GIT 튜토리얼

Git /GitHub 를 이용한

- 1. 소스 관리와
- 2. 협업 방법론

관련 자료 링크

누구나 쉽게 이해할 수 있는 Git 입문

https://backlog.com/git-tutorial/kr/

초심자를 위한 Github 협업

 https://milooy.wordpress.com/2017/06/21/working-together-with-github-t utorial/

Git cheat sheet

•

Git 사용 Flow

- 1. 새 프로젝트 생성 후 프로젝트 폴더에 git 설정
- 2. 개발진행 & commit
- 3. 초기버젼 완성
- 4. 새 기능 추가 팀에겐 feature branch 를, 버그 수정팀에겐 bugfix branch 를 주어 개발 진행
- 5. 각 branch 개발이 끝나면 master branch 로 merge (통합)

Git 설치 / GITHUB 계정등록

https://git-scm.com/

윈도우용 혹은 MAC 용을 다운 받은 후 설치

GitHub 에 계정등록 / 로그인

```
git
    <<git-scm.com 다운>>
    git init
    git config --global user.name "eventia"
    git config --global user.email "myEmail@gmail.com"
    git status
    git add --all .
    git commit -m "설명"
    <<github 계정 생성>>
    github.com 에서 repository 생성 (myrepository)
    <<github 저장>>
    git remote add origin https://github.com/mygitid/myreposit.git
    git push -u origin master
```

실습 1 - 준비

- git 을 설치한다.
- gitHub 에 계정을 등록한다.

실습 2 - 초기설정

```
최초 실행시
깃이 설치되어 있는지 확인
$ git -v
```

깃/깃허브 설정

```
$ git config --global user.name "eventia"
$ git config --global user.email "myEmail@gmail.com"
```

실습 3 - 온라인저장소(repo) 만들기

GitHub 에서 new repository 실행

실습 4 - 로칼저장소 만들기

- 1. PC 에 디렉토리 trygit 을 만든다.
- 2. 마우스 우클릭으로 "Git Bash Here" 을 실행
- 3. 그 위치에 편집기를 열어 텍스트 파일을 만든다.
 - a. babyshark.txt 만든다.
- 4. 가사를 넣는다.
 - a. babyshark.txt 를 편집기로 열어 적당한 가사를 넣는다.
- 5. command 창에서

```
$ git init
```

실습 5 - 로칼저장소 깃 실행

- 1. git init 을 실행했으면
- 2. 추가할 파일을 git add 명령으로 추가한다.
 - \$ git add babyshark.txt
- 3. 추가가 잘 되었다면 커밋한다.
 - \$ git commit -m "Add MyFirst Commit"

실습 6 - 로칼저장소와 깃허브 연결

1. 로칼저장소(PC의 디렉토리/폴더)와 깃허브의 repo 를 연결한다.

\$ git remote add origin 깃허브-레포지토리이름.git

2. 확인을 위해 다음을 입력한다.

\$ git remote -v

3. GitHub 로 변경사항을 업로드한다.

\$ git push -u origin master

실습 7 - 저장하지 않을 파일 선택

- 1. ".gitignore" 파일을 만들다.
- 2. 파일 내부에 저장하지 않고 무시할 파일을 적는다.

참고사이트

- https://jazzodevlab.tistory.com/50
- https://velog.io/@dsunni/Spring-properties-STS-.gitIgnore-%ED%8C%8C%EC%9D%
 BC-%EA%B4%80%EB%A6%AC

실습 8 - 이전으로 돌아갈 때

해시값 확인 : 지금부터 이전 4개까지만

- > git log -4
- > git reset --hard

#hashnumber = 3f690fc313e1...0f4865f375fa4f6

- > git reset --hard 3f690fc313e1...0f4865f375fa4f6
- > git push -f

GitHub 협업 튜토리얼

https://github.com/ludenscode/TestGitCoWork.git

- 프로젝트 디렉토리 만들고
 그 안에서 [Git Bash Here] 실행
- 3. \$ git clone https://github.com/ludenscode/TestGitCoWork.git
- 4. cd TestGitCoWork
- 5. git checkout dev
- 6.

단일저장소 협업방법

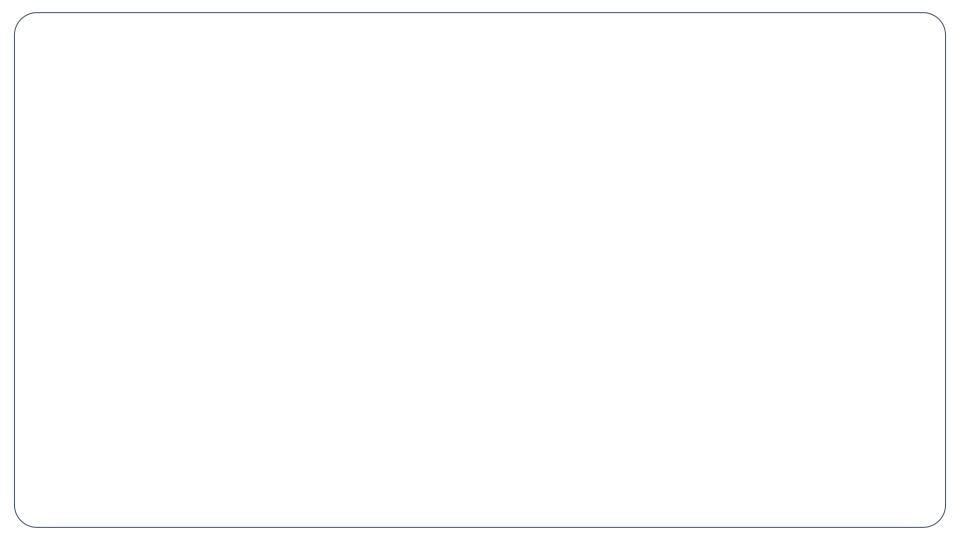
- 1. GitHub 에서 a. + 아이콘을 클릭, New Organization 을 만든다.
- 2. repo 를 생성한다.
- 3. Collaborator 을 초대한다.
 - a. 초대받은 사람은 알림이 뜨고, 초대를 수락하면 권한이 주어진다.
 - b. git clone [github주소.git] 입력

단일저장소 협업방법

- 4. 만든 이를 M 이라하고 초대받은 이를 S 라 하자.
 - M 과 S는 브랜치를 만든다.
 - \$ git branch dev
 - \$ git checkout dev
 - \$ git push -u origin dev
- 5. S 가 git pull 을 하고, dev 로 브랜치를 이동한다.
 - \$ git pull
 - \$ git checkout dev
- 6. branch 를 지울때는 git branch -d <브랜치이름>

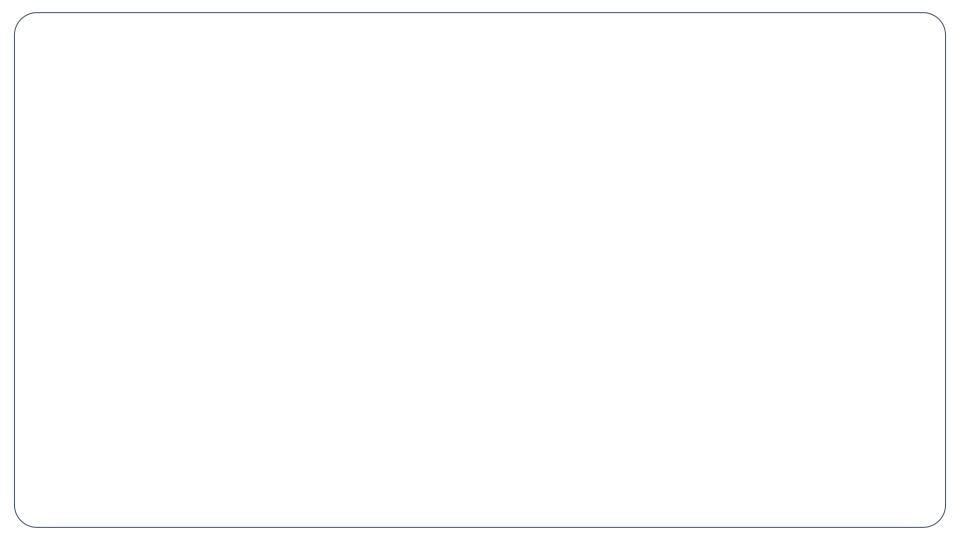
단일저장소 협업방법

- 7. M 은 feature/main 브랜치로, S 는 feature/form 브랜치로 개발
- 8. (M 또는 S가) Pull Request 로 리뷰를 받고 합치기 요청
- 9. (상대편 S 또는 M이) Merge pull Request 로 합치기 실행
- 10. Merge 에서 Master 로
- 11. (필요하면) releases 로 압축파일로 만듦



Git cheat sheet

```
git init
git status
git config --global user.name "사용자이름"
git config --global user.email "사용자email"
git add --all .
git add <파일이름>
git commit -m "설명"
github.com <<github 계정 생성 & repository 생성>>
git remote add origin <git repo 주소>
git pull
git push -u origin master
git push -u origin <브랜치이름>
git clone <git repo 주소>
git branch <브랜치이름>
git checkout <브랜치이름>
gitk
```



git / github 에러

Git / GitHub 에러 처리

git fatal: remote origin already exists. 에러

git / github 배치파일

배치파일 처리

배치파일 처리

```
echo [PROCESSING...] git pull
git pull
echo [PROCESSING...] git add --all .
git add --all .
if [%1] EQU [] (
echo.
echo [PROCESSING...] git commit -m "Dev JAVA Web Project %date%"
echo.
git commit -m "Dev JAVA Web Project %date%-[%time:~0,5%]"
) else (
echo.
echo [PROCESSING...] git commit -m %1
echo.
git commit -m %1
echo.
echo [PROCESSING...] git push
echo.
```

git push

배치파일 최소형

```
git pull
git add --all .
git commit -m "HTML Test Project %date%-[%time:~0,5%]"
git push
```

다른 PC에 git 설치 후

> git clone https://자신의깃허브repository

GIT 튜토리얼

Git /GitHub 를 이용한

- 1. 소스 관리와
- 2. 협업 방법론

관련 자료 링크

누구나 쉽게 이해할 수 있는 Git 입문

https://backlog.com/git-tutorial/kr/

초심자를 위한 Github 협업

 https://milooy.wordpress.com/2017/06/21/working-together-with-github-t utorial/

Git cheat sheet

•

Git 사용 Flow

- 1. 새 프로젝트 생성 후 프로젝트 폴더에 git 설정
- 2. 개발진행 & commit
- 3. 초기버젼 완성
- 4. 새 기능 추가 팀에겐 feature branch 를, 버그 수정팀에겐 bugfix branch 를 주어 개발 진행
- 5. 각 branch 개발이 끝나면 master branch 로 merge (통합)

Git 설치 / GITHUB 계정등록

https://git-scm.com/

윈도우용 혹은 MAC 용을 다운 받은 후 설치

GitHub 에 계정등록 / 로그인

```
git
    <<git-scm.com 다운>>
    git init
    git config --global user.name "eventia"
    git config --global user.email "myEmail@gmail.com"
    git status
    git add --all .
    git commit -m "설명"
    <<github 계정 생성>>
    github.com 에서 repository 생성 (myrepository)
    <<github 저장>>
    git remote add origin https://github.com/mygitid/myreposit.git
    git push -u origin master
```

실습 1 - 준비

- git 을 설치한다.
- gitHub 에 계정을 등록한다.

실습 2 - 초기설정

```
최초실행시

깃이설치되어 있는지 확인

$ git -v

깃/깃허브설정

$ git config --global user.name "eventia"

$ git config --global user.email "myEmail@gmail.com"
```

실습 3 - 온라인저장소(repo) 만들기

GitHub 에서 new repository 실행

실습 4 - 로칼저장소 만들기

- 1. PC 에 디렉토리 trygit 을 만든다.
- 2. 마우스 우클릭으로 "Git Bash Here" 을 실행
- 3. 그 위치에 편집기를 열어 텍스트 파일을 만든다.
 - a. babyshark.txt 만든다.
- 4. 가사를 넣는다.
 - a. babyshark.txt 를 편집기로 열어 적당한 가사를 넣는다.
- 5. command 창에서

```
$ git init
```

실습 5 - 로칼저장소 깃 실행

- 1. git init 을 실행했으면
- 2. 추가할 파일을 git add 명령으로 추가한다.
 - \$ git add babyshark.txt
- 3. 추가가 잘 되었다면 커밋한다.
 - \$ git commit -m "Add MyFirst Commit"

실습 6 - 로칼저장소와 깃허브 연결

1. 로칼저장소(PC의 디렉토리/폴더)와 깃허브의 repo 를 연결한다.

\$ git remote add origin 깃허브-레포지토리이름.git

2. 확인을 위해 다음을 입력한다.

\$ git remote -v

3. GitHub 로 변경사항을 업로드한다.

\$ git push -u origin master

실습 7 - 저장하지 않을 파일 선택

- 1. ".gitignore" 파일을 만들다.
- 2. 파일 내부에 저장하지 않고 무시할 파일을 적는다.

참고사이트

- https://jazzodevlab.tistory.com/50
- https://velog.io/@dsunni/Spring-properties-STS-.gitIgnore-%ED%8C%8C%EC%9D%
 BC-%EA%B4%80%EB%A6%AC

실습 8 - 이전으로 돌아갈 때

해시값 확인 : 지금부터 이전 4개까지만

- > git log -4
- > git reset --hard

#hashnumber = 3f690fc313e1...0f4865f375fa4f6

- > git reset --hard 3f690fc313e1...0f4865f375fa4f6
- > git push -f

GitHub 협업 튜토리얼

https://github.com/ludenscode/TestGitCoWork.git

- 프로젝트 디렉토리 만들고
 그 안에서 [Git Bash Here] 실행
- 3. \$ git clone https://github.com/ludenscode/TestGitCoWork.git
- 4. cd TestGitCoWork
- 5. git checkout dev
- 6.

단일저장소 협업방법

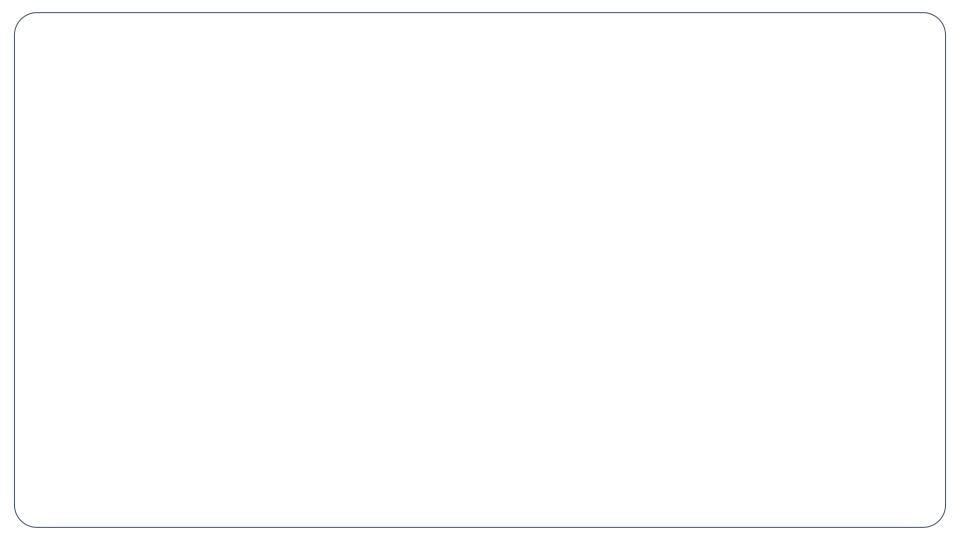
- 1. GitHub 에서
 - a. + 아이콘을 클릭, New Organization 을 만든다.
- 2. repo 를 생성한다.
- 3. Collaborator 을 초대한다.
 - a. 초대받은 사람은 알림이 뜨고, 초대를 수락하면 권한이 주어진다.
 - b. git clone [github주소.git] 입력

단일저장소 협업방법

- 만든 이를 M 이라하고 초대받은 이를 S 라 하자.
 M 과 S는 브랜치를 만든다.
 - \$ git branch dev
 - \$ git checkout dev
 - \$ git push -u origin dev
- 5. S 가 git pull 을 하고, dev 로 브랜치를 이동한다.
 - \$ git pull
 - \$ git checkout dev
- 6. branch 를 지울때는 git branch -d <브랜치이름>

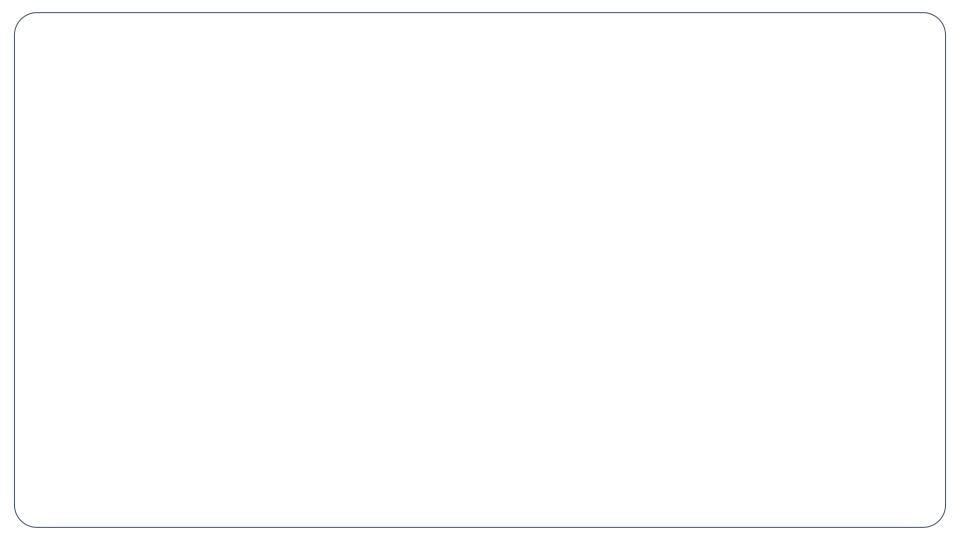
단일저장소 협업방법

- 7. M 은 feature/main 브랜치로, S 는 feature/form 브랜치로 개발
- 8. (M 또는 S가) Pull Request 로 리뷰를 받고 합치기 요청
- 9. (상대편 S 또는 M이) Merge pull Request 로 합치기 실행
- 10. Merge 에서 Master 로
- 11. (필요하면) releases 로 압축파일로 만듦



Git cheat sheet

```
git init
git status
git config --global user.name "사용자이름"
git config --global user.email "사용자email"
git add --all .
git add <파일이름>
git commit -m "설명"
github.com <<github 계정 생성 & repository 생성>>
git remote add origin <git repo 주소>
git pull
git push -u origin master
git push -u origin <브랜치이름>
git clone <git repo 주소>
git branch <브랜치이름>
git checkout <브랜치이름>
gitk
```



git / github 에러

Git / GitHub 에러 처리

git fatal: remote origin already exists. 에러

git / github 배치파일

배치파일 처리

배치파일 처리

```
echo [PROCESSING...] git pull
git pull
echo [PROCESSING...] git add --all .
git add --all .
if [%1] EQU [] (
echo.
echo [PROCESSING...] git commit -m "Dev JAVA Web Project %date%"
echo.
git commit -m "Dev JAVA Web Project %date%-[%time:~0,5%]"
) else (
echo.
echo [PROCESSING...] git commit -m %1
echo.
git commit -m %1
echo.
echo [PROCESSING...] git push
echo.
```

git push

배치파일 최소형

```
git pull
git add --all .
git commit -m "HTML Test Project %date%-[%time:~0,5%]"
git push
```

다른 PC에 git 설치 후

> git clone https://자신의깃허브repository