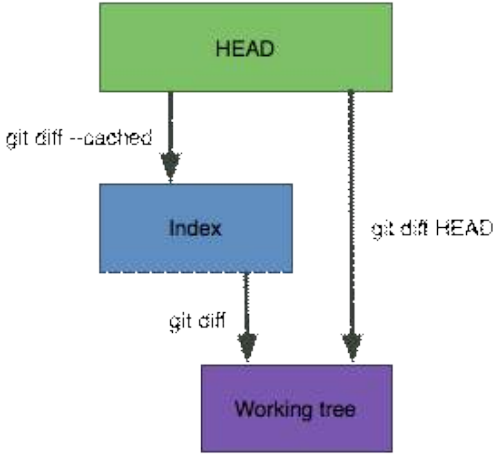


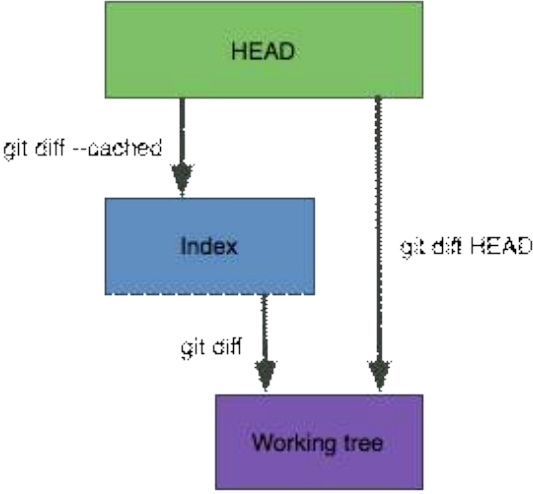
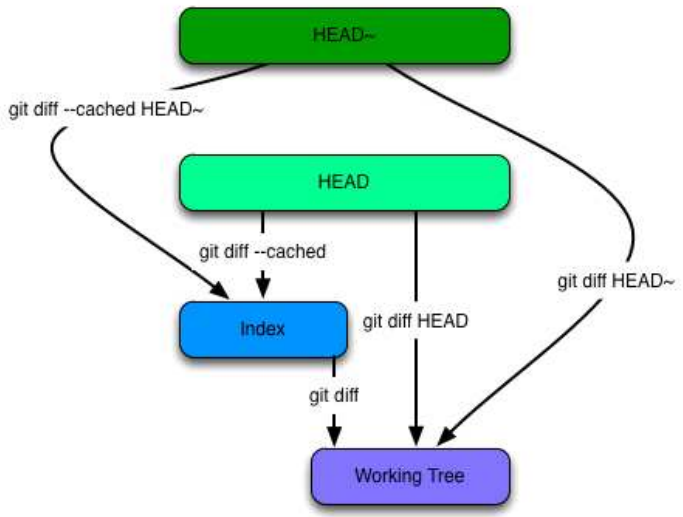
# 7주차 실습 시트

강 환수 교수

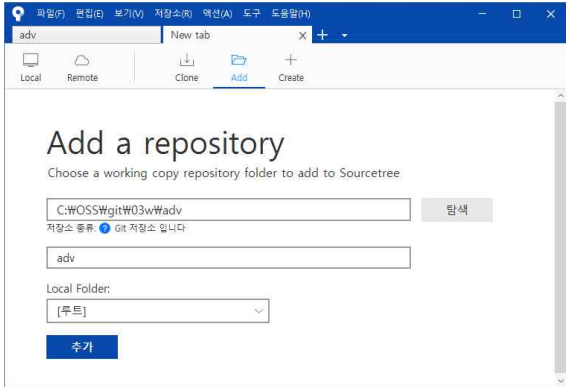
프로젝트	git diff: 스테이징 영역과 작업 공간, 커밋 이력의 파일 비교 [교재 4장]	
모듈1	<input type="checkbox"/> SA와 WD의 파일 차이 보기 diff <input type="radio"/> \$ git diff	
모듈2	<input type="checkbox"/> \$ git diff --cached <input type="radio"/> diff의 비교 기준(from): 커밋된 파일, diff의 비교의 대상(to): 작업 공간의 파일 <input type="radio"/> diff의 옵션 --cached의 의미: 비교의 대상(to)이 stage(index) 영역	
모듈3	<input type="checkbox"/> 명령어 diff : from a1 to a2(a1을 기준으로 a2를 비교)에서 to인 a2는 명령어로 정해짐 <input type="radio"/> --cached이면 index, --cached 없이 git diff이면 작업영역(WD) <input type="radio"/> from을 HEAD 등으로 기술하면 여러 커밋 중 하나가 가능 <input type="radio"/> 다만 \$ git diff : SA와 WD를 비교 <input type="checkbox"/> \$ git diff head1 head2 <input type="radio"/> 커밋 이력 간, 즉 head1에서 head2의 파일 비교 <div data-bbox="466 860 1316 1570" data-label="Diagram"> <p style="text-align: center;"><b>Git Basic Workflow Lifecycle</b> <b>Git diff</b></p> <pre> graph TD     subgraph Local         WD[Working Directory]         SA[Staging Area]         GD[.git directory Repository]     end     subgraph Remote         RA[Remote Area]     end     WD -- "git diff" --&gt; SA     SA -- "git diff HEAD" --&gt; WD     SA -- "git diff --staged HEAD" --&gt; GD     GD -- "git diff HEAD" --&gt; SA     GD -- "git diff HEAD" --&gt; RA     RA -- "git diff HEAD" --&gt; GD   </pre> </div>	
모듈4	<input type="checkbox"/> 소스트리에서 이미 생성된 저장소 살펴 보기 <input type="checkbox"/> 소스트리에서 유명 오픈소스 저장소 복제(clone)하고 살펴 보기	
준비	<input type="checkbox"/> 다음 폴더를 생성 후 git/05w에서 git bash 실행 후 시작 <input type="checkbox"/> vscode, github, source tree 준비	
모듈1 실습 (30분)	<input type="checkbox"/> SA와 WD의 파일 차이 보기(diff) <pre> \$ git init gdiff Initialized empty Git repository in C:/OSS/git/03w/gdiff/.git/  \$ cd gdiff  \$ touch dtype.py  \$ git st On branch main  No commits yet           </pre>	빈 파일 생성 <pre>\$ git config --global alias.st status</pre>

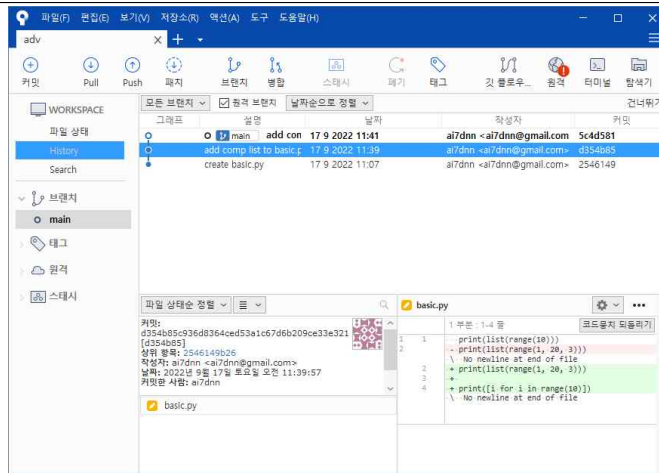
<p>Untracked files: (use "git add &lt;file&gt;..." to include in what will be committed) dtype.py</p> <p>nothing added to commit but untracked files present (use "git add" to track)</p> <p>\$ git sts ?? dtype.py</p>	<p>아직 커밋을 없고</p> <p>\$ git config --global alias.sts 'status -s'</p>
<p>\$ cat dtype.py</p> <p>\$ git diff</p>	<p>staging area와 WD의 파일 차이 비교: staging area 비어 있으면 출력 없음</p>
<p>\$ code dtype.py</p>	<p>print('python')</p>
<p>\$ cat dtype.py print('python')</p> <p>\$ git sts ?? dtype.py</p> <p>\$ git add .</p> <p>\$ git sts A dtype.py</p> <p>\$ git st On branch main</p> <p>No commits yet</p> <p>Changes to be committed: (use "git rm --cached &lt;file&gt;..." to unstage) new file: dtype.py</p>	
<p>\$ git diff</p>	<p>staging area 파일과 WD 파일이 차이가 없으면 출력 없음</p>
<p>\$ code dtype.py</p>	<p>한 줄 추가 저장 print('python') print(list('python'))</p>
<p>\$ git st On branch main</p> <p>No commits yet</p> <p>Changes to be committed: (use "git rm --cached &lt;file&gt;..." to unstage) new file: dtype.py</p> <p>Changes not staged for commit: (use "git add &lt;file&gt;..." to update what will be committed) (use "git restore &lt;file&gt;..." to discard changes in working directory) modified: dtype.py</p> <p>\$ git sts AM dtype.py</p>	<p>staging area에 있는 파일을 수정한 상태</p>
<p>\$ git diff diff --git a/dtype.py b/dtype.py index bfc5b23..63c2376 100644 --- a/dtype.py +++ b/dtype.py @@ -1,2 @@ -print('python') \ No newline at end of file +print('python') +print(list('python')) \ No newline at end of file</p>	<p>git diff: SA =&gt; WD 파일 비교 a/dtype.py(이전 파일): SA 파일 b/dtype.py(이후 파일): WD 파일 @@ -1,2 @@ -이전파일 1: 1줄 +이후파일 1,2: 1줄에서 다음 2줄을 참조</p>

	 <pre> graph TD     HEAD[HEAD] -- "git diff --cached" --&gt; Index[Index]     HEAD -- "git diff HEAD" --&gt; WT[Working tree]     Index -- "git diff" --&gt; WT </pre>	<p>SA(Staging Area, index, staged, indexed)  WD(Working Directory, modified, untracked)  GR(Git Repository. Committed)</p> <p>[git diff를 3가지로 구분]  ❶ git diff  SA =&gt; WD 파일 비교</p> <p>❷ git diff --cached  GR =&gt; SA 파일 비교</p> <p>❸ git diff HEAD  GR =&gt; WD 파일 비교</p>
<div> <div>모듈2</div> <div>실습</div> <div>(20분)</div> </div>	<div> <div>❑ \$ git diff --cached</div> <div>○ diff의 비교 기준(from): 커밋된 파일, diff의 비교의 대상(to): 작업 공간의 파일</div> <div>○ diff의 옵션 --cached의 의미: 비교의 대상(to)이 stage(index) 영역</div> </div>	
	<pre> \$ git diff --cached diff --git a/dtype.py b/dtype.py new file mode 100644 index 0000000..bfc5b23 --- /dev/null +++ b/dtype.py @@ -0,0 +1 @@ +print('python') \ No newline at end of file  \$ git diff HEAD fatal: ambiguous argument 'HEAD': unknown revision or path not in the working tree. Use '--' to separate paths from revisions, like this: 'git &lt;command&gt; [&lt;revision&gt;...] -- [&lt;file&gt;...]' </pre>	<p>커밋을 한 번도 안 한 상태에서 커밋과 SA를 비교  --- /dev/null  커밋은 없고  +++ b/dtype.py  SA에는 1번 줄 있는 파일</p> <p>커밋을 한 번도 안 한 상태에서 HEAD로는 아직 오류</p>
	<pre> \$ git commit -m 'add string type' [main (root-commit) 4dddf86] add string type 1 file changed, 1 insertion(+) create mode 100644 dtype.py  \$ git log1 4dddf86 (HEAD -&gt; main) add string type  \$ git diff --cached  \$ git diff HEAD diff --git a/dtype.py b/dtype.py index bfc5b23..63c2376 100644 --- a/dtype.py +++ b/dtype.py @@ -1,2 @@ -print('python') \ No newline at end of file +print('python') +print(list('python')) \ No newline at end of file </pre>	<p>첫 커밋 후</p> <p>git diff --cached  GR과 SA는 차이가 없으므로 출력 없고</p> <p>git diff HEAD  GR과 WD는 차이가 있으므로 출력</p>
	<pre> \$ git sts M dtype.py  \$ git add .  \$ git sts M dtype.py  \$ code dtype.py print('python') print(list('python')) </pre>	<p>GR, SA, WD를 모두 다르게 해 보기</p> <p>GR  print('python')</p> <p>SA  print('python')  print(list('python'))</p> <p>WD</p>

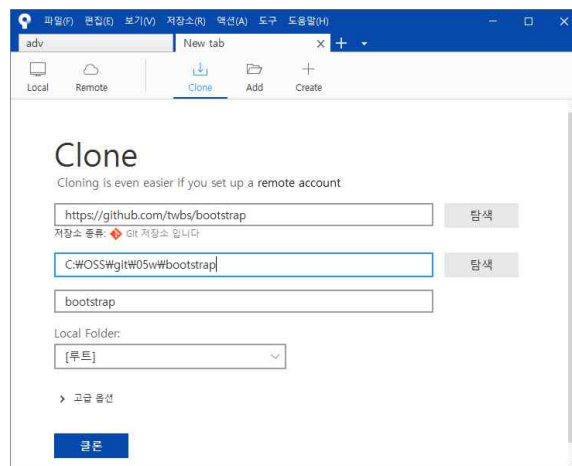
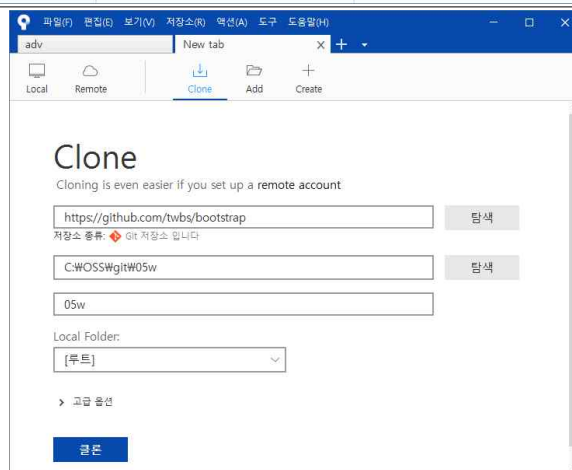
	<pre>print({i:ord(i) for i in 'python'})</pre> <pre>\$ git diff diff --git a/dtype.py b/dtype.py index 63c2376..a8cf6c7 100644 --- a/dtype.py +++ b/dtype.py @@ -1,2 +1,3 @@  print('python') -print(list('python')) \ No newline at end of file +print(list('python')) +print({i:ord(i) for i in 'python'})  \$ git diff --cached diff --git a/dtype.py b/dtype.py index bfc5b23..63c2376 100644 --- a/dtype.py +++ b/dtype.py @@ -1 +1,2 @@ -print('python') \ No newline at end of file +print('python') +print(list('python')) \ No newline at end of file  \$ git diff head diff --git a/dtype.py b/dtype.py index bfc5b23..a8cf6c7 100644 --- a/dtype.py +++ b/dtype.py @@ -1 +1,3 @@ -print('python') \ No newline at end of file +print('python') +print(list('python')) +print({i:ord(i) for i in 'python'})</pre>	<pre>print('python') print(list('python')) print({i:ord(i) for i in 'python'})</pre>  <pre>index 파일id1...파일id2 파일모드(100:일반 파일, 644:rw-r--r-, 110=6 100=4 100=4) index bfc5b23..a8cf6c7 100644</pre>
<p>모듈3 실습 (20분)</p>	<p><input type="checkbox"/> 명령어 diff : from a1 to a2(a1을 기준으로 a2를 비교)에서 to인 a2는 명령어로 정해짐</p> <ul style="list-style-type: none"> <li>○ --cached이면 index, --cached 없이 git diff이면 작업영역(WD)</li> <li>○ from을 HEAD 등으로 기술하면 여러 커밋 중 하나가 가능</li> <li>○ 다만 \$ git diff : SA와 WD를 비교</li> </ul> <p><input type="checkbox"/> \$ git diff head1 head2</p> <ul style="list-style-type: none"> <li>○ 커밋 이력 간, 즉 head1에서 head2의 파일 비교</li> </ul> 	<p>커밋을 총 2번하고 git diff의 5개를 모두 수행</p> <p>HEAD: 가장 마지막 커밋 HEAD~: 가장 마지막 이전 커밋 HEAD^: 가장 마지막 이전 커밋</p>
	<pre>\$ git commit -m 'add list type' [main cfe5341] add list type</pre>	<p>GR의 커밋한 이력 간의 파일 비교</p>

<pre> 1 file changed, 2 insertions(+), 1 deletion(-)  \$ git log1 cfe5341 (HEAD -&gt; main) add list type 4dddf86 add string type  \$ git diff head~ head diff --git a/dtype.py b/dtype.py index bfc5b23..63c2376 100644 --- a/dtype.py +++ b/dtype.py @@ -1,2 @@ -print('python') \ No newline at end of file +print('python') +print(list('python')) \ No newline at end of file </pre>	<pre> \$ git diff 커밋이력1 커밋이력2  GR(HEAD, 최근 커밋) print('python') print(list('python'))  GR(HEAD~, 최근 바로 이전 커밋) print('python')  head^ == head~ </pre>
<pre> \$ git diff head head^ diff --git a/dtype.py b/dtype.py index 63c2376..bfc5b23 100644 --- a/dtype.py +++ b/dtype.py @@ -1,2 @@ -print('python') -print(list('python')) \ No newline at end of file +print('python') \ No newline at end of file </pre>	<pre> head^ == head~  앞 뒤가 바뀌면 이전, 이후 파일이 바뀜 </pre>
<pre> \$ git diff # SA와 WD 비교 diff --git a/dtype.py b/dtype.py index 63c2376..a8cf6c7 100644 --- a/dtype.py +++ b/dtype.py @@ -1,2 @@  print('python') -print(list('python')) \ No newline at end of file +print(list('python')) +print({i:ord(i) for i in 'python'})  \$ git add dtype.py  \$ git diff </pre>	<pre> GR print('python') print(list('python'))  SA print('python') print(list('python'))  WD print('python') print(list('python')) print({i:ord(i) for i in 'python'}) </pre>
<pre> \$ code dtype.py  \$ git diff diff --git a/dtype.py b/dtype.py index a8cf6c7..7c69c05 100644 --- a/dtype.py +++ b/dtype.py @@ -1,3 @@  print('python') print(list('python')) print({i:ord(i) for i in 'python'}) +print({i for i in range(5)})  \$ git diff --staged diff --git a/dtype.py b/dtype.py index 63c2376..a8cf6c7 100644 --- a/dtype.py +++ b/dtype.py @@ -1,3 @@  print('python') -print(list('python')) \ No newline at end of file +print(list('python')) +print({i:ord(i) for i in 'python'})  \$ git diff head diff --git a/dtype.py b/dtype.py index 63c2376..7c69c05 100644 --- a/dtype.py +++ b/dtype.py @@ -1,2 @@  print('python') </pre>	<pre> 마지막 줄에 다음 추가 print({i for i in range(5)})  GR(HEAD) print('python') print(list('python'))  SA print('python') print(list('python')) print({i:ord(i) for i in 'python'})  WD print('python') print(list('python')) print({i:ord(i) for i in 'python'}) print({i for i in range(5)})  \$ git ls-files -s 100644 a8cf6c79fa6e11ae3e78632ebd630e2ada744ffe 0 dtype.py </pre>

	<pre> - print(list('python')) \ No newline at end of file + print(list('python')) + print({i:ord(i) for i in 'python'}) + print({i for i in range(5)})  \$ git diff --cached head~ diff --git a/dtype.py b/dtype.py index bfc5b23..a8cf6c7 100644 --- a/dtype.py +++ b/dtype.py @@ -1,3 @@ - print('python') \ No newline at end of file + print('python') + print(list('python')) + print({i:ord(i) for i in 'python'})  \$ git diff head~ diff --git a/dtype.py b/dtype.py index bfc5b23..7c69c05 100644 --- a/dtype.py +++ b/dtype.py @@ -1,4 @@ - print('python') \ No newline at end of file + print('python') + print(list('python')) + print({i:ord(i) for i in 'python'}) + print({i for i in range(5)}) </pre>	<p>GR, SA, WD를 모두 다른 상태</p> <p>GR(HEAD)  <pre> print('python') print(list('python')) </pre> </p> <p>GR(HEAD~)  <pre> print('python') </pre> </p> <p>SA  <pre> print('python') print(list('python')) print({i:ord(i) for i in 'python'}) </pre> </p> <p>WD  <pre> print('python') print(list('python')) print({i:ord(i) for i in 'python'}) print({i for i in range(5)}) </pre> </p> <p>커밋 간의 파일 비교  <pre> \$ git diff [cmtID1] [cmtID2] </pre> <p>이전 커밋과 최종 커밋의 파일 비교  <pre> \$ git diff HEAD^ HEAD </pre> </p></p>
<p>모듈4 실습 (20분)</p>	<p><input type="checkbox"/> 소스트리에서 이미 생성된 저장소 살펴 보기</p> <p><input type="checkbox"/> 소스트리에서 유명 오픈소스 저장소 복제(clone)하고 살펴 보기</p> <ul style="list-style-type: none"> <li>○ 부트스트랩 <a href="https://github.com/twbs/bootstrap">https://github.com/twbs/bootstrap</a></li> <li>○ 텐서플로 <a href="https://github.com/tensorflow/tensorflow">https://github.com/tensorflow/tensorflow</a></li> <li>○ 깃 <a href="https://github.com/git/git">https://github.com/git/git</a></li> <li>○ 리눅스 <a href="https://github.com/torvalds/linux">https://github.com/torvalds/linux</a></li> <li>○ 리액트 <a href="https://github.com/facebook/react">https://github.com/facebook/react</a></li> <li>○ vscode <a href="https://github.com/microsoft/vscode">https://github.com/microsoft/vscode</a></li> </ul> 	<p>New Tab에서 Add 클릭 탐색에서 생성된 저장소 선택</p> <p>git show에서 보듯이 현재 선택된 커밋과 바로 이전 커밋간의 파일</p>



비교를 바로 볼 수 있음



두 번째 행에서 폴더를 지정하고  
직접 하부 폴더로 이름 Wbootstrap을  
지정해 줌  
자동으로 그 하부는 이름이  
bootstrap으로 입력됨

참조: <https://ux.stories.pe.kr/181>

The screenshot displays the VS Code interface with a Git commit history on the left and a code diff on the right. The commit history shows a series of commits by GeoSot, including 'Add eslint-plugin-html to lint JS in', 'Update devDependencies', and 'Ref: add explicit imports on ou'. The code diff on the right shows changes to 'js/tests/unit/tab.spec.js', specifically adding a new 'tab3' element and updating the 'show' method to handle it.