

10주차 실습 시트

강 환수 교수

프로젝트	merge & conflict resolve : 병합 기초 [교재 8장 병합과 충돌] 참조: Visual studio 사용자를 위한 git https://wikidocs.net/153871	
모듈1	<input type="checkbox"/> 빨리 감기(fast-forward, 순차) 병합 <ul style="list-style-type: none"> ○ 브랜치 main에서 커밋 1개 ○ 브랜치 첫 커밋에서 브랜치 feat/dict 생성 후 커밋 2개 ○ 브랜치 main으로 돌아와 병합 	
모듈2	<input type="checkbox"/> 두 갈래의 브랜치를 main에서 3-way 병합(merge)한 후 이전 브랜치 삭제	
모듈3	<input type="checkbox"/> 기본이 빨리 감기(ff: fast-forward) 병합할 수 있는 상태에서 3-way 병합을 하도록 옵션 사용 <ul style="list-style-type: none"> ○ 3-way 병합(커밋 메시지 적용)을 위해 옵션 --no-ff 사용(non fast-forward) 	
모듈4	<input type="checkbox"/> 병합 충돌 해결을 위해 충돌 발생 <ul style="list-style-type: none"> ○ 충돌 발생 시 내용 살피고, 병합 취소 	
모듈5	<input type="checkbox"/> 충돌 해결 방법: 일반적인 add, commit이 필요 <ul style="list-style-type: none"> ○ 소스 파일 수정 => add => commit 	
준비	<input type="checkbox"/> 다음 폴더를 생성 후 git/10w에서 git bash 실행 후 시작	
모듈1 실습 (20min)	<input type="checkbox"/> 빨리 감기(fast-forward, 순차) 병합 <ul style="list-style-type: none"> ○ 브랜치 main에서 커밋 1개 <pre> PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w \$ git init mg Initialized empty Git repository in C:/OSS/git/10w/mg/.git/ PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w \$ cd mg PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ echo 'print("python sample code")' > main.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git add main.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git commit -m 'create main.py' [main (root-commit) b4d5e0f] create main.py 1 file changed, 1 insertion(+) create mode 100644 main.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git lg1 # log --oneline --graph * b4d5e0f (HEAD -> main) create main.py </pre>	

○ 브랜치 첫 커밋에서 브랜치 feat/dict 생성 후 커밋 2개	
<pre> PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git branch feat/dict # 브랜치 생성 PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git branch feat/dict * main PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git switch feat/dict # 브랜치 이동 Switched to branch 'feat/dict' PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (feat/dict) \$ echo 'pl = dict()' > mydict.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (feat/dict) \$ git add . PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (feat/dict) \$ git commit -m 'create mydict.py' [feat/dict 12433ae] create mydict.py 1 file changed, 1 insertion(+) create mode 100644 mydict.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (feat/dict) \$ git lgg # log --graph * commit 12433ae3108d296d51d02335706cd99d2f02e1d9 (HEAD -> feat/dict) Author: hskang <ai7dnn@gmail.com> Date: Sat Sep 10 16:23:47 2022 +0900 create mydict.py * commit b4d5e0f7b8f985441dd1e56fdae3c806e4a7ba49 (main) Author: hskang <ai7dnn@gmail.com> Date: Sat Sep 10 16:22:10 2022 +0900 create main.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (feat/dict) \$ git lgg1 * 12433ae (HEAD -> feat/dict) create mydict.py * b4d5e0f (main) create main.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (feat/dict) \$ echo 'pl["python"] = 1991' >> mydict.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (feat/dict) \$ git commit -am 'insert a python key & value' [feat/dict afeb7fd] insert a python key & value 1 file changed, 1 insertion(+) PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (feat/dict) \$ git lgg1 * afeb7fd (HEAD -> feat/dict) insert a python key & value * 12433ae create mydict.py * b4d5e0f (main) create main.py </pre>	
○ 브랜치 main으로 돌아와 병합	
<pre> PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (feat/dict) \$ git checkout main # 브랜치 이동 Switched to branch 'main' PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git merge feat/dict # Fast-forward 병합 Updating b4d5e0f..afeb7fd Fast-forward mydict.py 2 ++ 1 file changed, 2 insertions(+) # 하나의 파일 내용에 2개의 줄이 추가 create mode 100644 mydict.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git lgg1 * afeb7fd (HEAD -> main, feat/dict) insert a python key & value </pre>	fast-forward 병합은 새로이 커밋이 발생하지 않음

	<pre> * 12433ae create mydict.py * b4d5e0f create main.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git rev-parse feat/dict afeb7fd9d4f6c413e17a24a2f7726822c67afbb3 PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git log commit afeb7fd9d4f6c413e17a24a2f7726822c67afbb3 (HEAD -> main, feat/dict) Author: hskang <ai7dnn@gmail.com> Date: Sat Sep 10 16:26:07 2022 +0900 insert a python key & value commit 12433ae3108d296d51d02335706cd99d2f02e1d9 Author: hskang <ai7dnn@gmail.com> Date: Sat Sep 10 16:23:47 2022 +0900 create mydict.py commit b4d5e0f7b8f985441dd1e56fdae3c806e4a7ba49 Author: hskang <ai7dnn@gmail.com> Date: Sat Sep 10 16:22:10 2022 +0900 create main.py </pre>	rev-parse 확인
<div>모듈2</div> <div>실습</div> <div>(20min)</div>	<div>□ 브랜치 devp/dict 생성, 파일 mydict.py 한 줄 추가한 후 커밋</div> <pre> PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git branch devp/dict # 브랜치 생성 PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git switch devp/dict # 브랜치 이동 Switched to branch 'devp/dict' PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (devp/dict) \$ echo 'pl["java"] = 1995' >> mydict.py # 파일에 마지막 줄 추가 PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (devp/dict) \$ git commit -am'insert a java key & value' [devp/dict f72e831] insert a java key & value 1 file changed, 1 insertion(+) PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (devp/dict) \$ git lggl * f72e831 (HEAD -> devp/dict) insert a java key & value * afeb7fd (main, feat/dict) insert a python key & value * 12433ae create mydict.py * b4d5e0f create main.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (devp/dict) \$ cat mydict.py pl = dict() pl["python"] = 1991 pl["java"] = 1998 PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (devp/dict) \$ git checkout main # 브랜치 이동 Switched to branch 'main' PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ cat mydict.py pl = dict() pl["python"] = 1991 </pre>	<div>파일 mydict.py at devp/dict</div> <pre> pl = dict() pl["python"] = 1991 pl["java"] = 1998 </pre> <div>브랜치 main은 브랜치 devp/dict보다 하나 전 단계이므로 파일이 다름</div> <div>파일 mydict.py at main</div> <pre> pl = dict() pl["python"] = 1991 </pre>
	<div>□ 브랜치 main에서 다시 파일 main.py 한 줄 추가한 후 커밋</div> <pre> PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ cat main.py print("python sample code") PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ echo 'print("dict")' >> main.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git commit -am'add print dict' </pre>	<div>브랜치 main과 devp/dict의 공통조상은 커밋 afeb7fd</div>

	<pre>[main cb6273f] add print dict 1 file changed, 1 insertion(+) PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git lggl --all # 모든 브랜치의 로그를 모두 표시 * cb6273f (HEAD -> main) add print dict * f72e831 (devp/dict) insert a java key & value / * afeb7fd (feat/dict) insert a python key & value * 12433ae create mydict.py * b4d5e0f create main.py</pre>	
□ 두 갈래의 브랜치를 main에서 3-way 병합한 후 다시 취소		
	<pre>PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git merge devp/dict # 병합: main에서 devp/dict를 병합 Merge made by the 'ort' strategy. mydict.py 1 + 1 file changed, 1 insertion(+)</pre> <pre> MERGE_MSG C:\> OSS > git > 10w > mg > .git > MERGE_MSG 1 Merge branch 'devp/dict' 2 # Please enter a commit message to explain why this merge is necessary, 3 # especially if it merges an updated upstream into a topic branch. 4 # 5 # Lines starting with '#' will be ignored, and an empty message aborts 6 # the commit. 7</pre> <pre>PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git lggl --all * 40b0104 (HEAD -> main) Merge branch 'devp/dict' \ * f72e831 (devp/dict) insert a java key & value * cb6273f add print dict / * afeb7fd (feat/dict) insert a python key & value * 12433ae create mydict.py * b4d5e0f create main.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ cat mydict.py pl = dict() pl["python"] = 1991 pl["java"] = 1998</pre>	<p>편집기 설정 \$ git config --global core.editor 'code --wait'</p> <p>현재 설정된 편집기인 vscode에 메시지가 자동으로 생성: Merge branch 'devp/dict'</p> <p>vscode에서 저장(ctrl + s)한 후 파일 종료한 후</p> <p>병합전략 'ort' strategy. Ostensibly Recursive's Twin</p> <p>파일 mydict.py이 1줄 추가됨 mydict.py 1 + 1 file changed, 1 insertion(+)</p> <p>돌아와서 로그 정보를 보면 병합(그림 상 결합) 확인</p>
	<pre>PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git reset --hard head~ # 아직 모르는 내용, 바로 전 병합 취소 명령 HEAD is now at cb6273f add print dict PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git lggl --all * cb6273f (HEAD -> main) add print dict * f72e831 (devp/dict) insert a java key & value / * afeb7fd (feat/dict) insert a python key & value * 12433ae create mydict.py * b4d5e0f create main.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ cat mydict.py pl = dict() pl["python"] = 1991</pre>	<p>\$ git reset --hard head~ HEAD~의 커밋 내용으로 WD와 SA를 모두 복사해 이전 상태로 회귀</p> <p>이전 단계로 이동: 병합 취소</p>
□ 병합 시 커밋 메시지 명시적으로 편집 요청		
	<pre>PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git merge devp/dict --edit Merge made by the 'ort' strategy. mydict.py 1 + 1 file changed, 1 insertion(+)</pre> <pre> MERGE_MSG C:\> OSS > git > 10w > mg > .git > MERGE_MSG 1 Merge branch 'devp/dict' to main 2 # Please enter a commit message to explain why this merge is necessary, 3 # especially if it merges an updated upstream into a topic branch. 4 # 5 # Lines starting with '#' will be ignored, and an empty message aborts 6 # the commit. 7</pre>	

	<div data-bbox="304 152 667 181" data-label="Section-Header"> <p>□ 병합 후 이전 브랜치 삭제</p> </div> <div data-bbox="304 208 1015 1227" data-label="Text"> <pre> PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git lgg1 --all * bbfbf6a (HEAD -> main) Merge branch 'devp/dict' to main \ * f72e831 (devp/dict) insert a java key & value * cb6273f add print dict / * afeb7fd (feat/dict) insert a python key & value * 12433ae create mydict.py * b4d5e0f create main.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git branch --merged # 병합된 브랜치 목록 보기, *은 현재 브랜치 devp/dict feat/dict * main PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git branch --no-merged # 병합되지 않은 목록 보기, 현재 없음 PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git branch devp/dict feat/dict * main PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git branch -d devp/dict # 이미 병합된 브랜치 삭제 Deleted branch devp/dict (was f72e831). PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git lgg1 --all * bbfbf6a (HEAD -> main) Merge branch 'devp/dict' to main \ * f72e831 insert a java key & value * cb6273f add print dict / * afeb7fd (feat/dict) insert a python key & value * 12433ae create mydict.py * b4d5e0f create main.py </pre> </div> <div data-bbox="1109 510 1455 607" data-label="Text"> <p># 병합된 브랜치 목록 보기, *은 현재 브랜치 \$ git branch --merged</p> </div>
<div data-bbox="165 1624 248 1736" data-label="Text"> <p>모듈3 실습 (20min)</p> </div>	<div data-bbox="304 1247 1262 1321" data-label="Section-Header"> <p>□ 빨리 감기(ff: fast-forward) 병합할 수 있는 상태에서 3-way 병합을 하도록 ○ 먼저 브랜치 rels/dict를 생성해 파일 mydict.py에 한 줄 추가 후 커밋</p> </div> <div data-bbox="304 1339 1000 2107" data-label="Text"> <pre> PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git branch rels/dict PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git switch rels/dict Switched to branch 'rels/dict' PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (rels/dict) \$ echo 'pl["c#"] = 2000' >> mydict.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (rels/dict) \$ cat mydict.py pl = dict() pl["python"] = 1991 pl["java"] = 1998 pl["c#"] = 2000 PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (rels/dict) \$ git commit -am'insert a c# key & value' [rels/dict bc5c72c] insert a c# key & value 1 file changed, 1 insertion(+) PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (rels/dict) \$ git checkout main # 브랜치 이동 Switched to branch 'main' PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git lgg1 --all * bc5c72c (rels/dict) insert a c# key & value * bbfbf6a (HEAD -> main) Merge branch 'devp/dict' to main </pre> </div>

	<pre> \ * f72e831 insert a java key & value * cb6273f add print dict / * afeb7fd (feat/dict) insert a python key & value * 12433ae create mydict.py * b4d5e0f create main.py </pre>	
	<input type="checkbox"/> 바로 병합하면 빨리 감기(ff) 병합, 그러므로 병합 취소하고 다음에 옵션 --no-ff	
	<pre> PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git merge rels/dict Updating bbfbf6a..bc5c72c Fast-forward mydict.py 1 + 1 file changed, 1 insertion(+) PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git lgg1 --all * bc5c72c (HEAD -> main, rels/dict) insert a c# key & value * bbfbf6a Merge branch 'devp/dict' to main \ * f72e831 insert a java key & value * cb6273f add print dict / * afeb7fd (feat/dict) insert a python key & value * 12433ae create mydict.py * b4d5e0f create main.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git reset --hard HEAD^ HEAD is now at bbfbf6a Merge branch 'devp/dict' to main PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git lgg1 --all * bc5c72c (rels/dict) insert a c# key & value * bbfbf6a (HEAD -> main) Merge branch 'devp/dict' to main \ * f72e831 insert a java key & value * cb6273f add print dict / * afeb7fd (feat/dict) insert a python key & value * 12433ae create mydict.py * b4d5e0f create main.py </pre>	
	<input type="checkbox"/> 3-way 병합(커밋 메시지 적용)을 위해 옵션 --no-ff 사용(non fast-forward)	
	<pre> PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git lgg1 --all * bc5c72c (rels/dict) insert a c# key & value * bbfbf6a (HEAD -> main) Merge branch 'devp/dict' to main \ * f72e831 insert a java key & value * cb6273f add print dict / * afeb7fd (feat/dict) insert a python key & value * 12433ae create mydict.py * b4d5e0f create main.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git merge rels/dict --no-ff # 3-way 병합으로 Merge made by the 'ort' strategy. mydict.py 1 + 1 file changed, 1 insertion(+) ≡ MERGE_MSG X C: > OSS > git > 10w > mg > .git > ≡ MERGE_MSG 1 Merge branch 'rels/dict' 2 # Please enter a commit message to explain why this merge is necessary, 3 # especially if it merges an updated upstream into a topic branch. 4 # 5 # Lines starting with '#' will be ignored, and an empty message aborts 6 # the commit. 7 PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git lgg1 --all * f878cb5 (HEAD -> main) Merge branch 'rels/dict' \ * bc5c72c (rels/dict) insert a c# key & value </pre>	

	<pre> / * bbfbfba Merge branch 'devp/dict' to main \ * f72e831 insert a java key & value * cb6273f add print dict / * afeb7fd (feat/dict) insert a python key & value * 12433ae create mydict.py * b4d5e0f create main.py </pre>	
	<pre> PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git reset --hard HEAD^ HEAD is now at bbfbfba Merge branch 'devp/dict' to main PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/mg (main) \$ git lgg1 --all * bc5c72c (rels/dict) insert a c# key & value * bbfbfba (HEAD -> main) Merge branch 'devp/dict' to main \ * f72e831 insert a java key & value * cb6273f add print dict / * afeb7fd (feat/dict) insert a python key & value * 12433ae create mydict.py * b4d5e0f create main.py </pre>	다시 병합 취소
<div> <div>모듈4</div> <div>실습</div> </div>	<div> <div></div> <div>병합 충돌 해결을 위해 충돌 발생, 파일 mylst.py 생성 후 커밋</div> </div>	
	<pre> PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w \$ git init lstmg Initialized empty Git repository in C:/OSS/git/10w/lstmg/.git/ PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w \$ cd lstmg PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (main) \$ code mylst.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (main) \$ git add mylst.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (main) \$ git commit -m 'create mylst.py' [main (root-commit) 21fd176] create mylst.py 1 file changed, 3 insertions(+) create mode 100644 mylst.py </pre>	<pre> print('branch conflict') print(1, 2, 3) print([1, 2, 3]) </pre>
	<div> <div></div> <div>○ 브랜치 devp/lst 1줄 추가해 커밋</div> </div>	
	<pre> PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (main) \$ git branch devp/lst PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (main) \$ git switch devp/lst Switched to branch 'devp/lst' PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (devp/lst) \$ code mylst.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (devp/lst) \$ git commit -am 'add range' [devp/lst ea3c992] add range 1 file changed, 3 insertions(+), 1 deletion(-) </pre>	<pre> print('branch conflict') print(1, 2, 3) print([1, 2, 3]) print(list(range(1, 10))) </pre>
	<div> <div></div> <div>○ 브랜치 main으로 돌아와 파일 수정 커밋</div> </div>	
	<pre> PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (devp/lst) \$ git switch main Switched to branch 'main' PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (main) \$ code mylst.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (main) \$ git lgg1 --all * ea3c992 (devp/lst) add range * 21fd176 (HEAD -> main) create mylst.py </pre>	

	<pre> PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (main) \$ cat mylst.py print('branch conflict') print(1, 2, 3) print([1, 2, 3]) PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (main) \$ code mylst.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (main) \$ cat mylst.py print('branch conflict') print(1, 2, 3) print([1, 2, 3]) print([i for i in range(1, 10)]) PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (main) \$ git commit -am'add list comp' [main 805bc90] add list comp 1 file changed, 3 insertions(+), 1 deletion(-) PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (main) \$ git lg1 --all * 805bc90 (HEAD -> main) add list comp * ea3c992 (devp/lst) add range / * 21fd176 create mylst.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (main) \$ git switch - # 이전 브랜치로 이동 Switched to branch 'devp/lst' PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (devp/lst) \$ cat mylst.py print('branch conflict') print(1, 2, 3) print([1, 2, 3]) print(list(range(1, 10))) </pre>	<pre> print('branch conflict') print(1, 2, 3) print([1, 2, 3]) 다음 줄 추가 print([i for i in range(1, 10)]) main 브랜치에서 mylst.py print('branch conflict') print(1, 2, 3) print([1, 2, 3]) print([i for i in range(1, 10)]) devp/lst 브랜치에서 mylst.py print('branch conflict') print(1, 2, 3) print([1, 2, 3]) print(list(range(1, 10))) </pre>
<div> <input type="checkbox"/> 브랜치 main에서 브랜치 devp/lst 합병하면 충돌 발생 </div> <div> <input type="radio"/> 충돌 발생시 내용 살피고, 병합 취소 </div>		
	<pre> PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (main) \$ git merge devp/lst Auto-merging mylst.py CONFLICT (content): Merge conflict in mylst.py Automatic merge failed; fix conflicts and then commit the result. PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (main MERGING) \$ cat mylst.py print('branch conflict') print(1, 2, 3) print([1, 2, 3]) <<<<<<< HEAD # 현재 HEAD가 위치한 main의 내용 print([i for i in range(1, 10)]) ===== print(list(range(1, 10))) >>>>>>> devp/lst # 다른 브랜치 devp/lst에서 main의 내용 PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (main MERGING) \$ git ls-files -u # 병합되지 않은 파일 목록 100644 12844f94144dd2d5fce61a47aa4e8222fbe2390c 1 mylst.py 100644 ece0fc040a33fffc661e632c9926adee82d02350 2 mylst.py 100644 436b5161f8126a0869a4116f4da0c8d065acf51d 3 mylst.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (main MERGING) \$ git st On branch main You have unmerged paths. (fix conflicts and run "git commit") (use "git merge --abort" to abort the merge) Unmerged paths: </pre>	<p>브랜치 main의 파일과 브랜치 devp/lst 파일의 내용을 합칠 수 없으므로 충돌 발생</p> <p>다음 표시가 충돌을 의미 (main MERGING)</p> <p>충돌한 파일 내용을 보여 줌</p>

(use "git add <file>..." to mark resolution)
both modified: mylst.py

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

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg
(main|MERGING)

\$ git merge --abort # 충돌한 병합 취소

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (main)

\$ git lggl --all

* 805bc90 (HEAD -> main) add list comp

| * ea3c992 (devp/lst) add range

|/

* 21fd176 create mylst.py

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (main)

\$ git branch --merged

* main

PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (main)

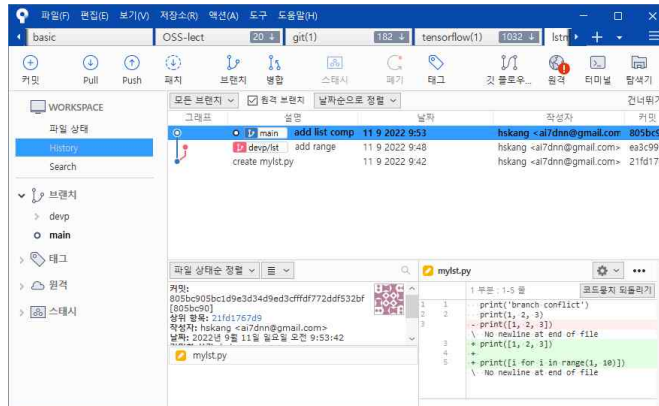
\$ git branch --no-merged

devp/lst

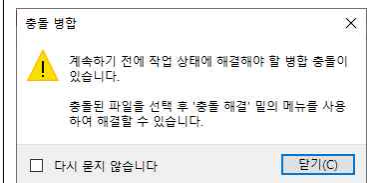
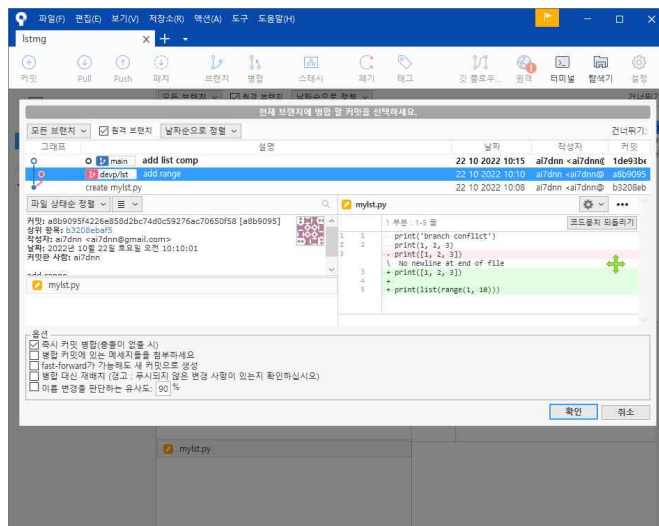
브랜치 별로 병합 여부를
판단하는 방법

☐ 소스 트리에서 합병 전 살피고, 취소

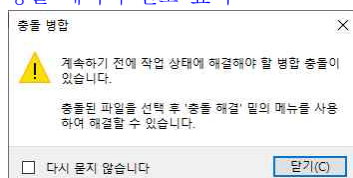
소스트리로 저장소 열기



병합 버튼 클릭 > 병합할 브랜치(커밋)을 선택한 후 확인 클릭

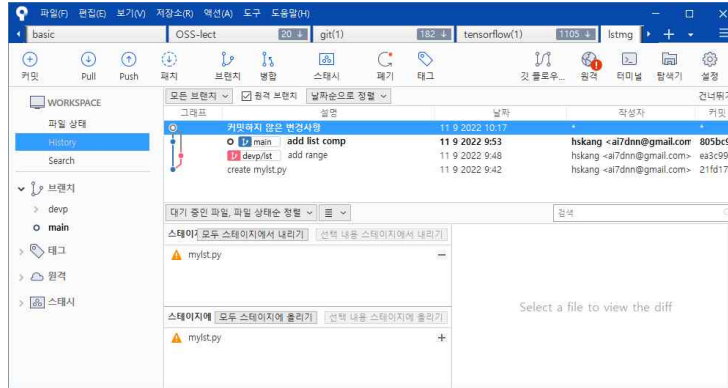


충돌 메시지 윈도 표시

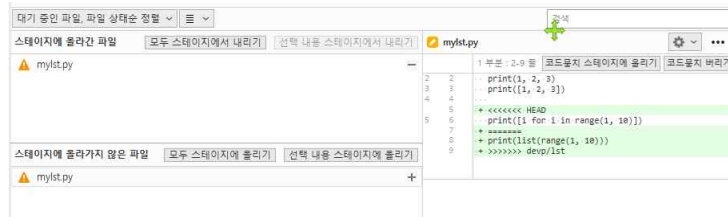


"커밋되지 않은 변경사항"이라는 것이 가장 상위에 추가됨

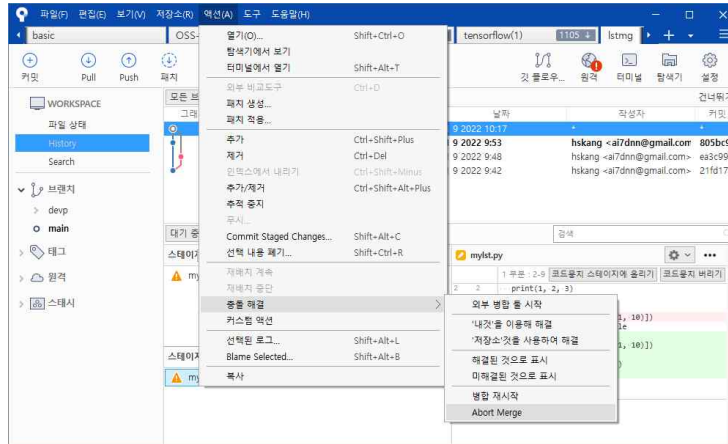
“커밋되지 않은 변경사항”을 클릭한 후 충돌된 파일 확인



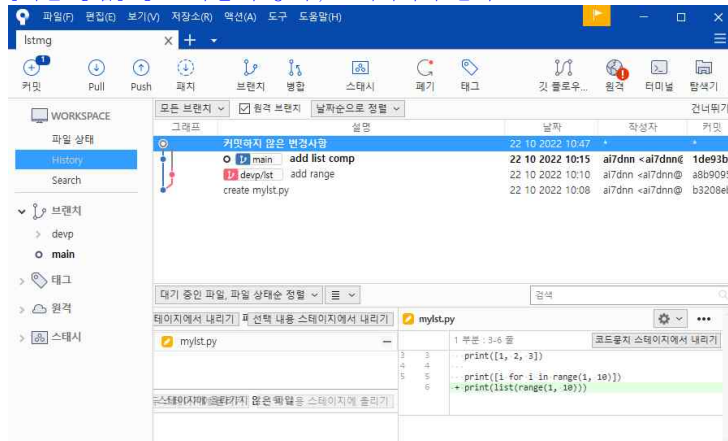
충돌 파일 확인



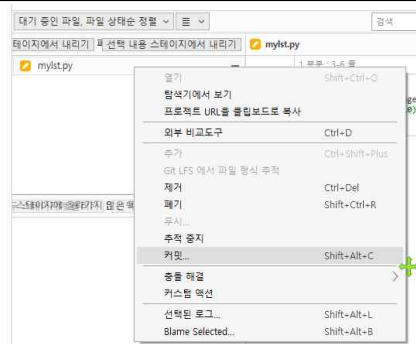
메뉴 액션 > 충돌 해결 > Abort Merge 선택



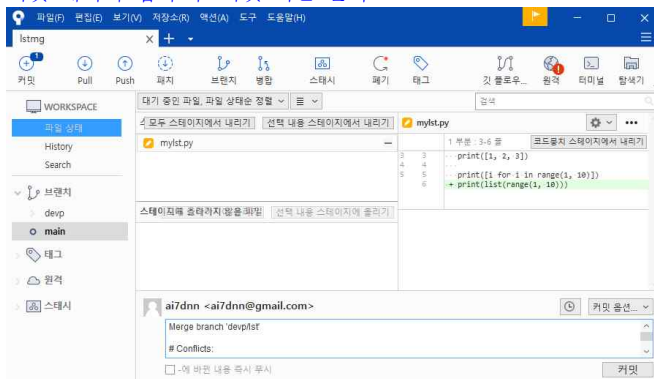
[해결 방법] 충돌 파일 수정 후, 스테이지에 올리고



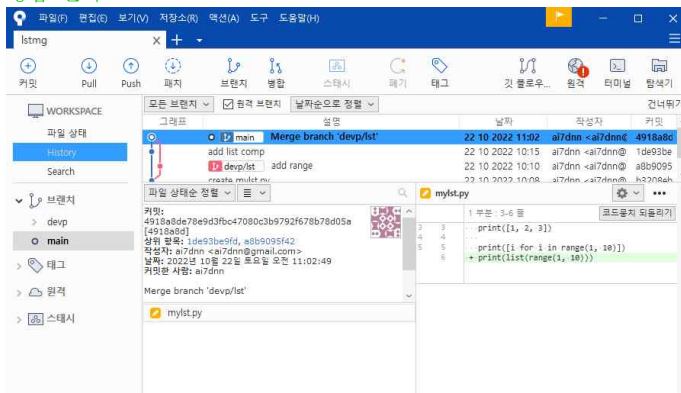
커밋 메뉴 선택



커밋 메시지 입력 후 커밋 버튼 클릭



병합 결과



- 충돌 해결 방법: 일반적인 add, commit이 필요
- 소스 파일 수정 => add => commit

모듈5 실습

```
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/istmg (main)
$ git st
On branch main
nothing to commit, working tree clean
```

```
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/istmg (main)
$ git merge devp/ist
Auto-merging mylst.py
CONFLICT (content): Merge conflict in mylst.py
Automatic merge failed; fix conflicts and then commit the result.
```

```
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/istmg
(main|MERGING)
$ git lggl --all
* 805bc90 (HEAD -> main) add list comp
| * ea3c992 (devp/ist) add range
|/
* 21fd176 create mylst.py
```

```
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/istmg
(main|MERGING)
$ code mylst.py
```

```
PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/istmg
```

충돌이 발생한 이후는
브랜치명이 바뀜
(main|MERGING)

	<pre> (main MERGING) \$ git add mylst.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (main MERGING) \$ git commit -m'resolve conflict in mylst.py' [main 7c5f2ea] resolve conflict in mylst.py PC@DESKTOP-482NOAB MINGW64 /c/OSS/git/10w/lstmg (main) \$ git log --all * 7c5f2ea (HEAD -> main) resolve conflict in mylst.py \ * ea3c992 (devp/lst) add range * 805bc90 add list comp / * 21fd176 create mylst.py </pre>	<p>해결 순서</p> <p>❶ 파일 직접 수정</p> <pre> print('branch conflict') print(1, 2, 3) print([1, 2, 3]) print([i for i in range(1, 10)]) print(list(range(1, 10))) </pre> <p>❷ add</p> <p>❸ commit</p>
--	---	---