회의록

1. **회의 개요**

|  |  |  |  |
| --- | --- | --- | --- |
| 회 의 명 | git설치 및 실습 | | |
| 일 시 | 2018.01.09 (15:00~16:00) | 장 소 | 리딩라운지 |
| 참 석 자 | 심수현, 배세은, 이연재 | 작성자 | 심수현, 배세은 |

1. **회의 내용**

|  |  |
| --- | --- |
| **Title** | **Contents** |
| **회의내용**  **및**  **결과** | 1. 실습 매뉴얼 구글 docs에서 참고 (하단에 내용 첨부하였음) [https://drive.google.com/drive/u/1/folders/1fG1kCcysjutgD7C0LUsL5QDXbcrgNGX6](file://https://drive.google.com/drive/u/1/folders/1fG1kCcysjutgD7C0LUsL5QDXbcrgNGX6)    1. Git   소스코드를 효과적으로 관리하기 위해 개발된 분산형 버전 관리 시스템.  원래는 Linux 소스코드를 관리할 목적으로 개발.  소스코드가 변경된 이력을 쉽게 확인할 수 있고  특정 시점에 저장된 버전과 비교하거나 특정 시점으로 되돌아 갈 수도 있음.   * 1. Git Clone   원격 저장소를 복제하려면 클론(Clone)이라는 조작을 수행함.  복제란 원격 저장소의 내용을 통째로 다운로드 하는 것을 말함.  복제한 저장소를 다른 PC에서 로컬 저장소로 사용할 수 있게 됨.   * 1. Git Pull   원격 저장소에서 로컬 저장소로 업데이트하려면 풀(Pull)을 실행함.  Pull을 실행하면 원격 저장소에서 최신 변경 이력을 다운로드 하여  내 로컬 저장소에 그 내용을 적용함.   * 1. Git Add   커밋할 파일을 지정하는 git명령어  폴더안에 있는 파일들을 git으로 등록하기 전에 사용   * 1. Git Commit   폴더에 새로운 파일을 추가하고 원격 저장소에 파일을 등록함.   * 1. Git Push   웹 상의 원격 저장소로 변경된 파일을 업로드 하는 것을 푸시(Push)라고 함.  Push를 실행하면, 원격 저장소에 내 변경 이력이 업로드 되어  원격 저장소와 로컬 저장소가 동일한 상태가 됨.   * 1. Branch 모든 버전 관리 시스템은 브랜치를 지원.  개발을 하다 보면 코드를 여러 개로 복사해야 하는 일이 자주 생긴다. 코드를 통째로 복사하고 나서 원래 코드와는 상관없이 독립적으로 개발을 진행할 수 있는데, 이렇게 독립적으로 개발하는 것.   2. Git Branch / Checkout 내가 사용 할 브랜치를 지정하는 것   3. Pull request GitHub의 아주 강력한 협업 기능 작업의 완료를 위한 과정이 아닌, 새로운 작업의 시작을 위한 기능 작업의 히스토리 관리에도 매우 효율적 |
| **기타사항** | 1. 다음 스터디 1/19( 수요일날 다시 공지 ) 2. 1/8 ~ 1/19 정경환 훈련소 일정있음 3. 1/15 배세은 출국 |

### 참고 사이트

<http://all-record.tistory.com/161> (gitHub가입)  
[https://backlog.com/git-tutorial/kr/](file://https://backlog.com/git-tutorial/kr/) (사용법)  
[https://opentutorials.org/course/1492](file://https://opentutorials.org/course/1492)(sourceTree를 이용한 방법... 생활코딩 강좌)

### 실습환경

1. 커맨드라인(cmd)이용 / git add , git push 등 명령어 사용
2. sourceTree같이 GUI로 기능을 조작할 수 있는 프로그램 사용

각자 편한 방법을 선택해서 실습

### 기본 명령어 실습

1. gitHub repository를 local repository(내 컴퓨터 저장소)에 clone 하기
   1. git clone
   2. [https://github.com/yjcolin34/test-repository](file://https://github.com/yjcolin34/test-repository) (미리 만들어둔 저장소 사용)
2. git pull
3. git add
   1. 구글드라이브에서 html파일을 받고(내 local 저장소에), 파일 이름을 다음과 같이 수정한다. (hello\_"내 이름적기".html )
   2. 파일을 열어 안녕하세요 *\*\*\** 입니다. 작성하고 저장
   3. git add 하여 stage에 올리기 ( 해당 파일을 이제 버전관리하겠다는 단계)
4. git commit
   1. "내 이름 first commit" 커밋메세지 작성
5. git push
   1. 원격 저장소에 코드를 동기화
   2. gitHub에 커밋과 파일이 잘 올라갔는지 확인

### Branch와 pull request

1. git checkout test-branch (새로만든 branch 이름입니다.) master branch(현재 branch)에서 새로운 branch로 변경
2. 각자 요청 받은 pull request를 확인하고 , 문제를 해결한 뒤 파일을 수정하여 push합니다. (Americano, FlatWhite, Latte를 지우고 , " Edits by 본인이름" 을 추가해주세요 )
3. pull request를 해결한 후 master branch로 merge시킵니다.

### 회의록 작성

[https://drive.google.com/drive/u/1/folders/1fG1kCcysjutgD7C0LUsL5QDXbcrgNGX6](file://https://drive.google.com/drive/u/1/folders/1fG1kCcysjutgD7C0LUsL5QDXbcrgNGX6)

1. 다음과 같은 내용에 대하여 개념정리 , 명령어 혹은 실습 내용 등
   1. git이란
   2. git clone
   3. git pull
   4. git add
   5. git commit
   6. git push
   7. branch란
   8. git branch / checkout
   9. pull request에 대하여
2. 다 작성하고 위 링크(구글드라이브)에 업로드