GIT(cli 사용법 요약)

git이란?

리누스토발즈에의해 개발된 버전관리 소프트웨어

협업사용시 또는 파일작업시 일어나는 파일간의 여러 시간적 상황적 문제들을 체크하고 관리할 수 있는 기능

---

git 설치

1. http://github.com 가입(계정은 구글로 가입하는것을 권함)
2. http://git-scm.com 에서 설치
3. gui사용자의 경우는 별도의 gui프로그램을 설치

기본 용어

1. **커멘드라인(command line => cli)** : 윈도우에선 **cmd**, 맥에선 **terminal** 라고 불리우며, 마우스로 사용하는것이 아닌 프롬프트화면을 이용하여 키보드 명령어를 통해 사용한다.
2. **저장소(repository):** 깃 프로젝트 작업시 데이터를 저장 할 수 있는 공간(작업공간)
3. **버전관리(version control):** 깃의 목적 파일을 사용시 겹쳐쓰거나, 어려버전으로 나뉘어서 사용할경우 시간별로 버전별로 분류하여 사용할 수있다.
4. **커밋(commit):** 깃에 자료를 올릴때 해당 부분의 삽입하는 메세지, 해당 작업시 사용되었던 내용에대해 이해할 수 있도록 설명해놓는다.
5. **브랜치(branch):** 하나의 프로젝트 사용시 여러명이 사용하거나, 버전을 별도로 분루해서 사용할경우 사용한다.(`branch`를 사용할경우 다시 메인(`master`)과 합칠경우 `merge` 기능을 사용한다.)

사용방법

1. git에서 로그인 후, 새로운 저장소를 생성(git자체를 개인사이트로 생성할 경우는 별도 참조)한다.
2. 내컴퓨터에 깃생성한 저장소와 동일한 이름의 폴더를 생성한다.
3. 해당 폴더에서 **terminal**(window 에서는 **git-bash**)을 실행 한다.
4. git이 설치되었는지 확인한다.

```
git --verion
```

5. **git init:** 내컴퓨터의 깃 계정을 사용할차례이니 이제 깃을 사용할 수 있도록 기초세팅한다(초기화)

```
git init
```

6. **git config:** 'configure'의 줄임말로, 처음 깃을 설정할때

```
git config --global user.name 사용자이름
git config --global user.email 사용자이메일
```

7. 내용을 작업한다.(md파일이든 html문서든 작업한다.)
8. **git status :** glt의 현재상태를 확인, push 하지않은 자료, commit 처리하지 않은 자료 등 전송 전후의 상태를 확인하기위해 사용함

```
git status
```

9. **git add :**
   git에 자료를 올리기위해 첨부할 파일을 선택하는것

```
git add test.css      // test.css 파일을 올리기위해 선택
```

10. **git commit :** git에 자료를 올리는 시점에대한 설명글을 적는곳, 작업시점에대한 설명이 있어야 차후의 협업시 빠르게 작업처리가 가능

```
git commit -m "올릴 메세지 작성"      // -m의 표현은 메세지를 작성하겠다고 처리하는 뜻
```

11. **git push :** git에 자료를 내보내기(해당 서버 주소에 자료가 올라감), 정확한 주소값이 첨부되어야함, 처음에는 계정아이디 및 비밀번호를 질문할 수 있으며 이후에는 명령어 만으로도 자료가 전송되어짐

```
git push    // git을 올리기 위한 주소가 첨부되었을경우에는 문제없이 올라감.
git push https://github.io/test.git      // git push 뒤에 올릴 위치값을 입력
```

12. **git pull :** 자료를 올린 후 다운받기위해 사용하는 명령어, 차후 다른컴퓨터에서 사용하거나 할때 유용하게 쓸 수 있음

```
git pull
```

13. git clone: git에서 생성한 repository를 내컴퓨터에 저장할 때 사용(기존에 내용이 많다면)

```
git clone git계정주소.git
```

14. git branch : git에 분리되어진 branch(작업과정)을 확인하는 명령어, 한번의 push이후 별도의 branch를 생성하여 작업할 수 있다.

```
git branch
```

15. git branch test : git에 별도의 branch를 분리해서 생성하는 방법(test는 임의로 만든 이름)

```
git branch test
```

16. git checkout test : git 위치를 test로 이동

```
git checkout test
```

깃 자료를 올리기위한 프로젝트 저장소 설정 **git remote add origin 깃주소**를 입력한다.

```
git remote add origin git계정주소.git
```

최초에 깃 자료 올릴때에는 **git push -u origin master** 를 입력한다.

```
git push -u origin master
```