







- 1. stash, stash save
- 2. stash apply, stash pop



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





임시저장기초

DMU소 동양미래대학교 인공지능소프트웨어학과

● 저장소 생성 후 커밋

✓ 저장소 gstash

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]
$ git init gstash
Initialized empty Git repository in C:/[smart git]/gstash/.git/
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]
$ cd qstash
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gstash (main)
$ echo 111 > f
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gstash (main)
$ git add f
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gstash (main)
$ git stash
You do not have the initial commit yet
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gstash (main)
$ git commit -m A
[main (root-commit) c56bd01] A
1 file changed, 1 insertion(+)
 create mode 100644 f
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gstash (main)
$ git lgag1
                               $ git config --global alias.lgag1 'log --all --graph --oneline'
* c56bd01 (HEAD -> main) A
```

1 임시저장기초



🐞 3 영역을 모두 다르게

333

₩ 파일 f가 3 영역이 모두 다르고 추적되지 않은 파일 g도 생성

# cit add f	71 L MINGW64 /c/[sm	art git]/gstash (ma	diff—q index 58	iffstaged git a/f b/f 8c9bdfa30a52a 100644	
\$ git add f			a/f +++ b/f		
\$ echo aaa > g			@ -1 +1 111		
\$ echo 333 >>> f			+222		
\$ git status -s M f ?? g			diff —	iff HEAD git a/f b/f 8 c9bdf641d574 100644	
			a/ i		
\$ git diff			+++ b/f		
diff —git a/f b/f	IE74 100C44		+++ b/f @@ -1 +£	1,3 @	
diff —git a/f b/f index a30a52a641d	1574 100644		+++ b/f @@ -1 +: 111	1,3 @ HEAD	
diff —git a/f b/f	1574 100644		+++ b/f @@ -1 +£	1,3 @	
diff —git a/f b/f index a30a52a641d —— a/f +++ b/f @@ -1,2 +1,3 @@		스테이징 영역	+++ b/f @ -1 +: 111 +222 +333	1,3 @ HEAD	
diff —git a/f b/f index a30a52a641d —— a/f +++ b/f	574 100644 작업 디렉토리 111	스테이징 영역 111	+++ b/f @ -1 +: 111 +222	1,3 @ HEAD	

임시저장기초



😷 임시 저장에 저장



옵션 -u가 없으면 untracked file은 그대로 남음

```
$ git lgag1
* c56bd01 (HEAD -> main) A
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified: f
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: f
Untracked files:
  (use "git add <file>..." to include in what will be committed)
$ git stash
Saved working directory and index state WIP on main: c56bd01 A
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
nothing added to commit but untracked files present (use "git add" to track)
$ git stash list
stash@{0}: WIP on main: c56bd01 A
```

작업 디렉토리	스테이징 영역	깃 저장소		
111	111	111		
stash@{0}				
작업 디렉토리 스테이징 영역				
111	111			
222	222			
333				





1 임시저장기초

🐡 임시 저장을 다시 작업 디렉토리에 복사

☑ 저장에서 삭제되지 않고 복사

- 스테이징 영역은 복원이 안 됨
 - \$ git stash apply

작업 디렉토리	스테이징 영역	깃 저장소
111	111	111
222		
333		

```
$ 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:
Untracked files:
  (use "git add <file>..." to include in what will be committed)
no changes added to commit (use "git add" and/or "git commit -a")
$ cat f
111
222
                                                         stash@{0}
333
                                                     작업 디렉토리
                                                                 스테이징 영역
$ git diff
diff --git a/f b/f
                                                       111
                                                                    111
index 58c9bdf..641d574 100644
--- a/f
                                                       222
                                                                    222
+++ b/f
                                                       333
@@ -1 +1,3 @@
111
+222
+333
```

임시저장기초



🗰 다시 파일 수정한 후 임시 저장 적용

☑ 스테이징 영역도 함께 복원

• 다시 처음 상태가 됨

```
$ echo 111 > f
                                                                       $ git diff --staged
                                                                       diff -- git a/f b/f
$ git stash list
                                                                       index 58c9bdf..a30a52a 100644
stash@{0}: WIP on main: c56bd01 A
                                                                       --- a/f
$ git stash apply --index
                                                                       +++ b/f
On branch main
                                                                       @@ -1 +1,2 @@
Changes to be committed:
                                                                        111
  (use "git restore --staged <file>..." to unstage)
                                                                       +222
       modified:
Changes not staged for commit:
                                                                       $ ait diff HEAD
  (use "git add <file>..." to update what will be committed)
                                                                       diff -- git a/f b/f
  (use "git restore <file>..." to discard changes in working directory)
                                                                       index 58c9bdf..641d574 100644
        modified: f
                                                                       --- a/f
                                                                       +++ b/f
Untracked files:
  (use "git add <file>..." to include in what will be committed)
                                                                       @@ -1 +1.3 @@
                                                                        111
                                                                       +222
$ git status -s
                                                                       +333
MM f
?? g
                                                                       $ git stash list
$ ait diff
                                                                       stash@{0}: WIP on main: c56bd01 A
diff --git a/f b/f
index a30a52a..641d574 100644
                                                                                             stash@{0}
                                  작업 디렉토리
                                                    스테이징 영역
                                                                       깃 저장소
--- a/f
+++ b/f
                                                                                       작업 디렉토리
                                                                                                      스테이징 영역
                                                         111
                                                                          111
                                      111
@@ -1,2 +1,3 @@
 111
                                                                                                         111
                                                         222
                                                                                           111
                                       222
 222
                                                                 modified
                                                                                          222
                                       333
                                                                                                         222
+333
                                                modified
```

333







임시저장관련다양한명령

Save로 저장

- ★ --keep-index, -k
 이미 index(스테이징 영역)에 추가된 모든 변경 사항은 그대로 유지됨
 - 스테이징 영역은 SA 제외하고, 작업 디렉토리더만 저장, 작업 디렉토리는 스테이징 영역의 내용으로
- --include-untracked, -u
 - Untracked 파일도 포함해 저장

작업 디렉토리	스테이징 영역	깃 저장소
111	111	111
222	222	

•	save 'stash -u -k' -u - g directory and index s	k tate On main: stash -u -k	\$ git stash sho f 2 ++	
	n main: stash -u -k	스테이징 영역의 파일은 임시 대자하지 않으므로 그대로 나온	file changed \$ git stash sho diffgit a/f	
\$ git status On branch ma	in	저장하지 않으므로 그대로 낡음	index 58c9bdf a/f	
(use "git	e committed: restorestaged <file> fied: f</file>	" to unstage)	+++ b/f @@ -1 +1,3 @@ 111	stash 생성 시 커밋과 지정한 stash(최신)와의 차이
\$ cat f 111	stash@{1}	stash@{0}	+222 +333 HEAD	a/hello.txt: 커밋 내용 b/hello.txt: 임시저장 내용
222 \$ git diff	작업 디렉토리 스테이징 영역	작업 디렉토리 스테이징 영역	A	
\$ 1s f	111 111 222 222	111 \$ git stash		
	333	333 Untracks	ed file	



임시저장관련다양한명령

🍅 임시 저장 복원 시 충돌 발생



營 충돌 시 상태 표시

Both modified

```
$ git stash list
                                                                  $ cat f
stash@{0}: On main: stash -u -k
                                                                  111
stash@{1}: WIP on main: c56bd01 A
                                                                  222
                                                                  <<<<< Updated upstream
$ git stash pop
Auto-merging f
                                                                  333
CONFLICT (content): Merge conflict in f
                                                                  >>>>> Stashed changes
On branch main
Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add <file>..." to mark resolution)
        both modified:
                                                                       Updated upstream:
Untracked files:
                                                                        현재 작업 디렉토리 내용
  (use "git add <file>..." to include in what will be committed)
                                                                        Stashed changed:
                                                                      임시 저장으로부터 복원될 내용
no changes added to commit (use "git add" and/or "git commit -a")
The stash entry is kept in case you need it again.
$ git status -s
uu f
?? g
```



임시저장관련다양한명령



\$ git status

₩ 파일 수정 후 add

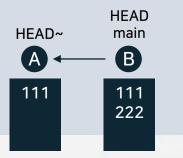
```
On branch main
Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add <file>..." to mark resolution)
        both modified:
Untracked files:
  (use "git add <file>..." to include in what will be committed)
no changes added to commit (use "git add" and/or "git commit -a")
                                   M X
$ code f
                            C: > [smart git] > gstash > ≡ f
$ git add f
                                 111
                                 222
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified: f
Untracked files:
  (use "git add <file>..." to include in what will be committed)
```

```
$ echo 333 >> f

$ git commit -m B
[main d8e0ca0] B
  1 file changed, 1 insertion(+)

$ git log --oneline
d8e0ca0 (HEAD -> main) B
c56bd01 A
```

작업 디렉토리	스테이징 영역	깃 저장소
111	111	111
222	222	222
333		





임시저장관련다양한명령



```
$ git log --oneline
                                                                     111
d8e0ca0 (HEAD -> main) B
                                                                     222
c56bd01 A
                                                                     333
                                                                     $ ait diff
$ git stash list
                                                                     diff --git a/f b/f
stash@{0}: On main: stash -u -k
                                                                     index a30a52a..641d574 100644
stash@{1}: WIP on main: c56bd01 A
                                                                     --- a/f
                                                                     +++ b/f
$ git status
                                                                     @@ -1,2 +1,3 @@
On branch main
                                                                     111
Changes not staged for commit:
                                                                      222
  (use "git add <file>..." to update what will be committed)
                                                                     +333
  (use "git restore <file>..." to discard changes in working
directory)
                                                                     $ git diff --staged
        modified: f
                                                                     $ git diff HEAD
                                                                     diff -- git a/f b/f
Untracked files:
                                                                     index a30a52a..641d574 100644
  (use "git add <file>..." to include in what will be committed)
                                                                     --- a/f
                                                                     +++ b/f
                                                                     @@ -1,2 +1,3 @@
no changes added to commit (use "git add" and/or "git commit -a")
                                                                      111
                                                                      222
$ cat f
                                                                     +333
                                                                                               stash@{0}
                                                                      stash@{1}
                                                                   작업 디렉토리 스테이징 영역
```

작업 디렉토리	스테이징 영역	깃 저장소
111	111	111
222	222	222
333		









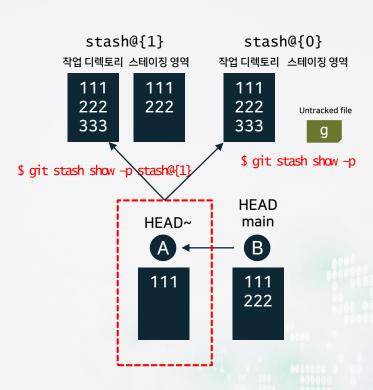
임시저장관련다양한명령

🛑 커밋과 임시 저장과의 비교



★ \$ git stash show -p

```
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gstash (main)
$ git stash show -p
diff -- git a/f b/f
                                        stash 생성 시 커밋과 지정한
index 58c9bdf. 641d574 100644
                                           stash(최신)와의 차이
--- a/f
                                           a/hello.txt: 커밋 내용
+++ b/f
                                         b/hello.txt: 임시저장 내용
@@ -1 +1.3 @@
111
+222
+333
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gstash (main)
$ git stash list
stash@{0}: On main: stash -u -k
stash@{1}: WIP on main: c56bd01 A
파이썬@DESKTOP-8TN3J1L MINGW64 /c/[smart git]/gstash (main)
$ git stash show -p stash@{1}
diff --qit a/f b/f
index 58c9bdf..641d574 100644
                                         stash 생성 시 커밋과 지정한
--- a/f
                                        stash(stash@{1})와의 차이
+++ b/f
                                            a/hello.txt: 커밋 내용
@@ -1 +1,3 @@
                                          b/hello.txt: 임시저장 내용
111
+222
+333
```



DMUA

임시저장관련다양한명령





★ \$ git stash show -p

```
$ 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:
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        q
no changes added to commit (use "git add" and/or "git commit -a")
$ git stash -m 'stash -u' -u
Saved working directory and index state On main: stash -u
$ git status
On branch main
nothing to commit, working tree clean
$ git stash list
stash@{0}: On main: stash -u
stash@{1}: On main: stash -u -k
stash@{2}: WIP on main: c56bd01 A
```





stash@{0}

111

222

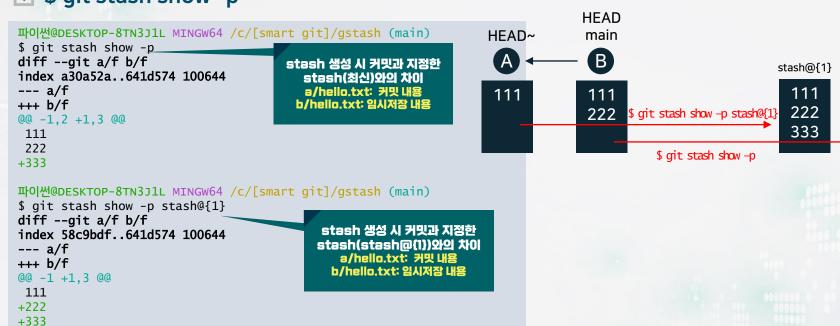
333

임시저장관련다양한명령

🛑 커밋과 임시 저장과의 비교



★ \$ git stash show -p



임시저장관련다양한명령

🌺 임시 저장 목록 삭제



○ 최신이나 지정한 임시 저장 삭제



○ 모든 임시 저장 삭제

```
$ git stash list
stash@{0}: On main: stash -u
stash@{1}: On main: stash -u -k
stash@{2}: WIP on main: c56bd01 A

$ git stash drop stash@{1}
Dropped stash@{1} (aedfded4177658e18dc55f8b047f2fad12945a73)

$ git stash list
stash@{0}: On main: stash -u
stash@{1}: WIP on main: c56bd01 A

$ git stash drop
Dropped refs/stash@{0} (e9dc3e557dbeece9a9eeabab982d5ccda1db650a)

$ git stash list
stash@{0}: WIP on main: c56bd01 A

$ git stash clear

$ git stash list
```





임시저장관련다양한명령

Untracked 파일 삭제

☑ 새로 생성한 파일 삭제

```
$ 1s
                                                                   $ git clean -i
                                                                   would remove the following items:
                                                                     q h i
$ git status
                                                                   *** Commands ***
On branch main
                                                                       1: clean
                                                                                               2: filter by pattern
                                                                                                                        3: select
nothing to commit, working tree clean
                                                                   by numbers
                                                                                               5: quit
                                                                       4: ask each
                                                                                                                        6: help
$ touch g h i
                                                                   what now> 4
                                                                   Remove g[y/N]? n
$ git status
                                                                   Remove h [y/N]? n
On branch main
                                                                   Remove i [y/N]? y
Untracked files:
                                                                   Removing i
  (use "git add <file>..." to include in what will be committed)
                                                                   $ 1s
                                                                   f g h
                                                                   $ git clean -f
nothing added to commit but untracked files present (use "git
                                                                   Removing q
add" to track)
                                                                   Removing h
$ git clean
                                                                   $ 1s
fatal: clean.requireForce defaults to true and neither -i, -n,
nor -f given; refusing to clean
```





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 임시 저장소 내용을 가져와 반영하고 삭제

- \$ git stash pop
- \$ git stash pop stash@{n}





Summary

- >> 최근 또는 지정된 임시 저장소 내용을 가져와 반영, 작업 디렉토리에 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}