summary

- A new working directory
- with the same version of files
- and its own set of commits

What is a branch?

Remote
Repositories

Branches

Working with
branches

Merging and merge
conflicts

Stash

Reflog

Remotes

Summary

- A new working directory
- with the same version of files
- and its own set of commits

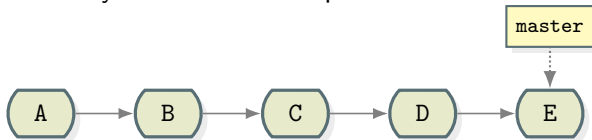
Allows you to apply changes only within the new working directory!

What is a branch?

- A new working directory
- with the same version of files
- and its own set of commits

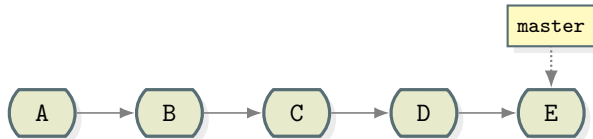
Allows you to apply changes only within the new working directory!

- internally a reference to a particular commit



Branching

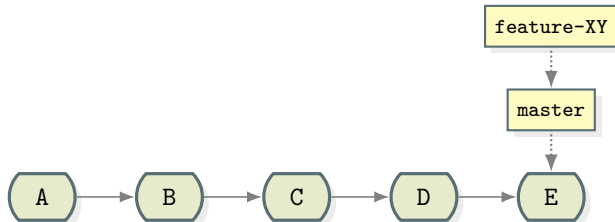
Creating a new branch:



Branching

Creating a new branch:

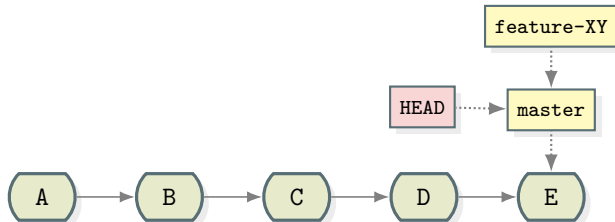
- `$> git branch <branchname>`



Branching

Creating a new branch:

- `$> git branch <branchname>`



Branching

Remote
Repositories

Branches

Working with
branches

Merging and merge
conflicts

Stash

Reflog

Remotes

Summary

Creating a new branch:

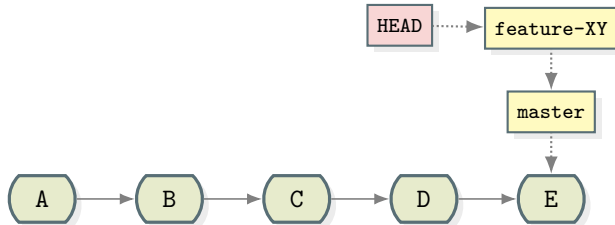
- `$> git branch <branchname>`

Switching branches:

- `$> git checkout feature-XY`, `$> git checkout master`

Creating and switching:

- `$> git checkout -b feature-XY`



Branching

Remote
Repositories

Branches

Working with
branches

Merging and merge
conflicts

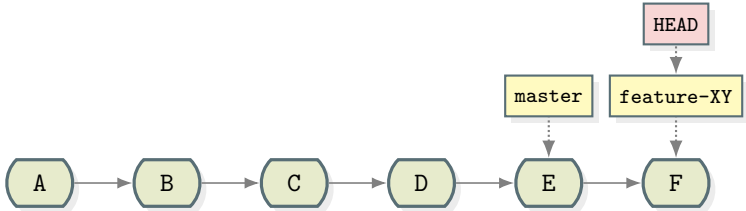
Stash

Reflog

Remotes

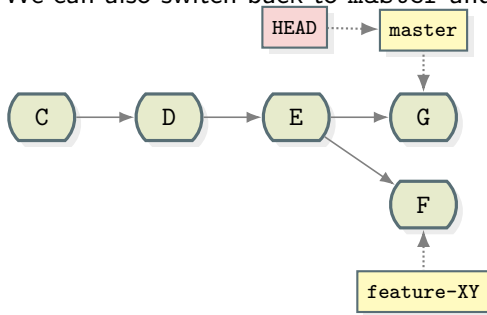
Summary

After one `$> git commit -m "<message>"`:



Branching

We can also switch back to master and commit there:



Branching

- Comparing the logs
Two "different" histories!

Remote
Repositories

Branches

Working with
branches

Merging and merge
conflicts

Stash

Reflog

Remotes

Summary

Branching

- Comparing the logs
Two "different" histories!
- We can compare these histories with

```
$> git diff <branch1> <branch2>
```

Remote
Repositories

Branches

Working with
branches

Merging and merge
conflicts

Stash

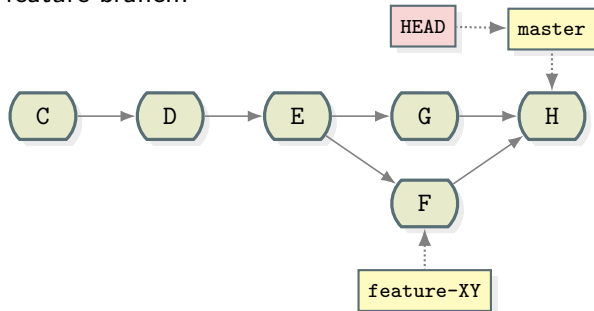
Reflog

Remotes

Summary

Branching

- Comparing the logs
Two "different" histories!
- We can compare these histories with
`$> git diff <branch1> <branch2>`
- master needs to integrate the information from the feature branch!



What does a merge do?

Remote
Repositories

Branches

Working with
branches

Merging and merge
conflicts

Stash

Reflog

Remotes

Summary

- Integrates commits from one branch into another

What does a merge do?

Remote
Repositories

Branches

Working with
branches

Merging and merge
conflicts

Stash

Reflog

Remotes

Summary

- Integrates commits from one branch into another
- `$> git merge <branchname>`

What does a merge do?

Remote
Repositories

Branches

Working with
branches

Merging and merge
conflicts

Stash

Reflog

Remotes

Summary

- Integrates commits from one branch into another
- `$> git merge <branchname>`
- Final step when implementing a new feature.

What does a merge do?

Remote
Repositories

Branches

Working with
branches

Merging and merge
conflicts

Stash

Reflog

Remotes

Summary

- Integrates commits from one branch into another
- `$> git merge <branchname>`
- Final step when implementing a new feature.
- Biggest source for problems (*merge conflicts*) and confusion.
 - different strategies, workflows, etc.

Merging

- We are done with implementing feature-XY

Remote
Repositories

Branches

Working with
branches

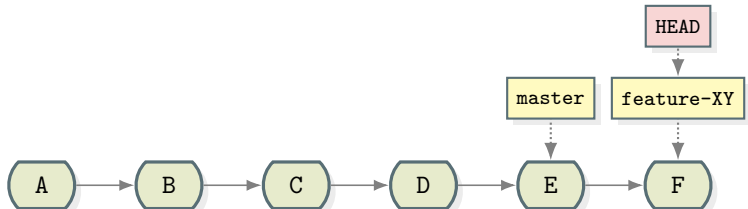
Merging and merge
conflicts

Stash

Reflog

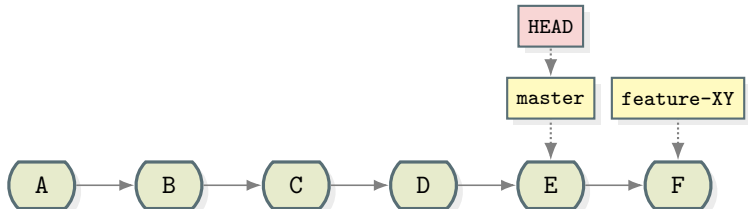
Remotes

Summary



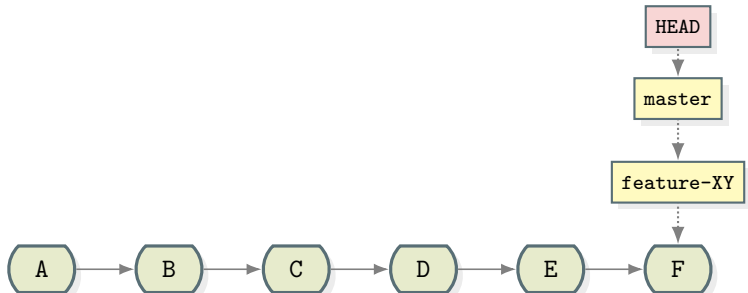
Merging

- We are done with implementing feature-XY
- `$> git checkout master`



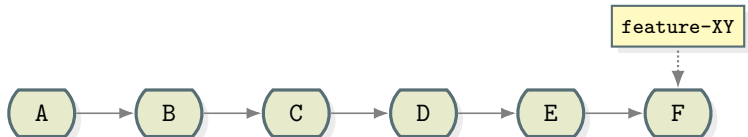
Merging

- We are done with implementing feature-XY
- `$> git checkout master`
- `$> git merge feature-XY`



Merging

- We are done with implementing feature-XY
- `$> git checkout master`
- `$> git merge feature-XY`
- No changes to your files
- Only a change of the branch reference



Merging - True Merge

- Actual merge of different content

Remote
Repositories

Branches

Working with
branches

Merging and merge
conflicts

Stash

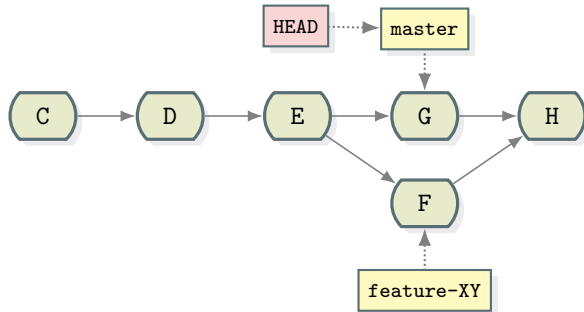
Reflog

Remotes

Summary

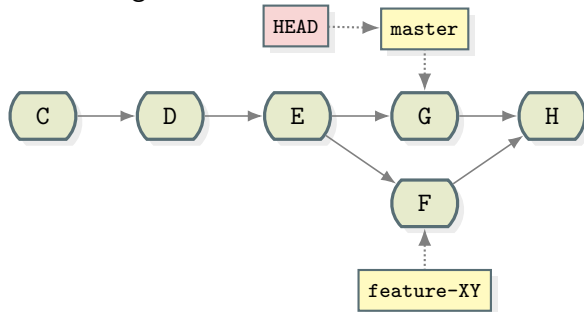
Merging - True Merge

■ Actual merge of different content



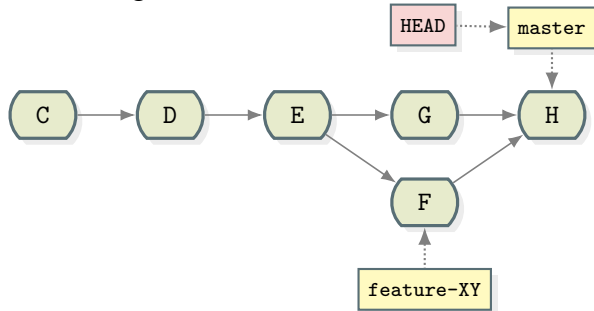
Merging - True Merge

- Actual merge of different content
- If everything runs smoothly, git will patch the new content together in a new commit



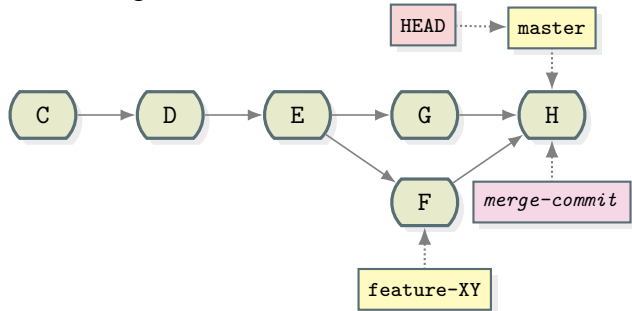
Merging - True Merge

- Actual merge of different content
- If everything runs smoothly, git will patch the new content together in a new commit



Merging - True Merge

- Actual merge of different content
- If everything runs smoothly, git will patch the new content together in a new commit



Merge Conflicts

- git branches allow you to work on different versions of the same code.

This entails that conflicts between these versions may occur!

Remote
Repositories

Branches

Working with
branches

Merging and merge
conflicts

Stash

Reflog

Remotes

Summary

Merge Conflicts

- git branches allow you to work on different versions of the same code.

This entails that conflicts between these versions may occur!

- git cannot understand the underlying logic of the changes.

You have to resolve them yourself.

Remote
Repositories

Branches

Working with
branches

Merging and merge
conflicts

Stash

Reflog

Remotes

Summary

Merge Conflicts

Remote
Repositories

Branches

Working with
branches

Merging and merge
conflicts

Stash

Reflog

Remotes

Summary

- git branches allow you to work on different versions of the same code.

This entails that conflicts between these versions may occur!

- git cannot understand the underlying logic of the changes.

You have to resolve them yourself.

- In case of a three-way merge, you have three different versions:
 - 1 Yours (`_LOCAL`)
 - 2 Theirs (`_REMOTE`)
 - 3 The common ancestor (`_BASE`)
 - 4 *a backup* (`_BACKUP`)

Merge Conflicts

Remote
Repositories

Branches

Working with
branches

Merging and merge
conflicts

Stash

Reflog

Remotes

Summary

```
<<<<<< HEAD
many many more here
=====
so many conflicts here
>>>>>> conflictBranch
```

- Conflicting hunk in a file

Merge Conflicts

Remote
Repositories

Branches

Working with
branches

Merging and merge
conflicts

Stash

Reflog

Remotes

Summary

```
<<<<<< HEAD
many many more here
=====
so many conflicts here
>>>>>> conflictBranch
```

- Conflicting hunk in a file
- Shows you the changes from HEAD and the REMOTE branch

Merge Conflicts

Remote
Repositories

Branches

Working with
branches

Merging and merge
conflicts

Stash

Reflog

Remotes

Summary

- After you have resolved all conflicts: `$> git commit`

```
Merge branch 'conflictBranch'
```

```
# Conflicts:
```

```
#         foobar
```

```
#
```

```
# It looks like you may be committing a merge.
```

Stash

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

- Remember: git only "knows" about changes that you have committed, but git knows when you have changes, which you have not yet committed.

Stash

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

- Remember: git only "knows" about changes that you have committed, but git knows when you have changes, which you have not yet committed.
- Would be overwritten on a `$> git checkout <branch> .`

Stash

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

- Remember: git only "knows" about changes that you have committed, but git knows when you have changes, which you have not yet committed.
- Would be overwritten on a `$> git checkout <branch>`.
- `$> git stash` gives us the option to stash away changes and recover them later.

Stash

- What happens when you stash changes?
- `$> git stash` saves records of staged and unstaged changed on tracked files and reverts your working directory to HEAD

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

Stash

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

- What happens when you stash changes?
- `$> git stash` saves records of staged and unstaged changed on tracked files and reverts your working directory to HEAD
- HEAD is definitely in a clean state → you can switch branches!
- After you have finished working on the other branch, you can switch to the old branch and recover changes with `$> git stash apply`

Stash

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

- What happens when you stash changes?
- `$> git stash` saves records of staged and unstaged changed on tracked files and reverts your working directory to HEAD
- HEAD is definitely in a clean state → you can switch branches!
- After you have finished working on the other branch, you can switch to the old branch and recover changes with `$> git stash apply`
- This will recover the changes stashed last.

Multiple Stashes

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

- `$> git stash` and `$> git stash apply` are not limited to one stash!

Multiple Stashes

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

- `$> git stash` and `$> git stash apply` are not limited to one stash!
- The stash works a bit like a stack (**F**irst **I**n - **F**irst **O**ut)
- You can recover any saved stash by listing them:
`$> git stash list` and passing the stash number:
`$> git stash apply stash@{n}`

Stash

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

- You can also spawn a branch from a stash via

```
$> git stash branch <branchname> stash@{n} .
```
- Save the stash via

```
$> git stash -u
```

 to also include untracked files!

Stash

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

- You can also spawn a branch from a stash via

```
$> git stash branch <branchname> stash@{n} .
```
- Save the stash via `$> git stash -u` to also include untracked files!
- Very useful if you have accidentally started working on the wrong branch!

Reflog

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

- Tracks whenever any of your local references change.
- Commits, checkouts, merges, etc.

Reflog

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

- Tracks whenever any of your local references change.
- Commits, checkouts, merges, etc.
- Can be seen as the undo-history of your local repository.
 - Even if you `$> git reset` to an older commit, you can still undo this with the reflog.

Remote Repositories

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

- Versions of your project hosted somewhere on the Internet, network or elsewhere.

Remote Repositories

Remote
Repositories

Branches

Stash

Reflow

Remotes

Summary

- Versions of your project hosted somewhere on the Internet, network or elsewhere.
- This is where `$> git pull` from and `$> git push` to!

Remotes

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

- If you have `$> git clone` d your first repository, git added origin as your first remote.

Remotes

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

- If you have `$> git clone` d your first repository, git added origin as your first remote.
- `$> git remote -v` gives you a list of remotes.

Remotes

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

What happens when you `$> git push` / `$> git pull` ?

Remotes

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

What happens when you `$> git push` / `$> git pull`?

- 1 As soon as your work is in a state in which you want to share it with other you push it via
`$> git push <remote> <branch>`

Remotes

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

What happens when you `$> git push` / `$> git pull` ?

- 1 As soon as your work is in a state in which you want to share it with other you push it via
`$> git push <remote> <branch>`
 - This will fail if there are changes on the remote branch which you have not incorporated yet → You would cause a merge conflict on the remote server!

Remotes

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

What happens when you `$> git push` / `$> git pull` ?

Remotes

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

What happens when you `$> git push` / `$> git pull` ?

- 1 You can get all new changes and branches from the remote via `$> git fetch <remote>`

Remotes

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

What happens when you `$> git push` / `$> git pull` ?

- 1 You can get all new changes and branches from the remote via `$> git fetch <remote>`
 - This only downloads those changes, no direct changes to your files!

Remotes

What happens when you `$> git push` / `$> git pull` ?

1 You can get all new changes and branches from the remote via `$> git fetch <remote>`

- This only downloads those changes, no direct changes to your files!
- You can then `$> git merge <remote>/<branch>` the changes into your local branch.
- Usually you just want to get changes from the remote master, or any other branch, and merge them. `$> git pull remote master` is doing exactly that.

Remotes

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

- If you want to work with multiple remotes you can add them via

```
$> git remote add <shortname> <git-url>
```
- This lets you push your work to different remote repositories!

Remotes

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

- If you want to work with multiple remotes you can add them via

```
$> git remote add <shortname> <git-url>
```
- This lets you push your work to different remote repositories!
- Can be useful if you want to work both with a private and a public repository.

Remotes

Remote
Repositories

Branches

Stash

Reflog

Remotes

Summary

- If you want to work with multiple remotes you can add them via

```
$> git remote add <shortname> <git-url>
```
- This lets you push your work to different remote repositories!
- Can be useful if you want to work both with a private and a public repository.
- ```
$> git remote show <shortname>
```

 gives you useful information about the remote.

# Summary

Remote  
Repositories

Branches

Stash

Reflog

Remotes

Summary

- What is a non-linear workflow?
- What is a branch?
- What is a git reference?
- What does a merge do?
- What is a merge-conflict?
- What is a stash?
- What is the reflog?
- What are remotes?