



# git

## Basics

Ondřej Sucharda  
sucharda@cngroup.dk

# Agenda

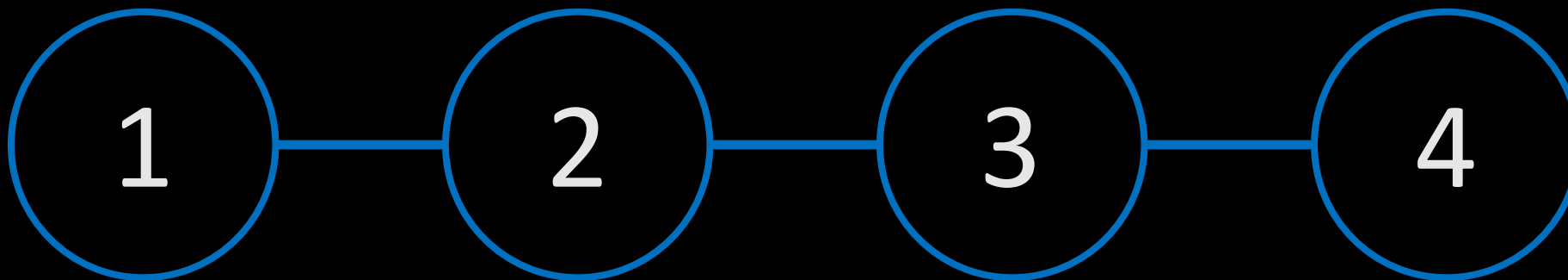
- Git intro
- Git vs GitHub
- How Git works ?
- Examples
- How to use Git ?
- Questions

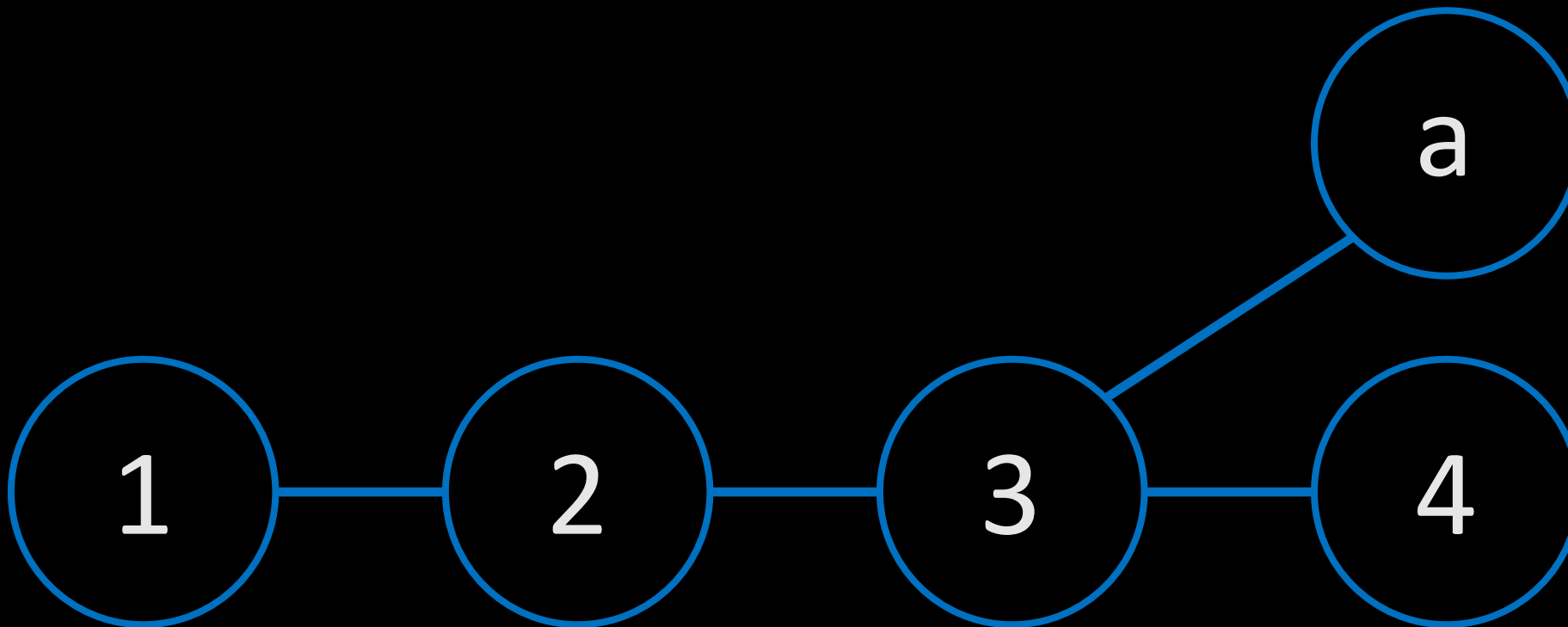
# Git

- Version control system (VCS)
- The most popular

# VCS ?

System that helps us track and manage changes in files



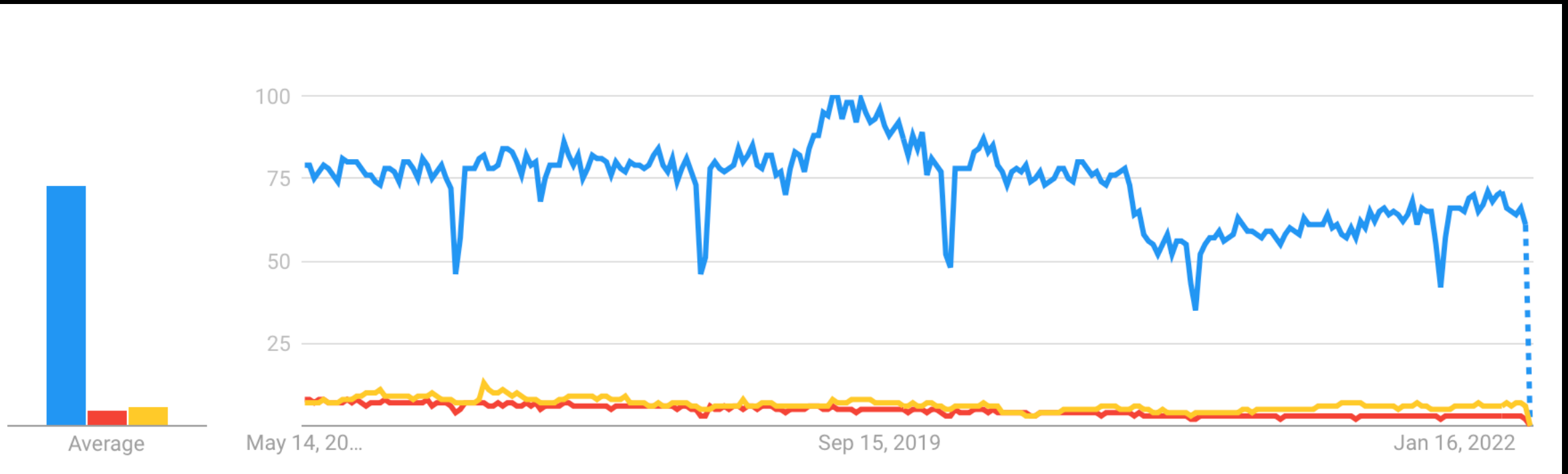


# Git

- Code history and backup
- Share code in team

# Why Git ?

● git ● svn ● mercurial



<https://trends.google.com/trends/explore?date=today%205-y&q=git,svn,mercurial>



# Git vs GitHub



Git

Running on your pc  
No registration  
No internet connection



GitHub

Repository manager  
Sharing your git repo in cloud  
Required GitHub account

# Git vs GitHub



Git

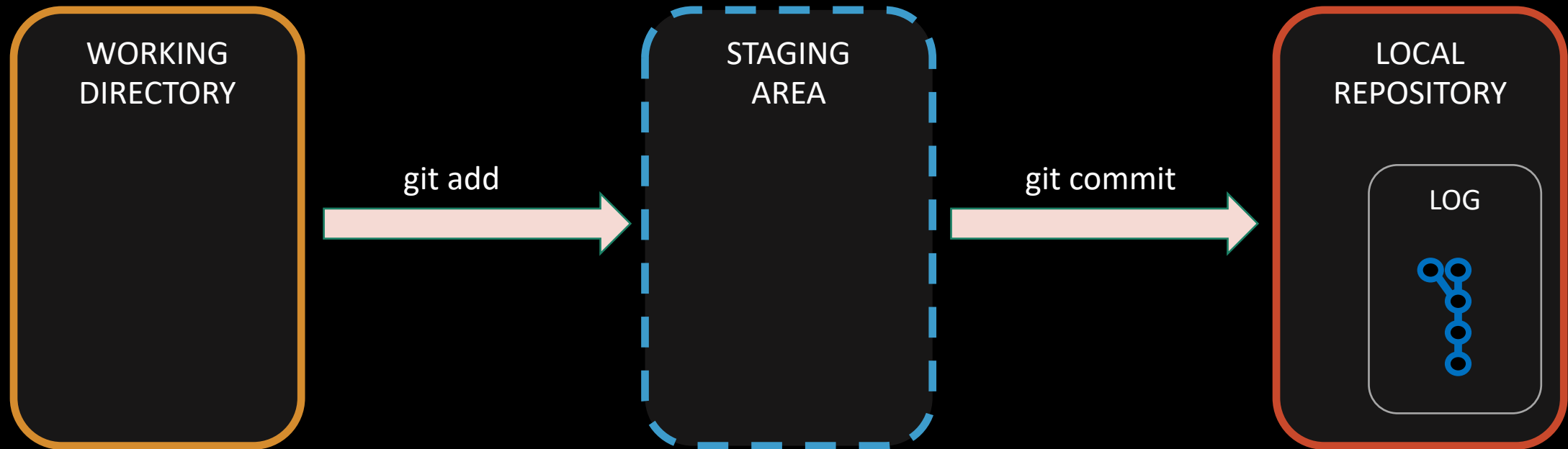


GitHub



# How Git works ?

## “First commit”



# Prepare repo

```
~/Desktop/repositories/git-training ➤ ls -la
total 8
drwxr-xr-x  3 sucharda  staff   96 May 11 09:21 .
drwxr-xr-x  8 sucharda  staff  256 May 10 23:04 ..
-rw-r--r--  1 sucharda  staff  270 May 11 00:11 index.html
~/Desktop/repositories/git-training ➤ git init
Initialized empty Git repository in /Users/sucharda/Desktop/repositories/git-training/.git/
~/Desktop/repositories/git-training ➤ ↗ master ➤ ls -la
total 8
drwxr-xr-x  4 sucharda  staff  128 May 11 09:21 .
drwxr-xr-x  8 sucharda  staff  256 May 10 23:04 ..
drwxr-xr-x  9 sucharda  staff  288 May 11 09:21 .git
-rw-r--r--  1 sucharda  staff  270 May 11 00:11 index.html
~/Desktop/repositories/git-training ➤ ↗ master ➤
```

# First commit

```
~/Desktop/repositories/git-training > master git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html

nothing added to commit but untracked files present (use "git add" to track)
~/Desktop/repositories/git-training > master git add index.html
~/Desktop/repositories/git-training > master + git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html

~/Desktop/repositories/git-training > master + git commit -m "Created base HTML structure"
[master (root-commit) 2ec47e0] Created base HTML structure
1 file changed, 10 insertions(+)
create mode 100644 index.html
~/Desktop/repositories/git-training > master
```

# Update file

```
~/Desktop/repositories/git-training ⌕ master ± git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
~/Desktop/repositories/git-training ⌕ master ± git diff

diff --git a/index.html b/index.html
index ee86e94..a98a401 100644
--- a/index.html
+++ b/index.html
@@ -6,5 +6,9 @@
     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
     <title>Document</title>
   </head>
-  <body></body>
+  <body>
+    <header>
+      <h1>My awesome title</h1>
+    </header>
+  </body>
</html>
```

# Update file

```
~/Desktop/repositories/git-training > ↗ master ± git add index.html
~/Desktop/repositories/git-training > ↗ master + git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html

~/Desktop/repositories/git-training > ↗ master + git commit -m "Added header to index.html"
[master 7020d05] Added header to index.html
1 file changed, 5 insertions(+), 1 deletion(-)
~/Desktop/repositories/git-training > ↗ master
```

# Show log (history)

```
~/Desktop/repositories/git-training ➤ master ➤ git log
commit f446921922626a56041fc5a685ee761d089a010f (HEAD → master)
Author: Ondřej Sucharda <sucharda.ondra@gmail.com>
Date: Tue May 10 23:31:21 2022 +0200

    Added header to index.html

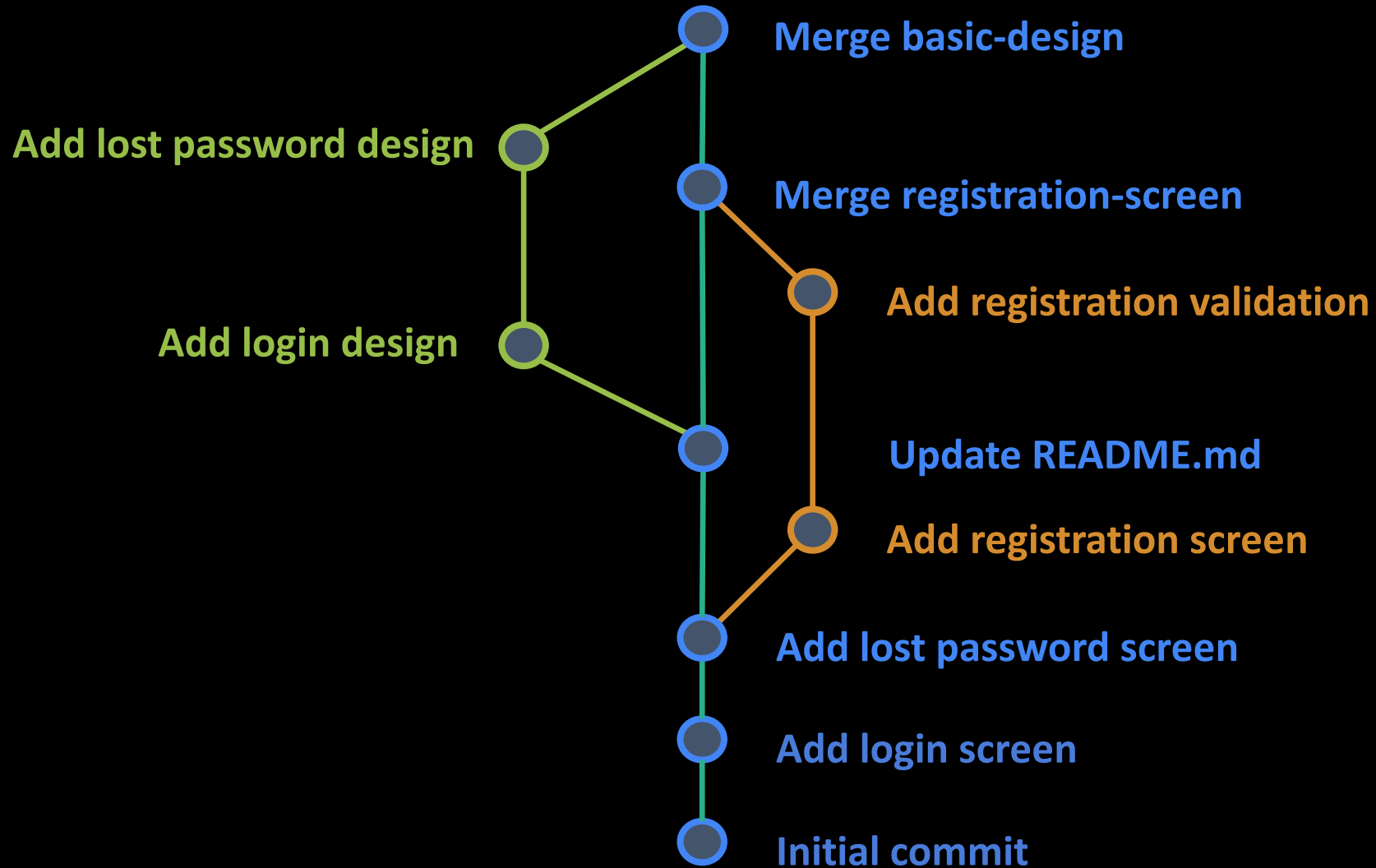
commit 2ec47e0e9eeefce2f41b8776897d7a4463fbb4a1
Author: Ondřej Sucharda <sucharda.ondra@gmail.com>
Date: Tue May 10 23:10:35 2022 +0200

    Created base HTML structure
```

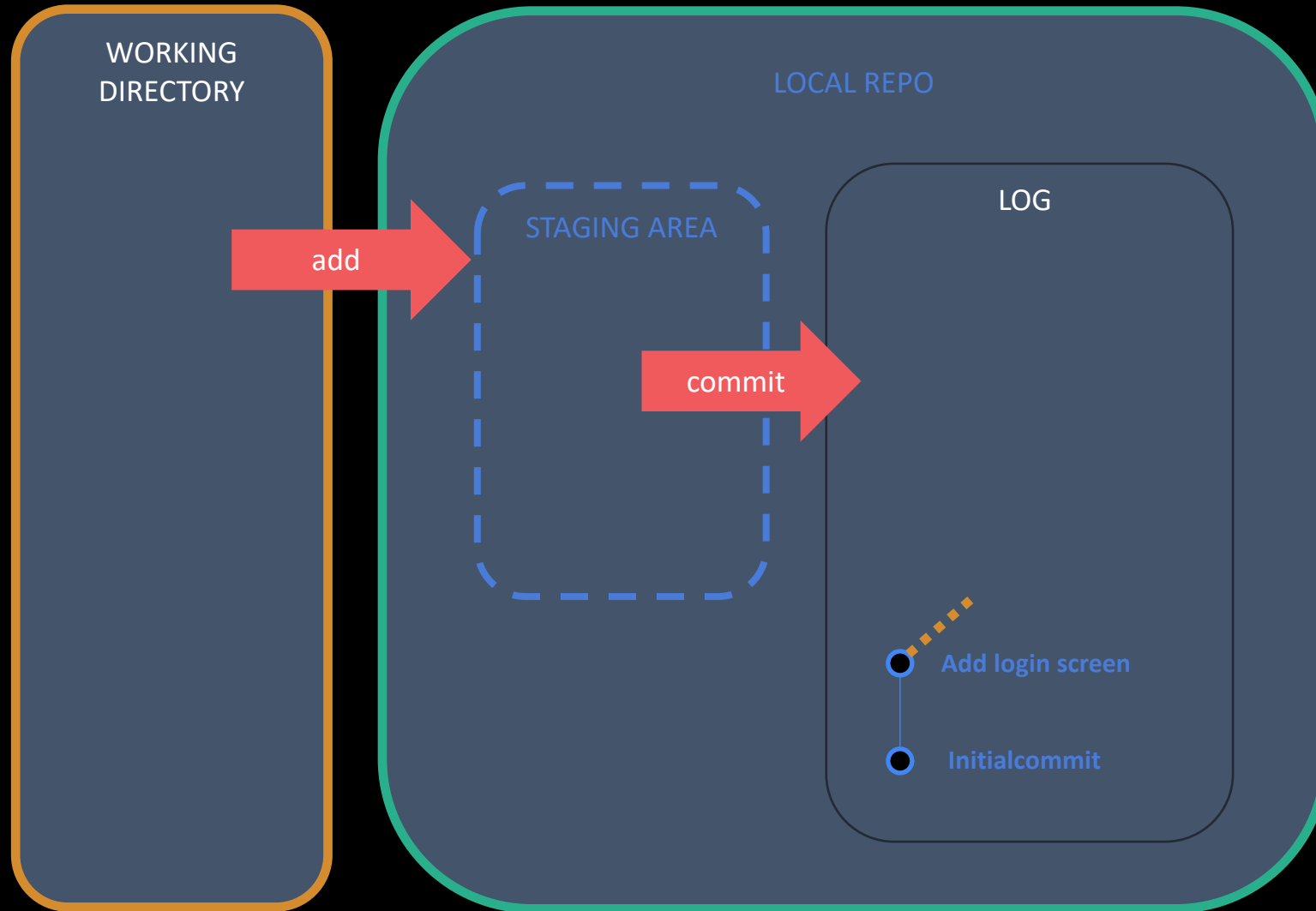
```
~/Desktop/repositories/git-training ➤ master ➤ git log --oneline
f446921 (HEAD → master) Added header to index.html
2ec47e0 Created base HTML structure
```



# Branches



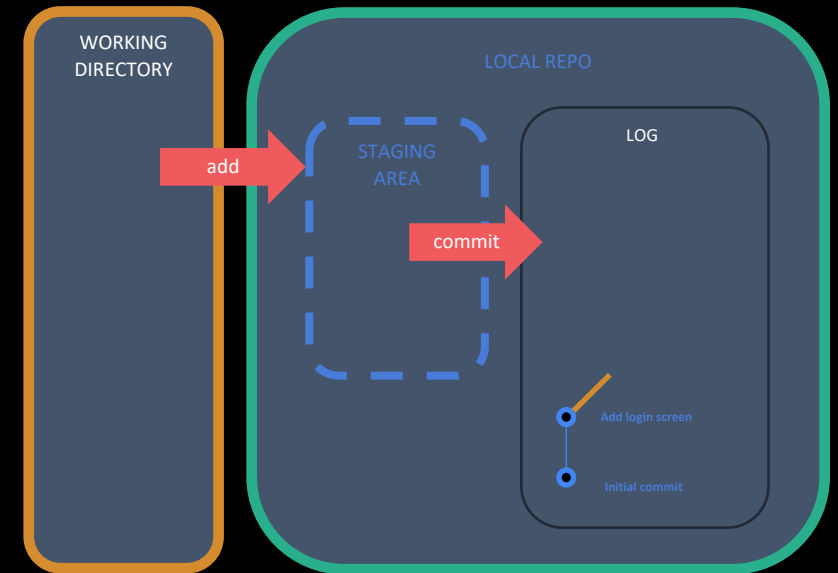
# Branches



# Branches

Create branch:

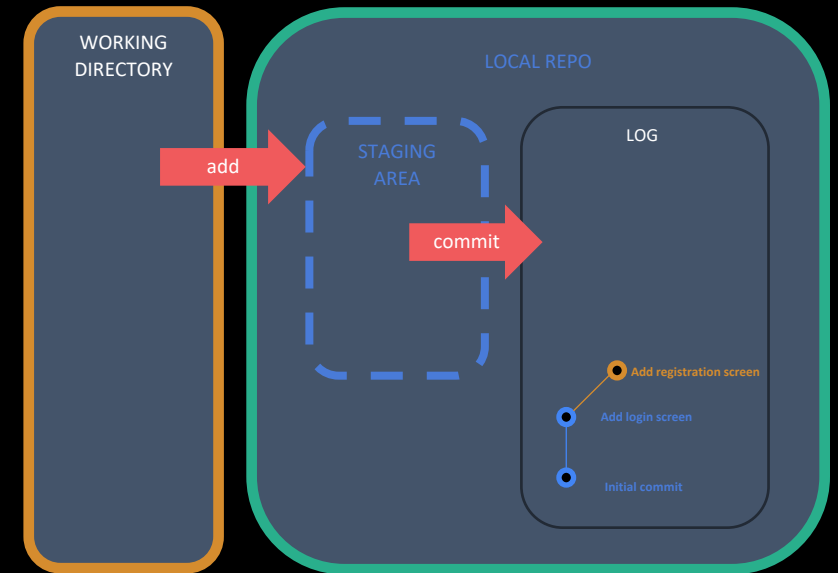
```
~/repositories/git-training ➤ master ➤ git checkout -b feature/registration  
Switched to a new branch 'feature/registration'  
~/repositories/git-training ➤ feature/registration
```



# Branches

Commit to branch:

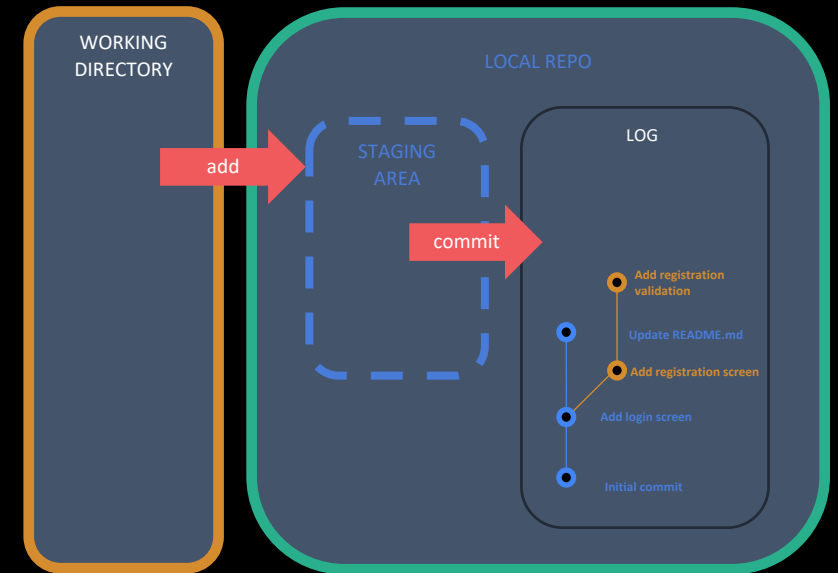
```
~/repositories/git-training ➤ feature/registration touch Registration.js
~/repositories/git-training ➤ feature/registration git add .
~/repositories/git-training ➤ feature/registration + git commit -m "Add registration screen"
[feature/registration 045e7e6] Add registration screen
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Registration.js
~/repositories/git-training ➤ feature/registration
```



# Branches

Second commit to branch:

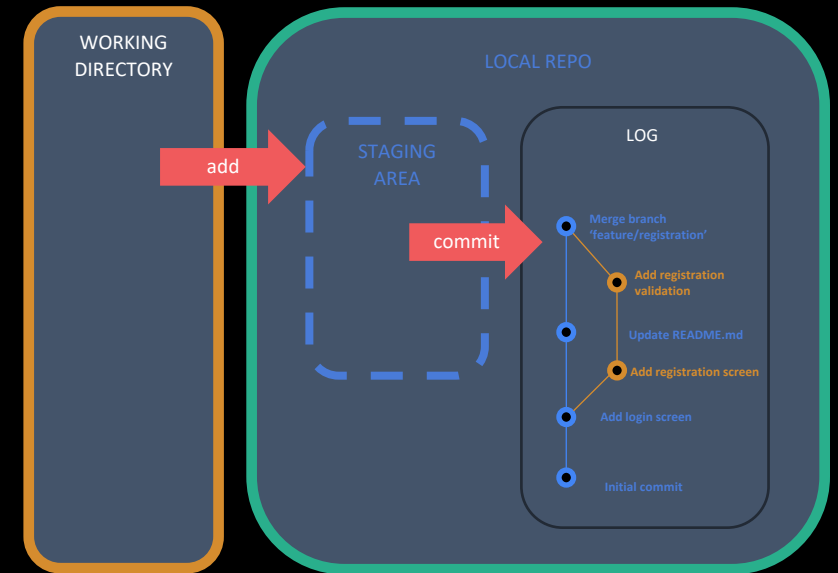
```
~/repositories/git-training  ♢ feature/registration  vim Registration.js
~/repositories/git-training  ♢ feature/registration ±  git commit -a -m "Add
registration validat
ion"
[feature/registration 14284c9] Add registration validation
 1 file changed, 3 insertions(+)
~/repositories/git-training  ♢ feature/registration  █
```



# Branches

Merge branch “feature/registration” into “master”

```
~/repositories/git-training ➤ feature/registration ➤ git checkout master  
Switched to branch 'master'  
~/repositories/git-training ➤ master ➤ git merge feature/registration  
Merge made by the 'recursive' strategy.  
Registration.js | 3 +++  
1 file changed, 3 insertions(+)  
create mode 100644 Registration.js  
~/repositories/git-training ➤ master ➤
```

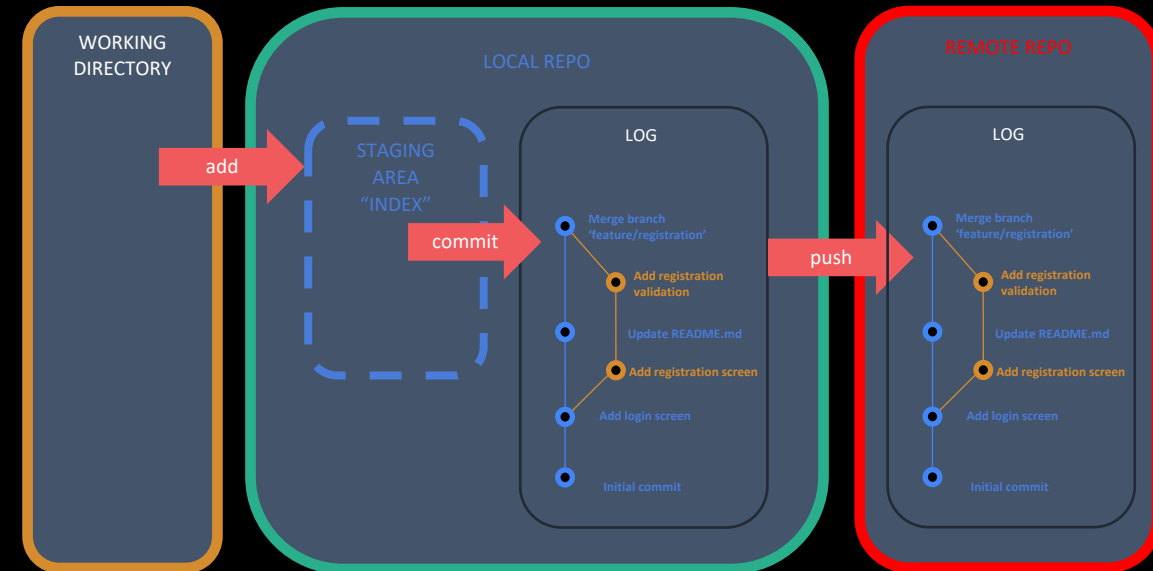


# Push to GitHub

Push to remote repo (GitHub):

1. Create repo on <https://github.com/>
2. Connect local with remote repo
3. Push

```
~/repositories/git-training ➤ master ➤ git remote add origin git
@github.com:suchardao/git-training.git
~/repositories/git-training ➤ master ➤ git push
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 12 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (15/15), 1.30 KiB | 664.00 KiB/s, done.
Total 15 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), done.
To github.com:suchardao/git-training.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'
.
```

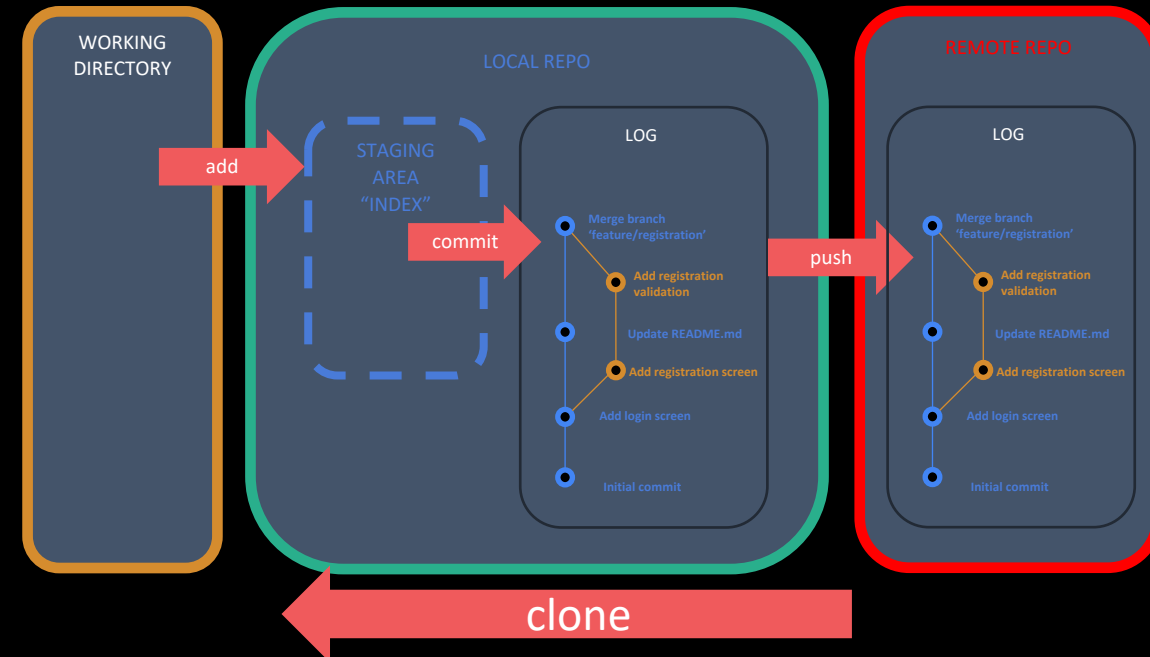


# Clone repo

Clone from remote repo (GitHub):

1. Find repo in <https://github.com/> and copy address
2. Run “git clone [address]”

```
~/Desktop/repositories ➤ git clone git@github.com:suchardao/git-training.git
Cloning into 'git-training' ...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 7 (delta 2), reused 7 (delta 2), pack-reused 0
Receiving objects: 100% (7/7), done.
Resolving deltas: 100% (2/2), done.
~/Desktop/repositories ➤ cd ./git-training
~/Desktop/repositories/git-training ➤ master ➤ ls -la
total 8
drwxr-xr-x  4 sucharda  staff  128 May 11 09:34 .
drwxr-xr-x  8 sucharda  staff  256 May 11 09:34 ..
drwxr-xr-x 12 sucharda  staff  384 May 11 09:34 .git
-rw-r--r--  1 sucharda  staff  270 May 11 09:34 index.html
~/Desktop/repositories/git-training ➤ master ➤
```

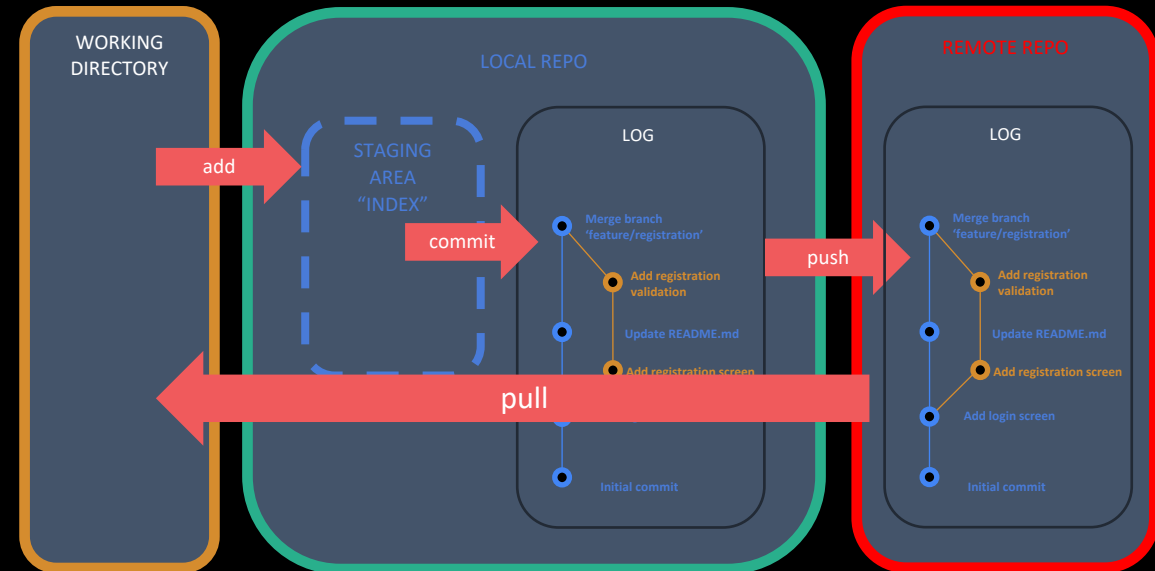


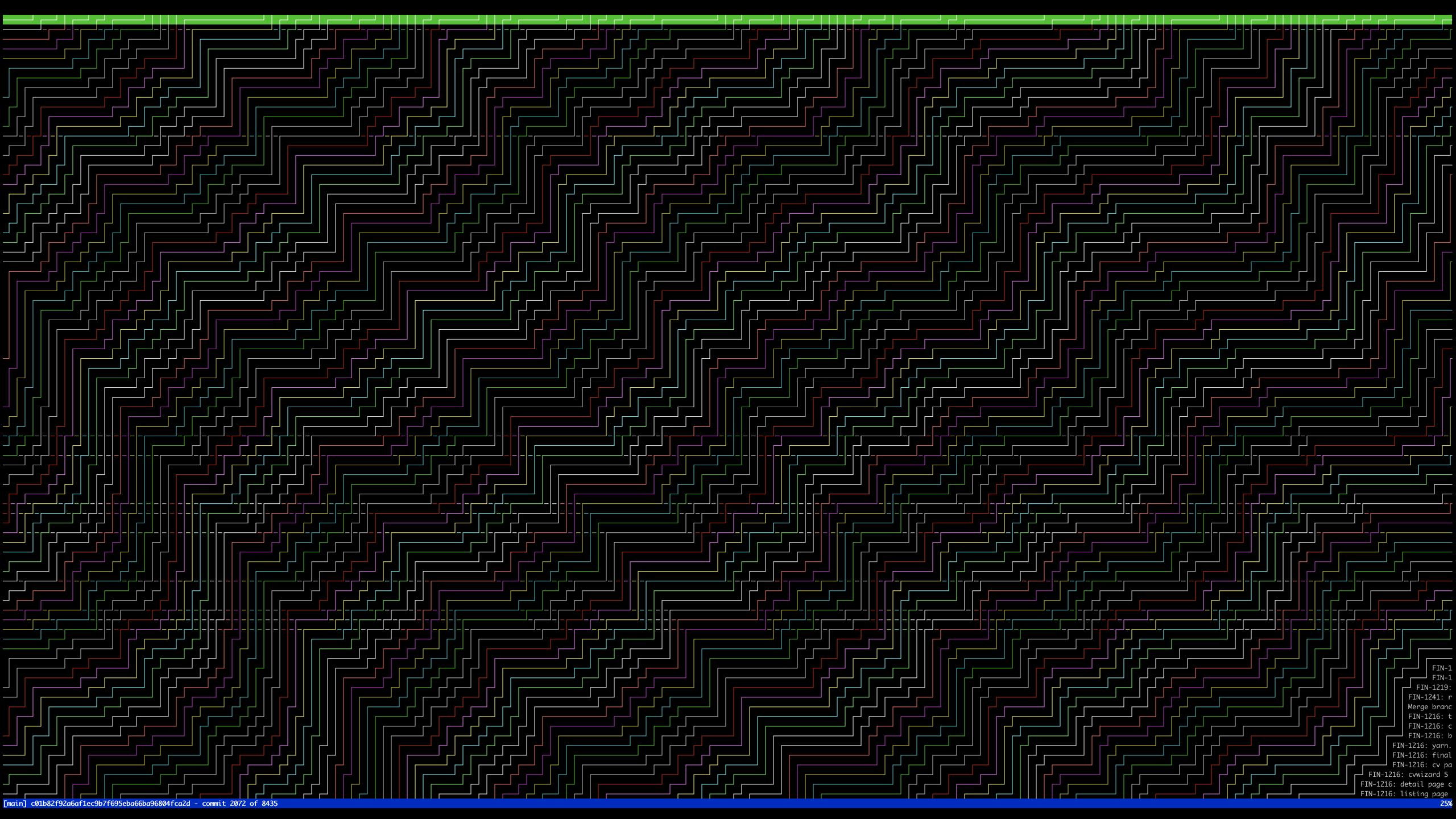


# Pull changes

Pull changes from remote repo (GitHub):

```
~/Desktop/repositories/git-training ➤ master ➤ git pull
Updating 9bd7493..9b6e02d
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
~/Desktop/repositories/git-training ➤ master ➤
```





FIN-1  
FIN-1  
FIN-1219:  
FIN-1241: r  
Merge branc  
FIN-1216: t  
FIN-1216: c  
FIN-1216: b  
FIN-1216: yarn.  
FIN-1216: finel  
FIN-1216: cv pa  
FIN-1216: cwtizard 5  
FIN-1216: detail page c  
FIN-1216: listing page

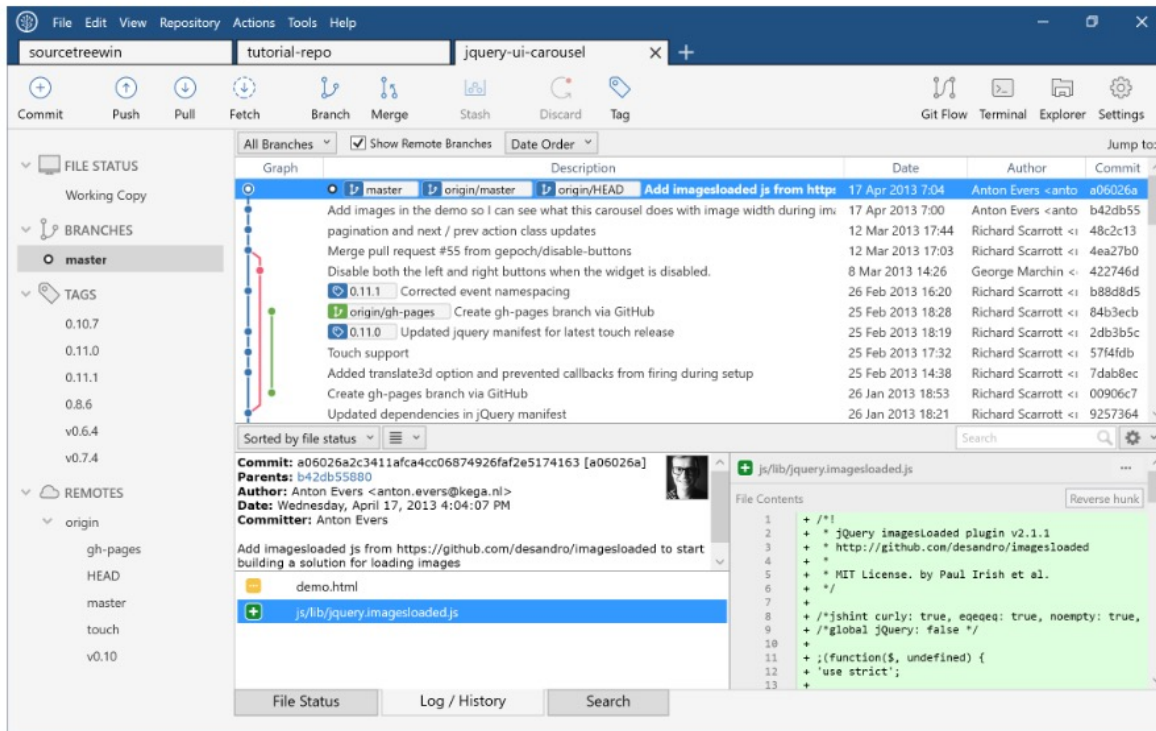
# Shits happen

FIN-1  
FIN-1  
FIN-1219:  
FIN-1241: r  
Merge branc  
FIN-1216: t  
FIN-1216: c  
FIN-1216: b  
FIN-1216: yarn.  
FIN-1216: finel  
FIN-1216: cv pa  
FIN-1216: cwtizard 5  
FIN-1216: detail page c  
FIN-1216: listing page

# How to use Git?

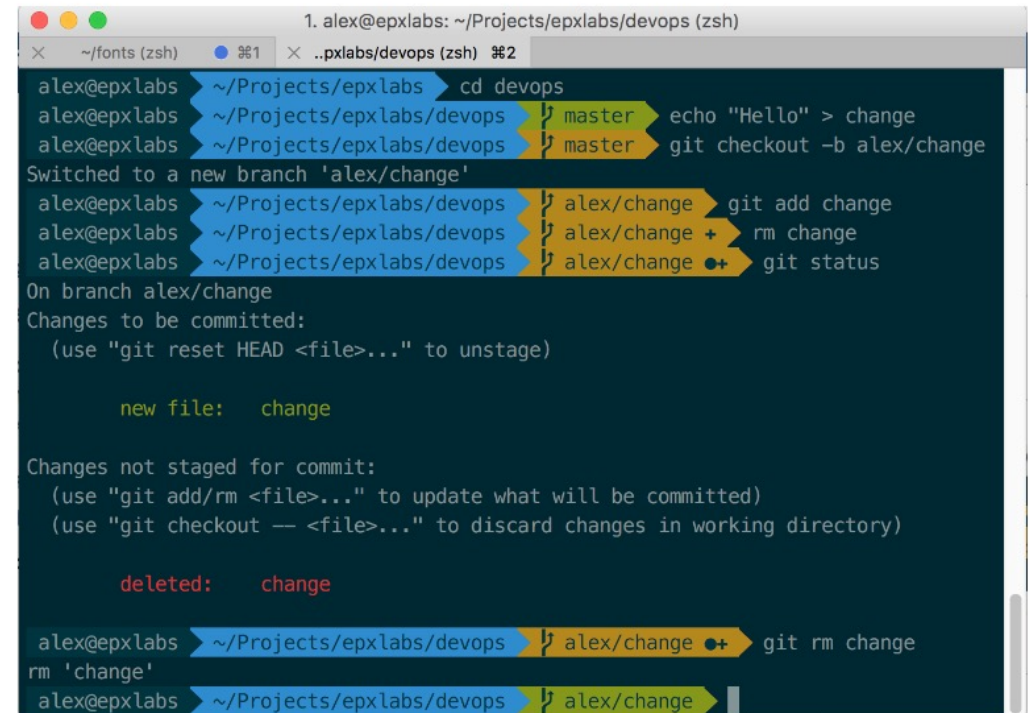
# GUI

## Graphical User Interface



# CLI

## Command-Line Interface



# TIG

- Tig: text-mode interface for Git
- Installation using Homebrew: "brew install tig"

```
~/repositories/git-training ➤ master ➤ tig
```

```
2022-05-09 21:26 +0200 Ondřej Sucharda M [master] {origin/master} Merge branch 'feature/registration'
2022-05-09 21:20 +0200 Ondřej Sucharda | o [feature/registration] Add registration validation
2022-05-09 21:16 +0200 Ondřej Sucharda | o Add registration screen
2022-05-09 21:16 +0200 Ondřej Sucharda o | Update README.md
2022-05-09 21:14 +0200 Ondřej Sucharda o | Add login screen
2022-05-09 21:13 +0200 Ondřej Sucharda I Initial commit
```

**Questions ?**

# Cheat Sheet

## Create a Repository

From scratch -- Create a new local repository

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

Download from an existing repository

```
$ git clone my_url
```

## Observe your Repository

List new or modified files not yet committed

```
$ git status
```

Show the changes to files not yet staged

```
$ git diff
```

Show the changes to staged files

```
$ git diff --cached
```

Show all staged and unstaged file changes

```
$ git diff HEAD
```

Show the changes between two commit ids

```
$ git diff commit1 commit2
```

List the change dates and authors for a file

```
$ git blame [file]
```

Show the file changes for a commit id and/or file

```
$ git show [commit]:[file]
```

Show full change history

```
$ git log
```

Show change history for file/directory including diffs

```
$ git log -p [file/directory]
```

## Working with Branches

List all local branches

```
$ git branch
```

List all branches, local and remote

```
$ git branch -av
```

Switch to a branch, my\_branch, and update working directory

```
$ git checkout my_branch
```

Create a new branch called new\_branch

```
$ git branch new_branch
```

Delete the branch called my\_branch

```
$ git branch -d my_branch
```

Merge branch\_a into branch\_b

```
$ git checkout branch_b
```

```
$ git merge branch_a
```

Tag the current commit

```
$ git tag my_tag
```

## Make a change

Stages the file, ready for commit

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

Stage all changed files, ready for commit

```
$ git add .
```

Commit all staged files to versioned history

```
$ git commit -m "commit message"
```

Commit all your tracked files to versioned history

```
$ git commit -am "commit message"
```

Unstages file, keeping the file changes

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

Revert everything to the last commit

```
$ git reset --hard
```

## Synchronize

Get the latest changes from origin (no merge)

```
$ git fetch
```

Fetch the latest changes from origin and merge

```
$ git pull
```

Fetch the latest changes from origin and rebase

```
$ git pull --rebase
```

Push local changes to the origin

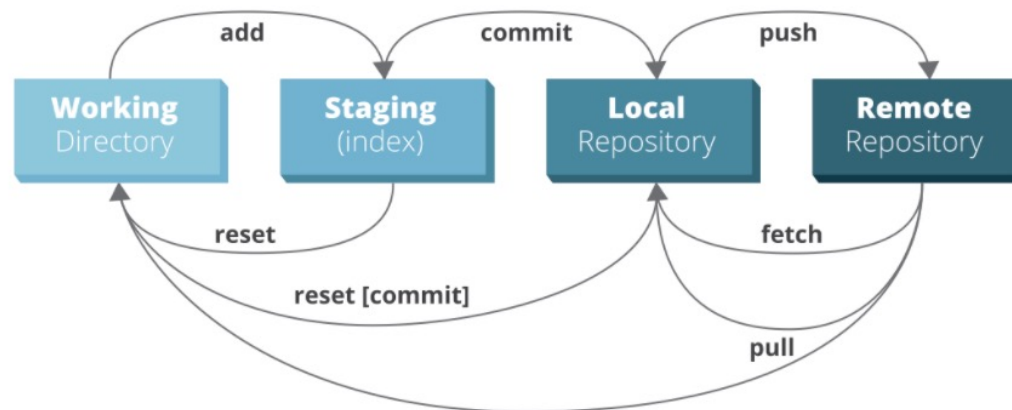
```
$ git push
```

## Finally!

When in doubt, use git help

```
$ git command --help
```

Or visit <https://training.github.com/> for official GitHub training.





# Thanks

