

작업폴더와 스테이지를 임시저장

Git stash

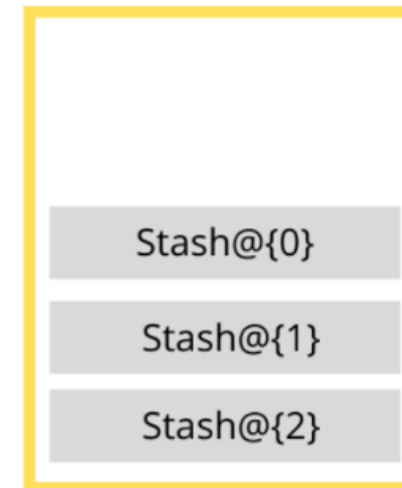
강환수 교수



- 사전적 의미 stash

- 놓다, 남겨두다, 감추다, 안전한 곳에 숨겨두다
- 커밋할 필요 없이 파일의 변경 사항을 숨기거나 비밀리에 저장할 수 있는 강력한 도구
 - WD, SA 의 원래의 내용을 stash 스택에 저장
 - WD, SA => GR의 마지막 커밋으로
- 따로 안전한 곳에 저장했다가 다시 가져올 수 있는 기능
 - 저장 내용
 - 작업 디렉토리 내용과 스테이지 내용

Stack of Git Stashes



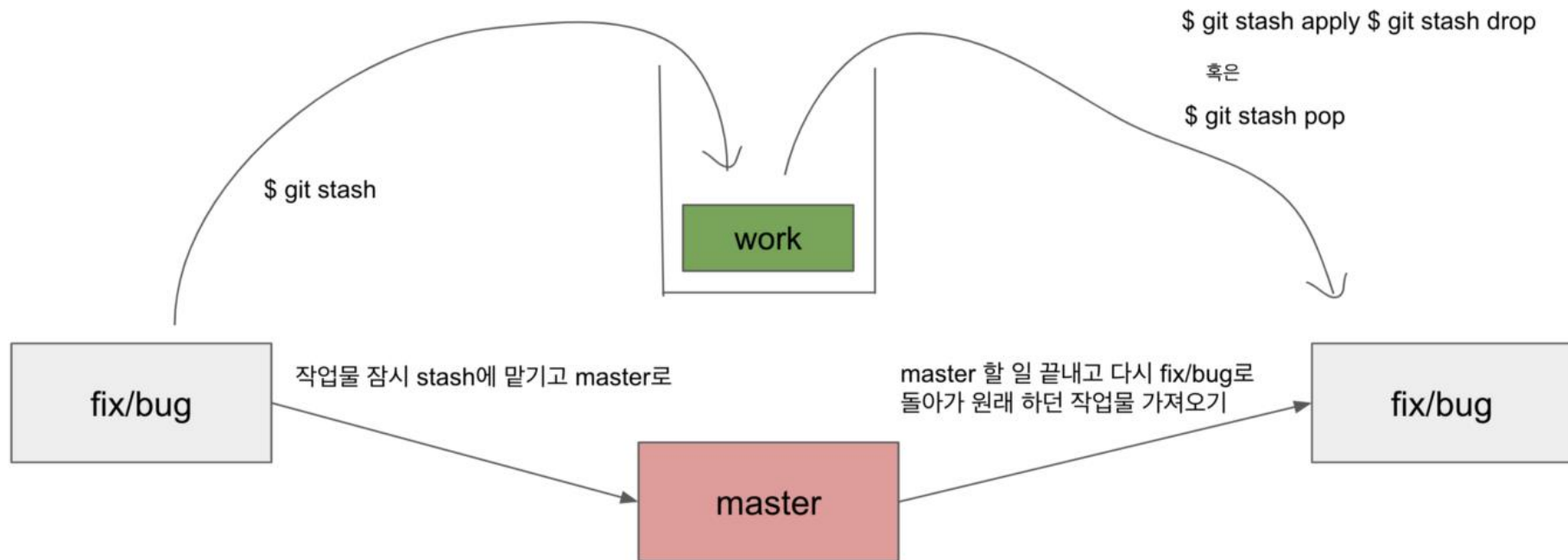
- 깃에서 임시저장소(스택 구조)에 저장

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

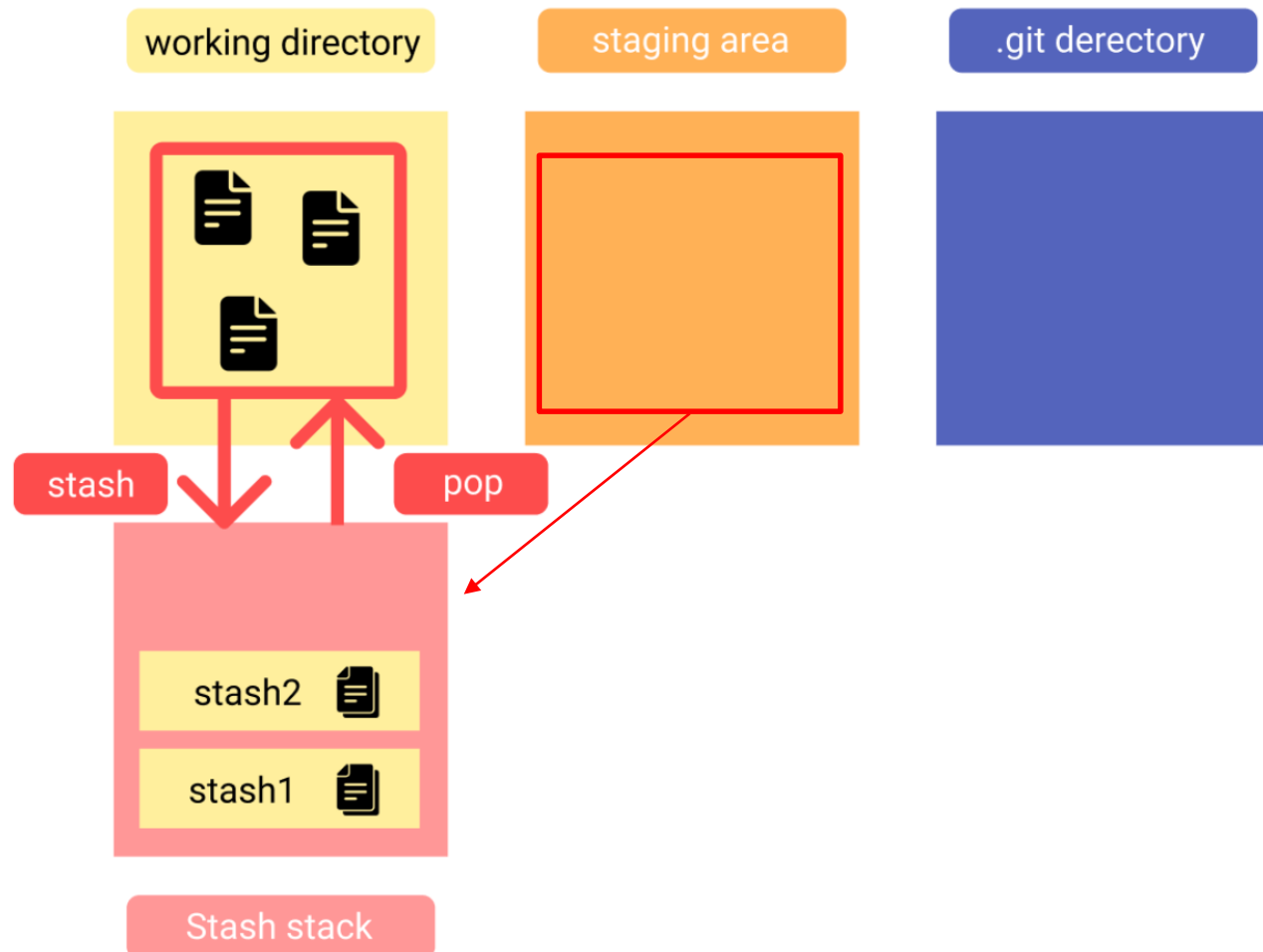
- 참고 사이트

- <https://www.becomebetterprogrammer.com/git-stash-with-name/>

- 특히 브랜치 이동 전에 작업 디렉토리에 정리에 필요할 수 있음
 - 작업 디렉토리가 정리(깨끗한 상태 clean)되고 git이 추적한 모든 파일이 스택에 저장
 - 이 시점에서 자유롭게 변경하고, 새 커밋을 만들고, 분기를 전환하고, 다른 Git 작업을 수행
 - 그런 다음 돌아와서 준비가 되면 보관함을 다시 적용



- Stash로 저장되는 내용
 - 작업 디렉토리 내용과 스테이지 내용
- Stash의 내용 가져다 사용할 때는 기본으로 작업 디렉토리 내용만 다시 복사해 활용
 - 스테이지 영역도 복사하기 위해서는 옵션 사용
 - `apply --index`



- 옵션 **-k | --keep-index**
 - SA는 저장하지 않고 그대로 놔둠
 - 그러므로 checkout할 수 없음
- 옵션 **-u | --include-untracked**
 - 추적되지 않는 파일도 임시저장에 보관
- 옵션 **-p | --patch**
 - 변경된 모든 사항들을 저장하는 것이 아니며 대화형 프롬프트를 통해 자신이 stash에 저장할 것과 저장하지 않을 것을 지정 가능
- 주요 참조
 - <https://godtaehee.tistory.com/m/29>

- `$ git stash`
- `$ git stash -m '메시지'`
- `$ git stash save`
- `$ git stash save '메시지'`
 - 작업 폴더와 스테이징 영역을 숨김(stash)에 저장하고 작업 폴더를 정리
 - 옵션
 - `--keep-index, -k`
 - SA 제외하고, 작업 폴더만 저장
 - `--include-untracked, -u`
 - Untracked 파일도 포함해 저장
- `$ git stash list`
 - 목록 보기
- `$ git stash show`
- `$ git stash show stash@{n}`
- `$ git stash show -p`
- `$ git stash show stash@{n} -p`
 - 해당 stash 항목이 생성되었을 때의 commit 자료(WD의 내용)와 임시로 저장된 stash 내용 간의 차이로 표시
 - `-p`: 내용의 차이까지 보이기

- `$ git stash pop`
- `$ git stash pop stash@{n}`
 - 최근 또는 지정된 임시저장소 내용을 가져와 반영하고 삭제
- `$ git stash apply`
- `$ git stash apply stash@{n}`
 - 최근 또는 지정된 임시저장소 내용을 가져와 반영, 작업 영역만 반영, stash 목록은 그대로
- `$ git stash apply --index`
- `$ git stash apply --index stash@{n}`
 - 최근 또는 지정된 임시저장소 내용을 가져와 반영, 작업 영역과 스테이징 영역도 반영, stash 목록은 그대로
- `$ git stash drop`
- `$ git stash drop stash@{n}`
 - 최근 또는 지정된 임시저장소 내용을 삭제
- `$ git stash clear`
 - stash 목록을 모두 제거

- `$ git clean`
- `$ git clean -f`
 - 무조건 삭제

- 브랜치의 변경 사항이 stash의 변경 사항과 다른 경우 stash를 팝업하거나 적용할 때 충돌이 발생할 수 있음
 - 대신 새 분기를 만들어 다음 항목에 숨김 변경 사항을 적용
- `$ git stash branch new-branch-name stash@{1}`
 - 스테시를 생성한 커밋을 기반으로 새 브랜치를 체크아웃한 다음, 스테시된 변경 사항을 여기에 표시

- 저장소 생성
 - 폴더 stash-lab
- 파일 생성
 - hello.txt

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]
```

```
$ git init stash-lab
```

```
Initialized empty Git repository in D:/[git-lab]/stash-lab/.git/
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]
```

```
$ cd stash-lab
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$ echo stash01 > hello.txt
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$ git add .
```

```
warning: LF will be replaced by CRLF in hello.txt.
```

```
The file will have its original line endings in your working directory
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$ git status
```

```
On branch main
```

```
No commits yet
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   hello.txt
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$ git commit -m 'A'
```

```
[main (root-commit) 740b871] A
```

```
1 file changed, 1 insertion(+)
```

```
create mode 100644 hello.txt
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$ git log --oneline
```

```
740b871 (HEAD -> main) A
```

- 파일 hello.txt 수정 후 stash
 - 작업 폴더가 정리

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$ cat hello.txt  
stash01
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$ echo stash02 >> hello.txt
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$ cat hello.txt  
stash01  
stash02
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$ git status  
On branch main  
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:   hello.txt
```

```
no changes added to commit (use "git add" and/or "git commit -a")
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$ git stash  
warning: LF will be replaced by CRLF in hello.txt.  
The file will have its original line endings in your working  
directory  
Saved working directory and index state WIP on main: 740b871 A
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$ git status  
On branch main  
nothing to commit, working tree clean
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$ cat hello.txt  
stash01
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$ git stash list  
stash@{0}: WIP on main: 740b871 A
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$
```

Work In Progress
CommitID와 메시지도 확인

- 임시 스택에서 제거하지 않으려면 apply 적용

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git stash apply
On branch main
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:   hello.txt

no changes added to commit (use "git add" and/or "git commit -a")

PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ cat hello.txt
stash01
stash02

PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git commit -am 'B'
[main 9e0bdc2] B
1 file changed, 1 insertion(+)

PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git log --oneline
9e0bdc2 (HEAD -> main) B
740b871 A
```

Stash 후 목록 확인과 show

오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ echo stash03 >> hello.txt
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git stash save '03'
warning: LF will be replaced by CRLF in hello.txt.
The file will have its original line endings in your working
directory
Saved working directory and index state On main: 03
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git stash list
stash@{0}: On main: 03
stash@{1}: WIP on main: 740b871 A
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git status
On branch main
nothing to commit, working tree clean
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ cat hello.txt
stash01
stash02
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git stash show
hello.txt | 1 +
1 file changed, 1 insertion(+)
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git stash show -p
diff --git a/hello.txt b/hello.txt
index 863ac87..b71178f 100644
--- a/hello.txt
+++ b/hello.txt
@@ -1,2 +1,3 @@
    stash01
    stash02
+stash03
```

stash 생성 시 이전 커밋과 지정한 stash와의 차이
a/hello.txt: 이전 커밋 내용
b/hello.txt: 그 당시 임시저장 내용

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git stash show -p stash@{1}
diff --git a/hello.txt b/hello.txt
index 5c1e9d9..863ac87 100644
--- a/hello.txt
+++ b/hello.txt
@@ -1 +1,2 @@
    stash01
+stash02
```

다시 적용 후 커밋, stash drop

오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git stash list
stash@{0}: On main: 03
stash@{1}: WIP on main: 740b871 A
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git stash apply stash@{0}
On branch main
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:   hello.txt
```

no changes added to commit (use "git add" and/or "git commit -a")

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git log --oneline
9e0bdc2 (HEAD -> main) B
740b871 A
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git commit -am'C'
[main 25f0559] C
1 file changed, 1 insertion(+)
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git log --oneline
25f0559 (HEAD -> main) C
9e0bdc2 B
740b871 A
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ echo stash04 >> hello.txt
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git status
On branch main
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:   hello.txt
```

no changes added to commit (use "git add" and/or "git commit -a")

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git stash -m '04'
warning: LF will be replaced by CRLF in hello.txt.
The file will have its original line endings in your working
directory
Saved working directory and index state On main: 04
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git stash list
stash@{0}: On main: 04
stash@{1}: On main: 03
stash@{2}: WIP on main: 740b871 A
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git stash drop stash@{1}
Dropped stash@{1} (f8f34b1c2d1d4b70acfded4e7b3d105decf5369c)
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git stash list
stash@{0}: On main: 04
stash@{1}: WIP on main: 740b871 A
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$
```

• Git stash apply

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$ git status  
On branch main  
nothing to commit, working tree clean
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$ git stash apply  
On branch main  
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:   hello.txt
```

```
no changes added to commit (use "git add" and/or "git commit -a")
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$ cat hello.txt  
stash01  
stash02  
stash03  
stash04
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$ git log --oneline  
25f0559 (HEAD -> main) C  
9e0bdc2 B  
740b871 A
```

이미 stash 한 상태이므로
변화되는 것이 없음

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$ git stash list  
stash@{0}: On main: 04  
stash@{1}: WIP on main: 740b871 A
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$ git stash  
Saved working directory and index state WIP on main: 25f0559 C
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$ git stash list  
stash@{0}: WIP on main: 25f0559 C  
stash@{1}: On main: 04  
stash@{2}: WIP on main: 740b871 A
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$ git stash  
No local changes to save
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$ git status  
On branch main  
nothing to commit, working tree clean
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
```

```
$
```

- 충돌 내용이 파일에 저장
 - 다시 수정해서 사용

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git stash list
stash@{0}: WIP on main: 25f0559 c
stash@{1}: On main: 04
stash@{2}: WIP on main: 740b871 A
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git stash apply stash@{2}
Auto-merging hello.txt
CONFLICT (content): Merge conflict in hello.txt
On branch main
Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add <file>..." to mark resolution)
    both modified:   hello.txt
```

no changes added to commit (use "git add" and/or "git commit -a")

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ cat hello.txt
stash01
stash02
<<<<<<< Updated upstream
stash03
=====
>>>>>>> Stashed changes
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git diff
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ code hello.txt
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   hello.txt
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:   hello.txt
```

```
PC@DESKTOP-482NOAB MINGW64 /d/[git-lab]/stash-lab (main)
$ git diff head
diff --git a/hello.txt b/hello.txt
index b71178f..87692cb 100644
--- a/hello.txt
+++ b/hello.txt
@@ -1,3 +1,2 @@
  stash01
-stash02
-stash03
+stash04
```