



Application Development with Python

DHKHDL18A

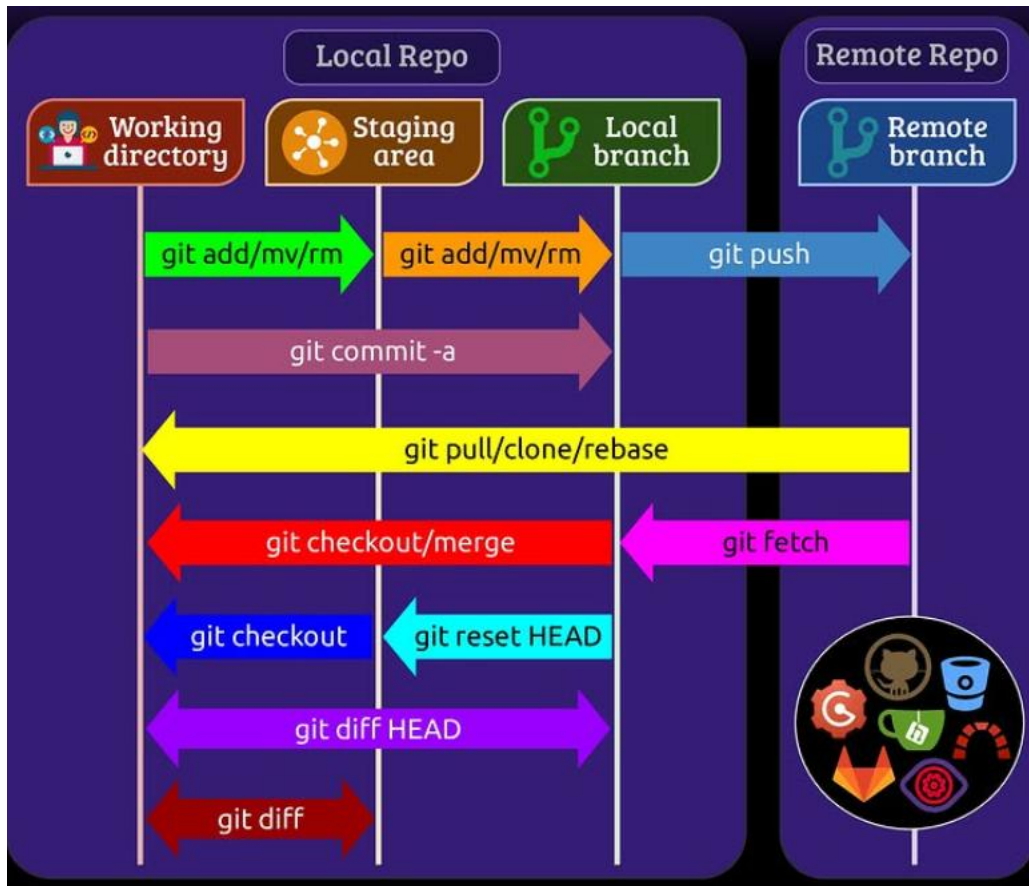
Lab 02: Git & Github

Mục tiêu

1. Cài đặt git trên máy tính và tạo tài khoản trên github
2. Thực hành một số câu lệnh git phổ biến (`git init`, `git clone`, `git add`, `git commit`, `git branch`, `git checkout`, `git status`, `git log`, `git revert`, `git branch`, `git merge`)
3. Hiểu được các quy trình làm việc phổ biến (`git common workflow`)

Nội dung

1. Cài đặt git và tạo tài khoản github
 - Tạo tài khoản trên github ([LINK](#))
 - Cài đặt git trên máy cá nhân ([LINK](#))
 - Tạo SSH key ([LINK](#))
 - Thêm SSH key vào github ([LINK](#))
2. Thực hành một số câu lệnh git phổ biến (`git init`, `git clone`, `git add`, `git commit`, `git branch`, `git checkout`, `git status`, `git log`, `git revert`, `git branch`, `git merge`)
 - Sử dụng git trên máy cá nhân ([LINK](#))
 - Tìm hiểu về git branch ([LINK](#))



([animated LINK](#))

3. Hiểu được các quy trình làm việc phổ biến (git common workflow)

[LINK](#)

Bài tập

Cơ bản

1. Khởi tạo một Git Repository

- Tạo một thư mục mới trên máy tính, sau đó khởi tạo một Git repository trong thư mục đó.
- Tạo một file README.md và commit thay đổi.

2. Thêm file vào Git

- Tạo thêm 2 file mới trong repository (ví dụ: file1.txt và file2.txt).
- Thêm các file này vào staging area bằng lệnh git add.

3. Ghi commit

- Thực hiện commit cho các file đã thêm với một thông điệp rõ ràng.
4. **Xem lịch sử commit**
- Sử dụng lệnh git log để xem lịch sử commit và thử sử dụng các tùy chọn như --oneline hoặc --graph.
5. **Tạo nhánh mới**
- Tạo một nhánh mới có tên là feature-branch.
 - Chuyển sang nhánh đó bằng lệnh git checkout hoặc git switch.
-
6. **Merge nhánh**
- Thêm một vài thay đổi trong nhánh feature-branch (ví dụ: thêm nội dung vào file1.txt).
 - Quay lại nhánh main và thực hiện merge nhánh feature-branch vào main.
7. **Giải quyết xung đột**
- Tạo xung đột bằng cách chỉnh sửa cùng một dòng trong file2.txt ở hai nhánh khác nhau.
 - Thực hiện merge và giải quyết xung đột.
8. **Sử dụng Gitignore**
- Tạo một file .gitignore và thêm một số quy tắc (ví dụ: bỏ qua file .log hoặc thư mục temp/).
 - Kiểm tra xem Git đã bỏ qua các file đó chưa.
-
9. **Làm việc với Remote Repository**
- Kết nối repository local với một remote repository (ví dụ: GitHub).
 - Đẩy các thay đổi lên remote repository và kiểm tra kết quả.
10. **Sử dụng Rebase**
- Tạo một nhánh mới từ main (ví dụ: experiment-branch).
 - Thực hiện vài commit trên cả hai nhánh main và experiment-branch.
 - Dùng lệnh git rebase để rebase experiment-branch lên main

Tham khảo

- https://dev.to/ajmal_hasan/beginner-friendly-git-workflow-for-developers-2g3g