



DoiT Web Project Team

5. Git & GitHub



Git



- Version Control System (VCS)
- 파일의 버전 관리를 위한 **분산 시스템**.
- 리눅스를 만든 Linus Torvalds가 개발.



- git을 위한 **호스팅 서비스**. (Git 자체와는 다름.)
- git 프로젝트를 관리할 수 있음.
- GitHub 외에도 GitLab, Bitbucket 등이 있다.

Git이 왜 필요한가요?

- 문제가 생기면 때 과거 특정 시점으로 되돌릴 수 있다.

Commits on Jul 19, 2018

Merge branch 'master' of https://github.com/AjouDoiT-WebProj/timetable ...

ParkSB committed on 19 Jul

dc9dc49

Commits on Jul 18, 2018

setting.php 생성 K-osA committed on 18 Jul	f8b4531
이름/비밀번호 변경 틀 제작 K-osA committed on 18 Jul	b85a916
style 태그 삭제 K-osA committed on 18 Jul	01e6ca8
setting.js 생성 K-osA committed on 18 Jul	e1c95f1
css, php 파일 분리 K-osA committed on 18 Jul	73dcf0c
new file: src/php/_index.php moggie617 committed on 18 Jul	f6ced84
newteam - 멤버 찾는 php 생성 jmchoi1225 committed on 18 Jul	2eae18a

Git이 왜 필요한가요?

- 여러 사람이 쉽게 코드를 공유하고, 충돌을 줄일 수 있다.

26 ■■■■ src/js/teamlist.js

```
...    ...    @@ -1,19 +1,17 @@
1      - import $ from "jquery";
2      - import "../css/teamlist.css";
3      -
4      - $.ajax({ // AJAX 통신을 이용
5      -   type: "POST",
6      -   url: "../src/php/_teamlist.php", // teamlist.php에 접근
7      -   dataType: "JSON", // JSON 형식으로 수신
8      -   success: function(result) { // 성공적으로 수신한 경우
9      -     renderList(result);
10     -   }
11     - });
12
13     + import $ from 'jquery';
14     + import '../css/teamlist.css';
15
16     3
17     4   function renderList(list) {
18     5     for (let i = 0; i < list.length; i += 1) {
19
20       -   $("#team-list").append(
21       -     `<li><a href="./teamtable.html?i=${list[i].id}>${list[i].name}</a></li>`
22       -   );
23     +   $('#team-list').append(`<li><a href="./teamtable.html?i=${list[i].id}>${list[i].name}</a></li>`);
24
25     6   }
26     7
27     8   }
```

Git이 왜 필요한가요?

- 코드 리뷰하기 편하다.



ParkSB reviewed on 31 Aug

[View changes](#)

코멘트 남겼으니 수정하고 테스트해보세요!

src/php/_setting.php

Outdated

```
62 + $currpwquery = "SELECT password FROM user WHERE id = $id";
63 + $currpwresult = mysqli_query($conn, $currpwquery);
64 +
65 + if($currentpassword != $pw){
```



ParkSB on 31 Aug Member

제가 지금 구동을 못해서 확실치는 않은데, \$pw는 암호화된 값이고 \$currentpassword는 평문이라 항상 true가 나올 것 같아요



K-osA on 31 Aug Member

앗 제가 푸시를 잘못해서 git reset했었는데 58~60줄에 current,want,checkpassword 다 md5 씌워준게 날아갔네요 그런데 이때도 작동이 안됐어가지고요....




Reply...

Resolve conversation






1. 프로젝트 만들기
2. 로컬 저장소 만들기
3. 코드 수정하기
4. 원격 저장소에 코드 올리기
5. 원격 저장소에서 코드 가져오기
6. 브랜치 만들기

프로젝트 만들기



[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)




New repository
Import repository
New gist
New organization

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner


Repository name

 ParkSB ▾


 /

Great repository names are short and memorable. Need inspiration? How about [stunning-happiness](#).

Description (optional)

☒  **Public**

Anyone can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.


☐ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

 |

Add a license: **None** ▾



Create repository



로컬에 저장소 가져오기

- 저장소(Repository): 파일을 저장하는 공간.
- 로컬 환경(Local Env.): 내 컴퓨터.

```
git clone https://github.com/user/project.git
```

- GitHub에서 만든 프로젝트의 URL을 `clone` 하면 컴퓨터에 로컬 저장소(Local repository)가 만들어진다.

코드 수정하기

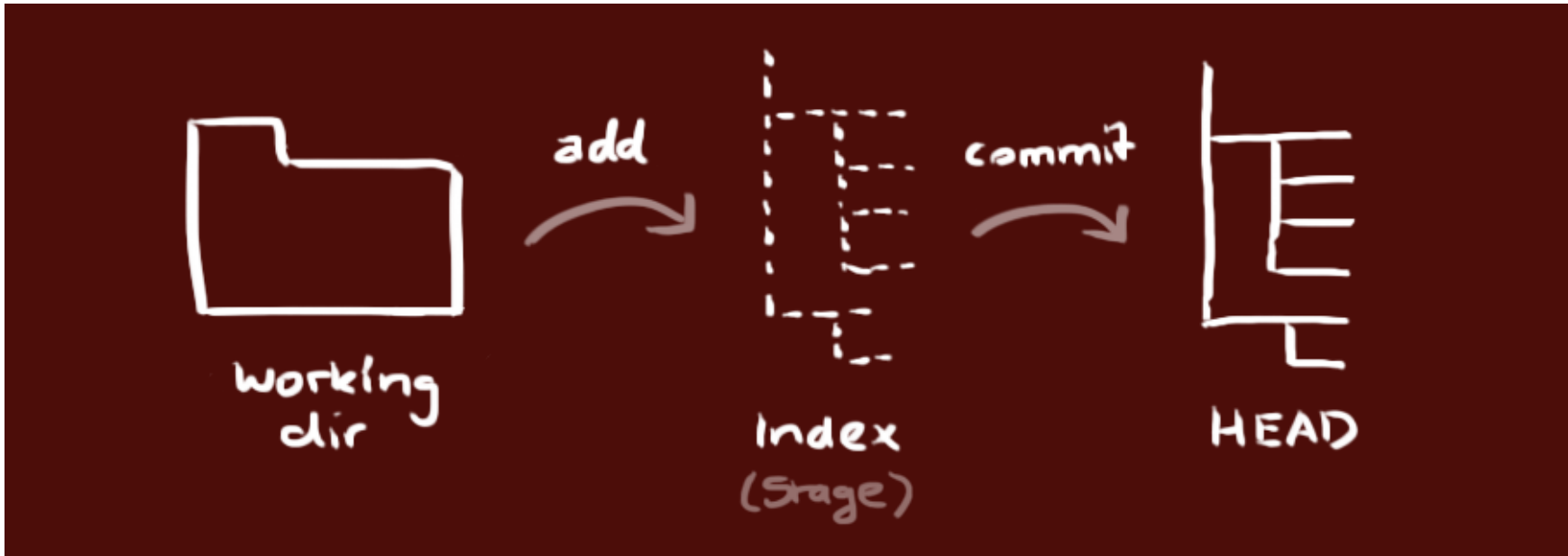
- 로컬 저장소는 아직 누구에게도 공유되지 않은 상태.
- 마음대로 파일을 수정해도 다른 사람에게 반영되지 않는다.

원격 저장소에 코드 올리기

- 원격 저장소(Remote repository): GitHub에 호스팅된 저장소.
- 로컬 저장소에서 원격 저장소로 파일을 올려보자.

원격 저장소에 코드 올리기

로컬 저장소 (Working dir, Index, HEAD)



- 작업 디렉토리(Working directory): 파일이 있는 디렉토리.
- 인덱스(Index): 올릴 준비가 된 파일들만 모인 영역.
- 헤드(HEAD): 최종으로 올려진 확정 파일들이 모인 영역.

원격 저장소에 코드 올리기

1. Staging

- a.html, b.html을 수정했지만 a.html만 올리고 싶을 때.

```
git add a.html
```

- a.html이 작업 디렉토리에서 인덱스로 스테이징(Staging)된다.

원격 저장소에 코드 올리기

2. Commit

- 스테이징된 파일들을 올리는 단계.

```
git commit -m "commit message"
```

- a.html이 인덱스에서 헤드로 커밋(Commit)된다.

원격 저장소에 코드 올리기

3. Push

- 커밋한 파일들을 최종적으로 올리는 단계.

```
git push origin master
```

- a.html이 헤드에서 원격 저장소로 푸시(Push)된다.

원격 저장소에서 코드 가져오기

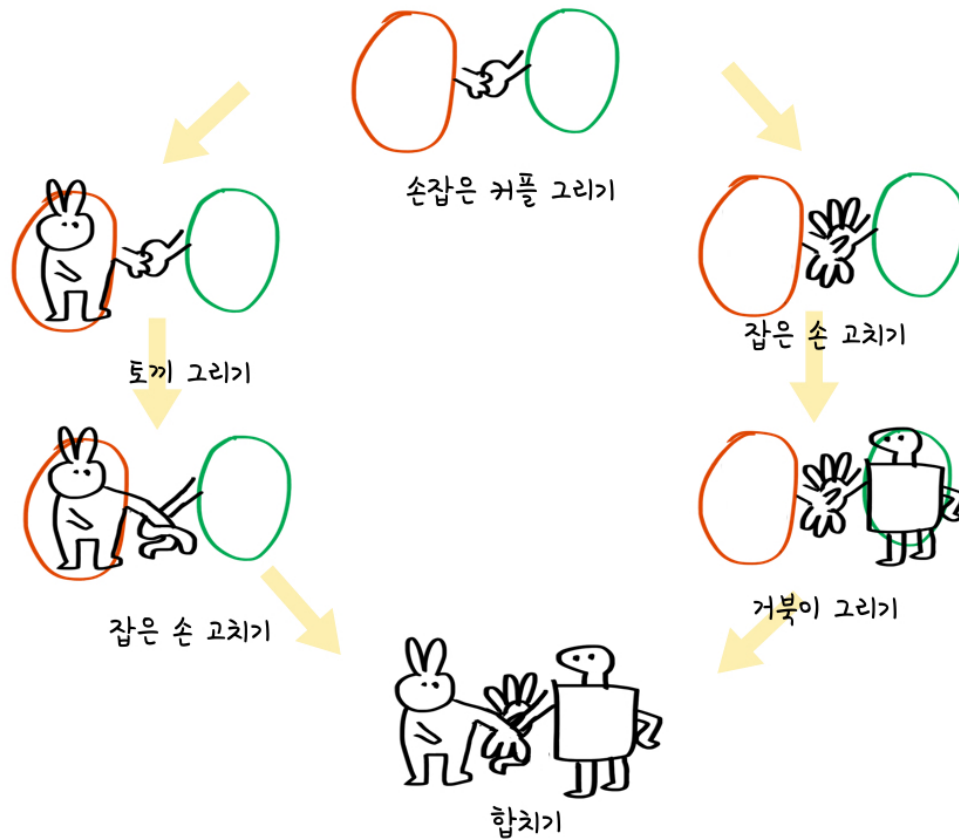
- 누군가 원격 저장소에 코드를 수정해 올린 상황.
- 로컬 저장소를 최신으로 업데이트해줘야 한다.

```
git pull
```

- 원격 저장소의 변경 내용이 작업 디렉토리에 받아지고(Fetch), 병합(Merge)된다.

코드를 올리거나 가져올 때

- ! 동일한 부분이 수정됐다면 충돌(Conflict)이 발생한다 !



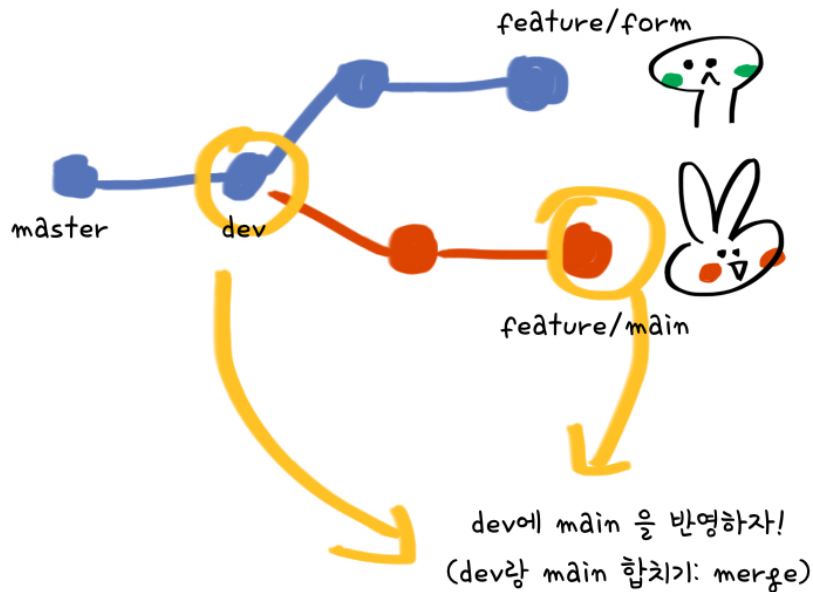
브랜치 만들기

- 충돌을 피하기 위해 만드는 가지(Branch).
- 충돌이 발생하면 어떤 브랜치를 적용할지 선택할 수 있다.

브랜치 만들기

- master 브랜치에서 dev 브랜치를 땀다.
- 토끼와 거북이가 dev 에서 각자 브랜치를 만들어 개발.

origin/feature/form	feature/form	달리기 신청 폼 제작	f16dab4
		달사 타이틀 다정하게 수정	65e13f8
feature/main	origin/feature/main	달사 타이틀 시크하게 수정	9aa0077
		메인 화면에 달사 설명 추가	627619d
origin/dev	dev	배대 만들기	0b2f9f4
origin/master	master	README 파일 추가	a556be5



브랜치 만들기

1. Create new branch

- `develop` 브랜치 만들기:

```
git checkout -b develop
```

- 원격 저장소에 `develop` 브랜치 올리기:

```
git push origin develop
```

- `master` 브랜치로 돌아가기:

```
git checkout master
```

브랜치 만들기

2. Create pull request

- 한 브랜치를 다른 브랜치에 합치기 위해 요청하는 것.
- 다른 사람에게 리뷰를 받을 수 있다.

브랜치 만들기

3. Merge pull request

- 풀 리퀘스트를 반영하는 것.
- 충돌이 발생하면 수정 사항을 반영할지, 유지할지 선택.

좋은 커밋 메시지 작성하기

Keyword: Description

- "feat: 아이템 구입 기능 구현"
- "fix: 아이템 중복 구입 문제 해결"
- "refactor: 이중루프 로직 개선"
- "chore: 불필요한 파일 제거"
- "docs: 설명서 오타 수정"

✧ 깔끔한 브랜치 관리

- `master` 브랜치는 '배포가능한 상태'로 유지.
- 실제 개발은 `develop` 브랜치에서 진행.
- `develop` 이 배포가능한 상태가 되면 `master` 로 병합.
- (여럿이 개발하는 경우) 기능별로 브랜치를 나누고, 기능을 완성하면 `develop` 브랜치에 병합.

Git GUI

- CLI가 익숙치 않다면 [Source Tree](#)나 [Sublime Merge](#)를 사용.
- vscode에서도 GUI 기반으로 git을 사용할 수 있다.