



**GitHub 를 활용한
프로젝트 환경구축하기**



git 저장소 호스팅을 지원하는 웹 서비스로
온라인 상에서 여러 사용자들이 컴퓨터 파일을
공유하도록 도와준다

GitHub의 장점

소스 코드를 주고 받을 필요 없이,
같은 파일을 여러 명이 동시에 작업하는
병렬 개발이 가능

GitHub의 장점

프로젝트 관리, 배포, 이슈 추적 기능등
프로젝트 진행에 있어 편리한 기능들을 지원

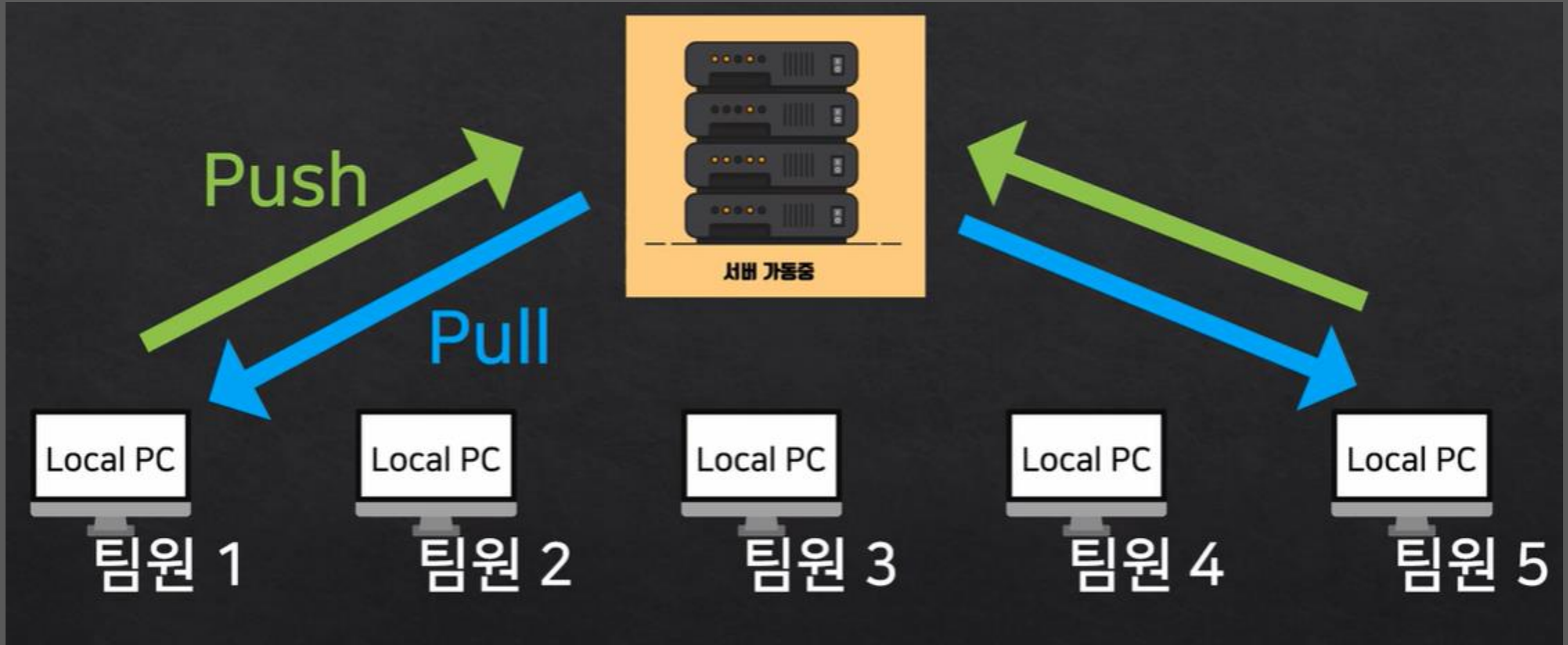
GitHub와 Git의 차이점

GitHub – 소스코드를 올리는 어떤 공간(저장소)

Git – 소스코드를 내 컴퓨터에서 인터넷으로
올려주는 역할

(로컬에서 버전 관리를 할 수 있게 도와주는 도구)

GitHub의 원리




GitHub의 필요성

- 재택근무, 비대면 작업이 활성화 되는 요즘
온라인 협업도구의 필요성 증가
- GitHub 에 남겨진 프로젝트 결과물들이 포트폴리오 역할을
을 하고 동시에 그 사람의 작업 능력을 증명
- 협업 능력을 중요시 하는 요즘 GitHub 사용 경험은 필수


GitHub활용시 꿀팁!

master ▾ 2 branches 0 tags Go to file Add file ▾ Code ▾

 Chickenater


검색 기능 과거 데이터 필터링 추가 -준현

39ea4a9 2 hours ago ⌚ 88 commits

 Golf


검색 기능 과거 데이터 필터링 추가 -준현

2 hours ago

 README.md

Update README.md

13 days ago

README.md 

QRmoney

- QR+=money 의 프로젝트
- 1. 매일 아침 9시(주말포함) pull 받고 개발 들어가기
- 2. 아침 회의때 각자 당일 수정할 파일 정하기
- 3. 만약에 다른 사람이 수정하는 파일을 수정해야할 경우에는 그 사람에게 말하기
- 4. push는 자신의 하루 분량이 끝났을때만 push하기
- 5. push하기전에 팀 독방에 말하고 push하기, 말할때는 개발한 기능과 이름을 명시 ex) 로그인 기능 버그 수정 - 준현
- 6. push 코멘트는 자기가 개발한 기능과 자신의 이름 적기 ex) 로그인 기능 버그 수정 - 준현

GitHub활용시 지켜야 하는 규칙

- 수정을 할 때 **같은 파일을 동시에 수정하지 않기!**
- pull, push 작업을 할때 꼭 commit 을 통해서 **local repository** 최신화 시켜주기!
- 개발을 한 후 '**commit -> push**' 할 때 꼭!
팀원들에게 알려주기

- 순서는
- pull -> add -> commit -> push
- Push 할 때는 push 한다고 팀원들한테 말해주기
- 무조건 pull 하고 push (습관화 하기)

실습

1. 원격 저장소는 이미 만들어져 있음
원래는 처음에 저장소 먼저 만들어야 함
(팀원 초대까지)
2. 초기 코드 작성해서 팀장이 master에 push 함
3. 팀원1이 pull 받음

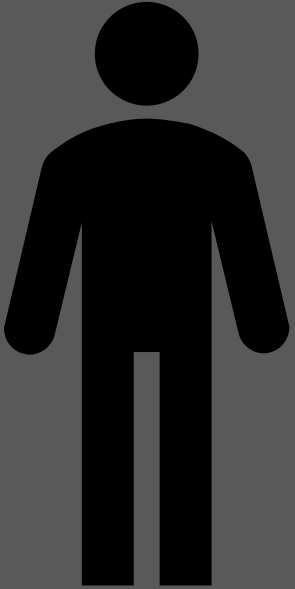
4. 팀원2 pull 하기
5. 팀원3 pull 하기
6. 팀원1이 코드 작성하고 push하기
7. 팀원1이 작성한 코드 pull 하기
8. 팀원2 코드 작성하고 push
9. 팀원2이 작성한 코드 pull
10. 팀원 3도 똑같이

새로운 실습

- 프로젝트 파일 새로 만들어서 해보기
- 프로젝트 파일로 해봤으면 새로운 워크스페이스(새폴더) 만들어서 해보기
- 순서는
 - pull -> add -> commit -> push
 - Push 할 때는 push 한다고 팀원들한테 말해주기
- 무조건 pull 하고 push

충돌이란?

운비(Join)

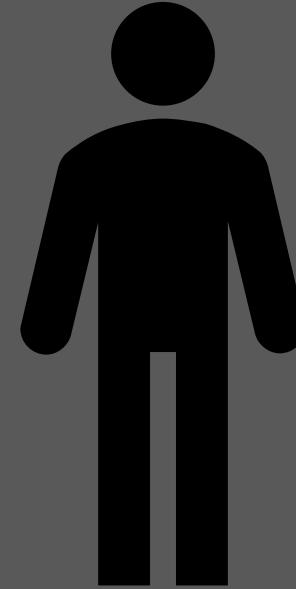


```
class A{
```

```
10 - - - -Join()
```

```
}
```

승환(Login)



```
class A{
```

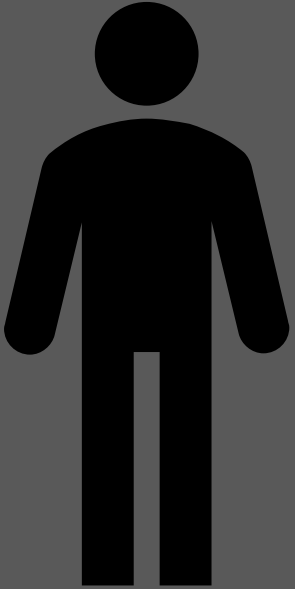
```
10 - - - - Login()
```

```
}
```

GitHub

충돌이란?

운비(Join)

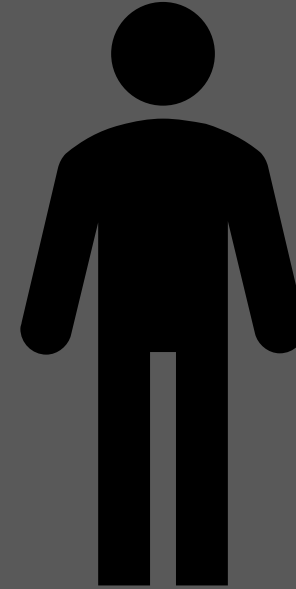


```
class A{
```

```
10 - - Join()*Login()  
// error 충돌! 발생!
```

```
}
```

승환(Login)



```
class A{
```

```
10 - - - - Login()
```

```
}
```

GitHub