git CLI로 간단하게 조작하기!

by 윤선지

CLI?

명령어 인터페이스 (Command Line interface)
텍스트 터미널을 통해 사용자와 컴퓨터가 상호 작용하는 방식

편한 GUI프로그램 대신 사용하는 이유?

1. GUI프로그램보다 가볍다.
   (CJO경우 보안프로그램이 설치되어있어 소스트리(GUI)실행을 버거워한다.)
2. CLI를 사용할 수 있으면 GUI를 사용하는 것은 쉽지만 그 반대는 힘들다.
3. 멋있어 보인다. 😜 (찡긋~)

CLI 프로그램을 사용하면서 불편한 점

1. 로그 확인시 불편하다.
2. diff 확인시 불편하다.
3. 체리픽, 머지시 불편하다.

CLI를 더 편하게 사용하기

다양한 포터블 콘솔 애뮬레이터 프로그램(conEmu, cmder)이 있습니다.

conEmu를 사용하면 기본 cmd창에서 제공하지 않는 기능
(탭, 디자인세팅, 단어 하이라이팅 등..) 을 설정할 수 있습니다.

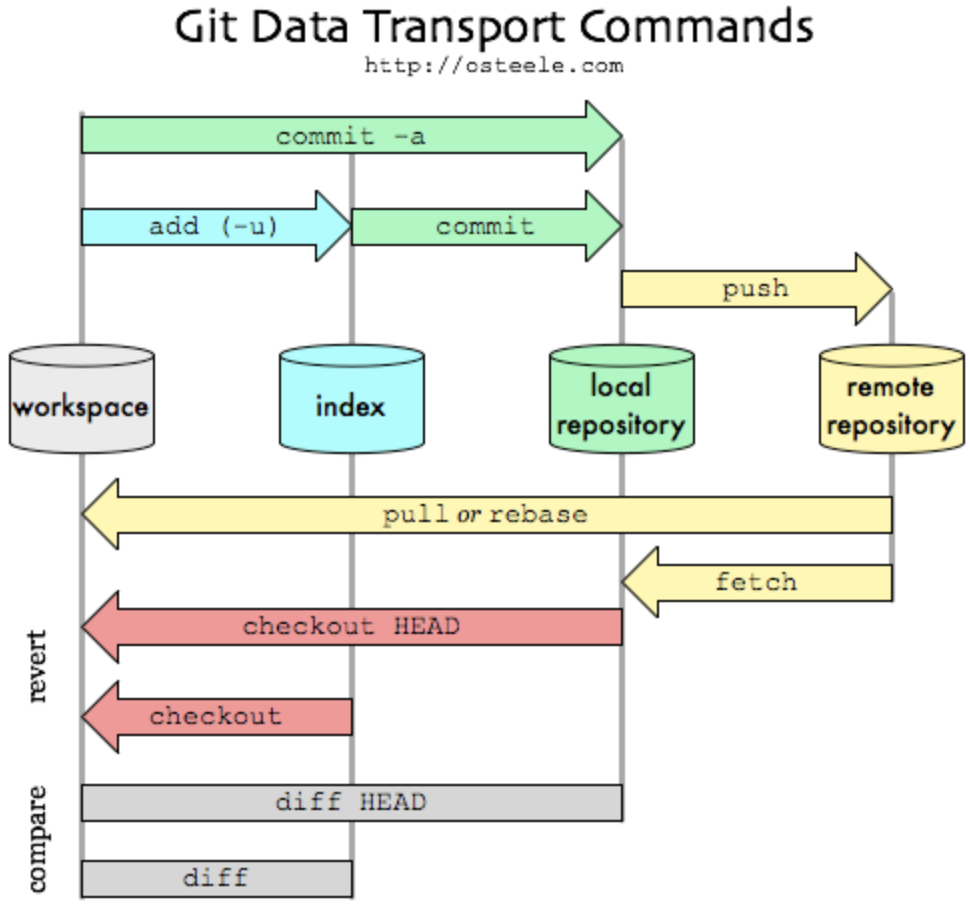
참고페이지
설치 - http://conemu.github.io/
에디터꾸미기 - http://programmingsummaries.tistory.com/352
단축키세팅 - http://commin.tistory.com/37

CLI 간단하게 조작하기

1. git clone
2. git status
4. git add .
5. git commit
3. git pull
6. git push
7. git branch
8. git checkout
9. 작업하면서 겪었던 conflict 사례와 해결방법

참고이미지

아래 이미지를 참고하시면서
내용을 확인하시면 도움이 됩니다 ~



1. git clone

리모트 저장소 복사하기

※저장소 복사 시 다음페이지의 소스를 순서대로 cli에 입력해 주시면 됩니다 😄

git init

(.git이라는 하위 디렉토리를 만든다 .git 디렉토리에는 저장소에 필요한 뼈대 파일이 들어있다.)

git config --global user.name "이름"

git config --global user.email "이메일주소"

(--global설정은 한번만 설정하면 된다)

git config --global pack.windowMemory "100m"

git config --global pack.SizeLimit "100m"

git config --global pack.threads "1"

(ex. 맥북프로 8 thread. 개인 컴터사양에 맞게 설정을 해주면 됩니다.)

git remote add origin http://git.cjmall.com/cjos-markup

git remote add [단축이름] [url]

git clone http://git.cjmall.com/cjos-markup

2. git status

현재 위치한 브랜치, 파일의 상태를 확인하기 위해 사용한다.

파일상태

1. Untracked files

(신규추가 된 파일)

2. Changes to be committed

(스테이징 상태의 수정 된 파일)

3. Changes not staged for commit

(스테이징 상태가 아닌 수정 된 파일)

3. git add

새롭게 생성한 파일, 수정한 파일, 삭제 된 파일 스냅샷을 커밋하기 위해 스테이징 상태로 변경해 주는 작업

git add .

(수정,신규,삭제 된 모든변경 점 스테이징 하는 작업)

git add 파일경로

(원하는 파일만 스테이징 하는 작업)

(ex . git add module/tvshop_program/md_171222_program.html)

스테이징파일을 언스테이징(unstage)으로 변경하기

git reset HEAD 파일경로

modified 파일을 수정이전으로 되돌리기

git checkout -- 파일경로

git checkout 파일경로

4. git commit

신규생성, 수정, 삭제한 파일 스냅샷을 커밋하는 작업

git commit -m "커밋메시지"

git commit -a -m "커밋메시지"

("git add"+"git commit")

git commit --amend

(이전 커밋 수정하기!)

5. git pull

새로운정보가 있으면 모두 내려받고
받은데이터를 로컬저장소에 업데이트한다.

fetch, pull 차이점

fetch : 중앙저장소의 소스를 로컬저장소로 가져온다. 현재작업중인 소스를 변경하는 merge작업을 하지 않는다.

pull : 중앙저장소의 소스를 로컬저장소로 가져온다. 현재작업중인 소스를 변경하는 merge작업도 한다.

(pull = fetch + merge)

6. git push

개인환경에서 작업한 내용을 리모트 저장소에 올리는 작업

7. git branch

브랜치 관련 작업

git branch

(브랜치목록으로 보여준다)

git branch issue1

(issue 1 브랜치 생성)

8. git checkout

git checkout branch

(HEAD가 가르키는 브랜치를 바꿀때)

git checkout -b issue1

(브랜치만들고 checkout도 한번에 할때)

checkout, reset의 차이점

reset

(커밋초기화)

reset --soft

(옵션 최근에 한 commit을 이전으로 돌린다. git commit --amend 명령하고 기능이 같다)

reset --mixed

(최근으로 커밋돌리고, 스테이징상태 비우기)

reset --hard

(수정하고 있던 파일을 삭제한다. 위험한 요소!!!!)

checkout

head는 움직이지 않고 index의 내용이 해당커밋버전으로 변경

워킹 디렉토리의 파일도 해당커밋버전으로 변경된다.

git reset --hard명령의 동작이랑 같다.

9. 경험했던 conflict 사례와 해결방법

1. master브랜치에서 develop 브랜치 pull 받는경우

2. 체리픽하는경우 순서가 꼬인경우

3. conflict 날때

1. master브랜치에서 develop 브랜치 pull 받는경우

해결방법 : conflict난 저장소는 삭제하고 다시 clone 받는다.
git clone 저장소
가장 깔끔하고 안전한 방법인 것 같습니다.

2. 체리픽하는경우 순서가 꼬여서 conflict 나는 경우

해결방법 : conflict 나는 파일을 열어서 conflict 수정 후
커밋하여 push 합니다.

3. conflict나면서 내가 수정하지 않은 파일도 같이 묶이는 경우

해결방법 : 이전커밋으로 돌아가는 "reset --mixed" 명령어를 사용합니다.
개인작업환경에 자신이 수정한 파일만 존재하는지 확인 후 다시 스테이징(git add)하고 이후과정을 진행합니다.

기타유용한 팁

CLI 단축키 사용

http://blog.jeonghwan.net/2016/08/16/git-alias.html

ex. pl = pull, ps = push

cli 설정확인

git config --list

(처음세팅한 옵션값, 에디터 설정을 확인할 수 있다)

확인하고 싶은 설정만 확인하기

git config user.name

편집기 설정

git config --global core.editor 에디터

diff도구

참고사이트
git에 대해서 깊게 공부하고 싶으시다면!!

https://git-scm.com/book/ko/