







- 1. 임시 저장 개요
- 2. stash, stash save
- 3. stash apply, stash pop



- 1. 임시 저장의 필요성을 이해하고 수행할 수 있다.
- 2. 임시 저장 목록을 보고 삭제할 수 있다.
- 3. 임시 저장 목록에서 특정한 목록의 내용으로 작업 디렉토리와 인덱스를 복원할 수 있다.



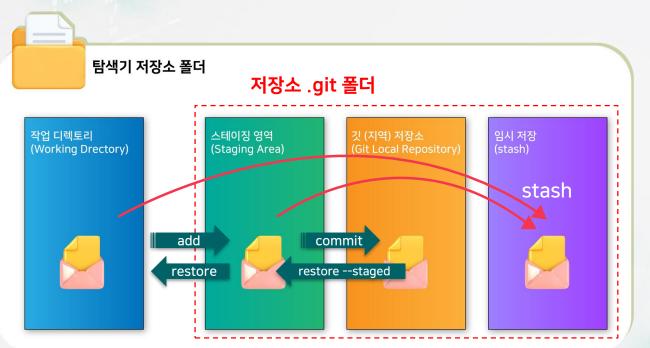




## 기4영역과임시저장개요

# 및 4 영역

✓ 깃 3 영역 + stash 영역





- 1 깃4영역과임시저장개요
- 🤷 깃 영역의 가장 단순한 상태
  - Mothing to commit, working tree is clean







## 깃4영역과임시저장개요

# Git Stash

# ₩ 사전적 의미 stash

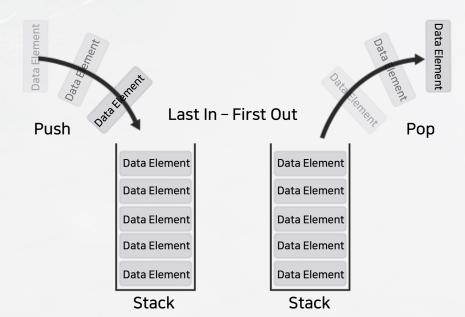
- 놓다, 남겨두다, 감추다, 안전한 곳에 숨겨두다
- 커밋할 필요 없이 파일의 변경 사항을 숨기거나 비밀리에 저장할 수 있는 강력한 도구
  - 작업 디렉토리와 스테이징 영역의 원래의 내용을 stash 스택에 저장
  - 작업 디렉토리와 스테이징 영역의 값을 깃 저장소의 마지막 커밋 내용으로 복사
- 따로 안전한 곳에 저장했다가 다시 가져올 수 있는 기능
  - 저장 내용
    - ➡ 작업 디렉토리 내용과 스테이지 내용





# ☑ 깃에서 임시저장소(스택 구조)에 저장

- Stack of git stashes
  - 가장 최근의 내용이 가장 위에 저장되는 구조









### <sup>7</sup> 깃 4 영역과임시저장개요

## 🐞 Git stash 필요성

- ☑ 브랜치 이동 또는 이전 커밋으로 이동하려면
  - \$ git switch bname
  - \$ git checkout HEAD~
  - 커밋할 게 없고, 작업 트리가 깨끗해야 함
- ★ 수정한 내용이 있거나 커밋할 게 있는 상황에서 다른 브랜치 이동 또는 이전 커밋으로 이동하려면
  - 현재 작업 공간의 수정 내용과 인덱스의 내용을 보관할 필요가 있음
    - 나중에 다시 사용하기 위해





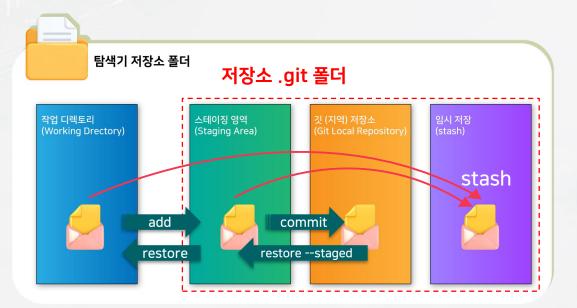


## 깃4영역과임시저장개요

## 🐡 임시저장 영역에 저장되는 내용

## ★ stash로 저장되는 내용

- 작업 디렉토리 내용과 스테이징 영역 내용이 stash에 저장되고
  - 작업 디렉토리 내용과 스테이징 영역 내용이 최신 커밋 자료로 남음











## 임시저장명령 stash

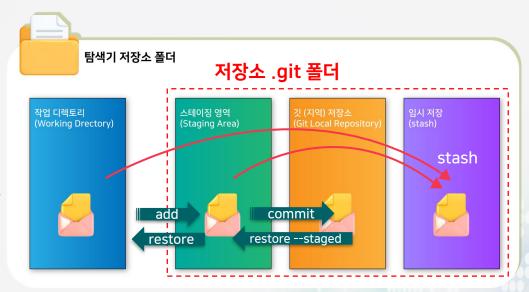
# 🌣 임시저장 명령 stash

## ▼ 작업 폴더와 스테이징 영역을 숨김(stash)에 저장하고 작업 폴더를 정리

- \$ git stash
- \$ git stash -m '메시지'
- \$ git stash save
- \$ git stash save '메시지'

## ☑ 옵션

- --keep-index, -k
  - 스테이징 영역은 제외하고 작업 폴더만 저장
- --include-untracked, -u
  - Untracked 파일도 포함해 저장

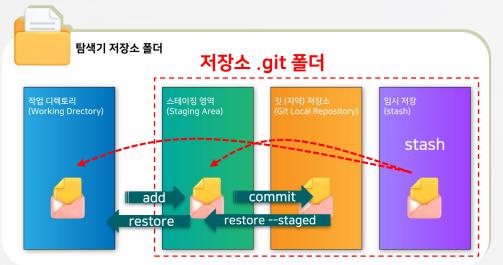






## 👛 임시저장의 최신 저장 내용으로 다시 복원

- ★ 최근 임시저장 내용을 가져와 반영, stash 목록은 그대로
  - 기본으로 작업 디렉토리 내용만 다시 복사해 활용
    - \$ git stash apply
  - 스테이지 영역도 함께 복사하기 위해서는 옵션 사용
    - \$ git stash apply --index





### 임시저장명령 stash



- 옵션 -k | --keep-index
  - 스테이징 영역은 저장하지 않고 그대로 놔둠
  - 그러므로 checkout할 수는 없음
- 옵션 -u | --include-untracked
  - Untracked 파일도 포함해 저장
- - 변경된 모든 사항들을 저장하는 것이 아니며 대화형 프롬프트를 통해 자신이 stash에 저장할 것과 저장하지 않을 것을 지정 가능

- 임시저장명령stash
- Stash 목록 보기
  - ☑ 목록 보기
    - \$ git stash list

가장 최신 것이 D번

- stash@{0}
- stash@{1}
- stash@{2}





# DM



## 2 임시저장명령*s*tash



- ★ 해당 stash 항목이 생성되었을 때의 커밋 자료와 최신 stash 항목 간의 차이로 표시
  - \$ git stash show
    - h.txt | 2 ++
    - 1 file changed, 2 insertions(+)

변화된 파일과 변화된 수만 표시

- \$ git stash show -p
  - -p: 파일 내용의 차이까지 보이기
- ★ 해당 stash 항목이 생성되었을 때의 커밋 자료와 해당 stash 항목 간의
  차이로 표시
  - \$ git stash show stash@{n}
  - \$ git stash show stash@{n}-p
    - -p: 내용의 차이까지 보이기





## 기 임시저장명령 stash

## 🐞 임시 저장된 stash 반영

- ☑ 최근 또는 지정된 임시저장소 내용을 가져와 반영하고 삭제
  - \$ git stash pop
  - \$ git stash pop stash@{n}
- ★ 최근 또는 지정된 임시저장소 내용을 가져와 반영, 작업 디렉토리만 반영, stash 목록은 그대로
  - \$ git stash apply
  - \$ git stash apply stash@{n}
- ★ 최근 또는 지정된 임시저장소 내용을 가져와 반영, 작업 디렉토리와 스테이징 영역도 반영, stash 목록은 그대로
  - \$ git stash apply --index
  - \$ git stash apply --index stash@{n}









- ☑ 최근 임시저장 내용을 삭제
  - \$ git stash drop
- ☑ 지정된 임시저장 내용을 삭제
  - \$ git stash drop stash@{n}
- ▼ 모든 stash 목록을 모두 제거
  - \$ git stash clear





## 임시저장명령 stash

## Untracked 파일 삭제

- **S** git clean
  - fatal: clean.requireForce defaults to true and neither -i, -n, nor -f given; refusing to clean
    - 바로 삭제되지 않음
- ★ \$ git clean -i
  - Would remove the following items:
  - bcd
  - \*\*\* Commands \*\*\*
  - 1: clean 2: filter by pattern 3: select by numbers
  - 4: ask each 5: quit 6: help
  - What now> a
  - Remove b [y/N]? y
  - Remove c [y/N]? y
  - Remove d [y/N]? y
  - Removing b
  - Removing c
  - Removing d
- **★** \$ git clean -f
  - 무조건 삭제





# Summary

- >>> 작업 디렉토리와 스테이징 영역을 숨김(stash)에 저장하고 작업 폴더를 정리
  - \$ git stash
  - ◆ \$ git stash -m '메시지'
  - \$ git stash save
  - ◆ \$ git stash save '메시지'
- >> 최근 또는 지정된 임시저장소 내용을 가져와 반영하고 삭제
  - \$ git stash pop
  - \$ git stash pop stash@{n}
- 최근 또는 지정된 임시저장소 내용을 가져와 반영, 작업 디렉토리만 반영, stash 목록은 그대로
  - \$ git stash apply
  - \$ git stash apply stash@{n}





# Summary

- 최근 또는 지정된 임시저장소 내용을 가져와 반영, 작업 디렉토리와 스테이징 영역도 반영, stash 목록은 그대로
  - \$ git stash apply --index
  - \$ git stash apply --index stash@{n}
- >> 임시저장 목록 보기
  - \$ git stash list
- >> 커밋 자료와 최신 stash 항목 간의 차이로 표시
  - \$ git stash show
  - \$ git stash show -p
- >> 커밋 자료와 해당 stash 항목 간의 차이로 표시
  - \$ git stash show stash@{n}
  - \$ git stash show stash@{n} -p





# Summary

- >> 최근 임시저장 내용을 삭제
  - \$ git stash drop
- >> 지정된 임시저장 내용을 삭제
  - \$ git stash drop stash@{n}
- >>> 모든 stash 목록을 모두 제거
  - \$ git stash clear