

4주차 : Git, 웹 프레임워크

- + 게시판 update
 - update.php (사용자가 정보를 수정하는 페이지)
1. 내용입력 이전

포카칩

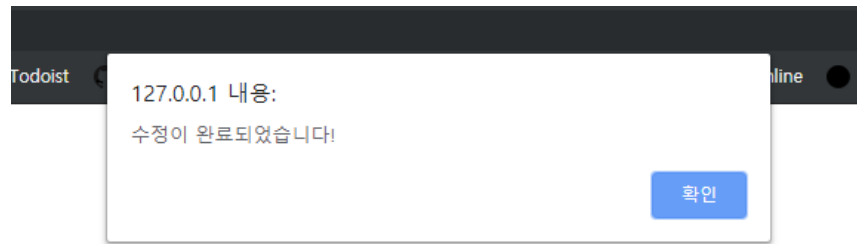
| | |
|-----|---|
| 과자명 | 포카칩 |
| 원산지 | 충적북도 청주시 흥덕구 |
| 제조사 | 오리온 |
| 칼로리 | 583 |
| 가 격 | 1840 |
| 성 분 | 생감자, 혼합식용유, 염파맛씨즈닝, 우유, 대두, 밀, |

2. 내용입력 이후

포카칩

| | |
|-----|--|
| 과자명 | 포카칩 |
| 원산지 | 충적북도 청주시 흥덕구 |
| 제조사 | 오리온 |
| 칼로리 | 583 |
| 가 격 | 1840 |
| 성 분 | 생감자, 혼합식용유, 염파맛씨즈닝, 우유, 대두, 밀, <u>dddddddd</u> |

3. 정상적으로 수정완료시 알림창



4. 알림창의 확인 버튼 누른 후 홈화면(snack.php)으로 이동

과자 등록 목록

| 번호 | 과자명 | 날짜 | 조회수 |
|----|---------|---------------------|-----|
| 1 | 고북칩136g | 2019-08-02 20:33:13 | 1 |
| 2 | 바나나킥75g | 2019-08-02 21:10:09 | 1 |
| 3 | 스윙칩 | 2019-08-02 21:14:23 | 0 |
| 4 | 포카칩 | 2019-08-02 21:24:17 | 0 |
| 5 | 허니버터칩 | 2019-07-27 18:43:18 | 3 |

제품등록하기

* 내용이 수정된 과자 '포카칩'의 등록 날짜와 조회수가 변경됨

- 날짜 = NOW()
- 조회수 = 0

5. 이미 등록된 과자명으로 수정하는 경우

- 포카칩을 이미 등록된 과자명인 '스윙칩'으로 바꾸려는 경우

포카칩

| | |
|-----|--|
| 과자명 | 스웨치 |
| 원산지 | 충청북도 청주시 흥덕구 |
| 제조사 | 오리온 |
| 칼로리 | 583 |
| 가 격 | 1840 |
| 성 분 | 생감자, 혼합식용유, 양파맛씨즈닝, 우유, 대두, 밀, ddddddd |

수정

127.0.0.1 내용:

수정된 과자이름이 이미 존재합니다!

확인

* 위와 같은 경고창이 뜨고 확인버튼을 누르면 다음과 같은 코드에 의해 이전 페이지로 돌아간다.

```
echo "<script>alert(W"수정된 과자이름이 이미  
존재 합니다!W");history.back();</script>";
```

- update_collect.php(수정된 정보를 처리하는 script)
 - 다른 등록된 과자명과 수정한 과자명이 일치하는 경우를 확인

```
// snack_name 중복여부 판단
$query_overlap = "SELECT snack_name FROM snack_table_list WHERE snack_name <>
'$_POST[previous_snackName]';";

$result_is_overlap = mysqli_query($connect,$query_overlap);
$is_overlap = 0;
while($arr_is_overlap = mysqli_fetch_array($result_is_overlap)){
  if($arr_is_overlap['snack_name']==$_POST['update_snackName']){
    $is_overlap += 1;
  }
}
```

- 수정 대상(\$_POST[previous_snackName])을 제외한 모든 snack_name를 SELECT로 뽑은 후 사용자가 수정한 과자명과 동일한 이름을 snack_name의 컬럼값으로 가질 때마다 횟수(\$is_overlap)를 센다.
- 동일한 이름이 나오지 않은 경우 횟수(\$is_overlap)은 0이기 때문에 횟수가 0인 경우는 정상적으로 수정작업을 수행하고 횟수가 0이 아닌 경우에는 경고창을 띄우고 사용자가 다시 수정값을 입력할 수 있게 이전 수정페이지로 이동한다.

- UPDATE를 통해 각 테이블 정보 수정

+ snack_table_list, snack_table_almost

```
$query_update_table_almost =
"UPDATE snack_table_almost
SET snack_name=$_POST[update_snackName],origin=$_POST[update_origin],
manufacturer=$_POST[update_manufacturer]',
calorie=$_POST[update_calorie]', price=$_POST[update_price]'
WHERE
snack_name=$_POST[previous_snackName]";

$query_update_table_list =
"UPDATE snack_table_list
SET snack_name=$_POST[update_snackName]', time=NOW(), hit=0
WHERE snack_name=$_POST[previous_snackName]";

mysqli_query($connect,$query_update_table_almost);
mysqli_query($connect,$query_update_table_list);
```

+ snack_table_ingredient

- 해당 table의 경우, 수정된 ingredient 정보 갯수에 의해 기존의 tuple 갯수를 줄이거나 늘려야 하는 case가 발생
 - 수정된 ingredient 정보 split, 기존 tuple의 컬럼인 id값을 array로 가져오기

```
// ingredient parsing
$array_ingredient = explode("",".PHP_EOL,$_POST['update_ingredient']");
$array_ingredient = array_filter($array_ingredient);
$size_array_ingredient = count($array_ingredient);
// Bring id which is in tuple to be updated
$query_id_list = "SELECT id FROM snack_table_ingredient
WHERE snack_name=$_POST[previous_snackName]";
$result_id_list = mysqli_query($connect,$query_id_list);
$id_list=array();
$count_id=mysqli_num_rows($result_id_list);
for($i=0;$i<$count_id;$i++){
    $id_list[$i]=(int)mysqli_fetch_row($result_id_list)[0];
}
```

- 수정된 ingredient 정보 갯수 <= 기존의 tuple 갯수
 - 기존의 tuple에 수정된 정보를 update한 후 남은 tuple을 삭제

```
if($size_array_ingredient<=$count_id){
  for($i=0;$i<$size_array_ingredient;$i++){
    $query_update_table_ingredient =
      "UPDATE snack_table_ingredient
      SET
        snack_name='$_POST[update_snackName]',
        snack_ingredient='$array_ingredient[$i]'
        WHERE id='$id_list[$i]';";
    mysqli_query($connect,$query_update_table_ingredient);
  }
  for($i=$size_array_ingredient;$i<$count_id;$i++){
    $query_delete_remainder=
      "DELETE from snack_table_ingredient WHERE id='$id_list[$i]';";
    mysqli_query($connect,$query_delete_remainder);
  }
}
```

- 수정된 ingredient 정보 갯수 > 기존의 tuple 갯수
 - 기존 tuple에 수정된 ingredient를 update한 후 입력되지 못한 ingredient를 insert

```
else{
  for($i=0;$i<$count_id;$i++){
    $query_update_table_ingredient = "UPDATE snack_table_ingredient
      SET snack_name='$_POST[update_snackName]',
        snack_ingredient='$array_ingredient[$i]'
        WHERE id='$id_list[$i]';";
    mysqli_query($connect,$query_update_table_ingredient);
  }
  for($i=$count_id;$i<$size_array_ingredient;$i++){
    $query_insert_remainder = "INSERT INTO snack_table_ingredient(snack_name,snack_ingredient)
      VALUES('$_POST[update_snackName]','$array_ingredient[$i]');";
    mysqli_query($connect,$query_insert_remainder);
  }
}
```

● Git

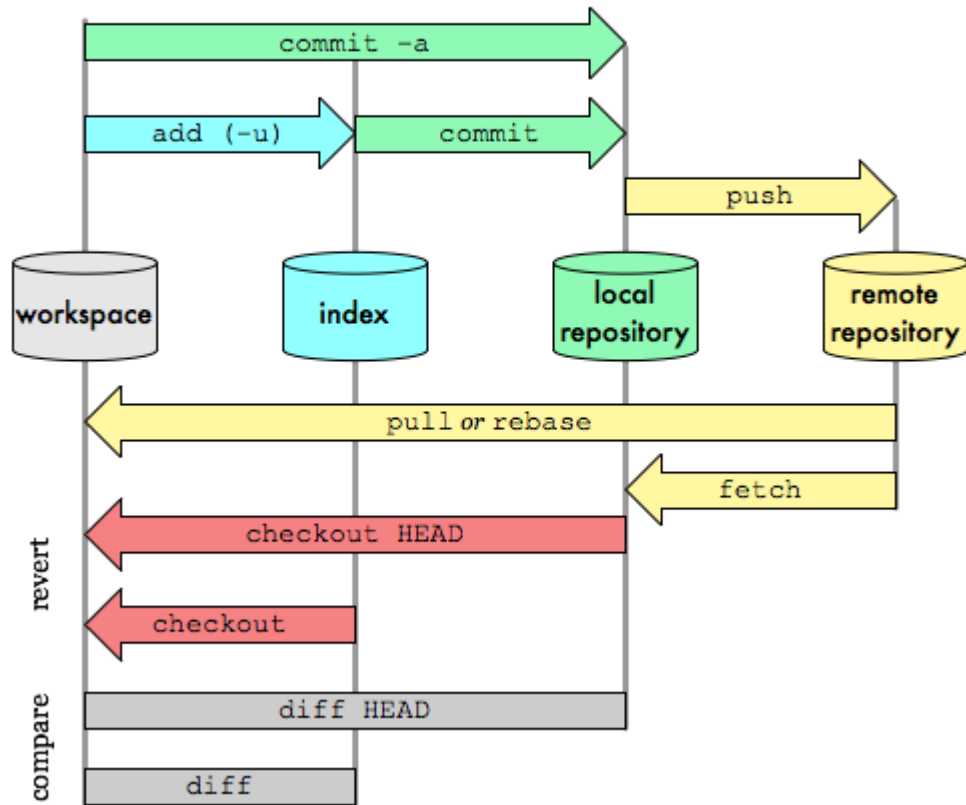
a. Git이란?

- 컴퓨터 파일을 작업단위로 관리할 수 있게 해주고 협업시에 여러 명의 사용자들이 한 작업을 조율할 수 있게 해주는 분산 버전 관리 시스템

b. Git의 내부 구조

Git Data Transport Commands

<http://osteale.com>



- workspace
 - 최신 commit에서 수정된 파일과 디렉토리의 내용을 저장하는 저장소(일반 디렉토리)
- index
 - workspace에서 수정된 작업을 add하여 commit을 등록할 준비가 완료시킨 작업들을 저장하는 저장소
- local repository
 - workspace와 같은 PC에서 commit별(변경 이력별)로 파일이나 디렉토리를 저장하는 저장소(.git)
- remote repository
 - 원격 저장소
 - 작업을 수행하지 못하고 local repository로 commit된 내용을 등록시키는 곳으로 저장 기능만을 가짐
 - git init --bare 로 설정해야함
(=.git만이 존재하게 설정)

c. commit

- 프로젝트를 변경 이력 단위로 저장하는 것

- commit을 하기 전 `git add filename`으로 파일을 관리 대상으로 등록시켜야 함.
 - 한번도 등록하지 않은 파일은 untracked 상태이다.
 - 등록 후 파일은 tracked상태가 된다
 - 후에 commit후 파일을 수정한 경우, modified 상태가 되서 다시 재등록을 시키고 commit을 해야한다.
 - 상태는 `git status`로 확인가능

1. snack_project를 working directory로 지정

```
정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/snack_project
$ git init
Initialized empty Git repository in C:/Users/정현석/study_programming/BCSD/documents/snack_project/.git/

정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/snack_project
(master)
$ ls -al
total 8
drwxr-xr-x 1 정현석 197609 0  8월  4 17:57 ./
drwxr-xr-x 1 정현석 197609 0  8월  4 16:49 ../
drwxr-xr-x 1 정현석 197609 0  8월  4 17:57 .git/
```

2. 관리할 파일들을 add

```
정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/snack_project
(master)
$ ls -al
total 16
drwxr-xr-x 1 정현석 197609 0  8월  4 17:58 ./
drwxr-xr-x 1 정현석 197609 0  8월  4 16:49 ../
drwxr-xr-x 1 정현석 197609 0  8월  4 18:01 .git/
-rw-r--r-- 1 정현석 197609 3625 7월 27 20:41 snack.php
drwxr-xr-x 1 정현석 197609 0  8월  4 17:58 snackF/

정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/snack_project
(master)
$ git add snack.php

정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/snack_project
(master)
$ git add snackF/

정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/snack_project
(master)
$ git status
On branch master

no commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   snack.php
        new file:   snackF/board_style.css
        new file:   snackF/delete.php
        new file:   snackF/update.php
        new file:   snackF/update_collect.php
        new file:   snackF/view.php
        new file:   snackF/write.php
        new file:   snackF/write_collect.php
```

3. 현재 add된 파일들을 commit

```
정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/snack_project
(master)
$ git commit -m "first-commit"
[master (root-commit) 85287bb] first-commit
8 files changed, 471 insertions(+)
create mode 100644 snack.php
create mode 100644 snackF/board_style.css
create mode 100644 snackF/delete.php
create mode 100644 snackF/update.php
create mode 100644 snackF/update_collect.php
create mode 100644 snackF/view.php
create mode 100644 snackF/write.php
create mode 100644 snackF/write_collect.php
```

```

정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/snack_project (master)
$ git log
commit 85287bb67350130b9c1e8af49eb4b2b4ed62ce27 (HEAD -> master)
Author: HyunseokJung <ghi02124@gmail.com>
Date: Sun Aug 4 18:15:06 2019 +0900

first-commit

```

d. branch

- 나뉘어가지가 여러 갈래로 나뉘어져 있는 것처럼 프로젝트의 버전관리를 나누어 관리할 수 있게 해주는 것
- 기본적으로 master branch가 설정되어 있다.
- branch 생성은 `git branch 브랜치명`으로 가능하다.
 - 생성된 branch는 생성 기점 기준 최신 commit을 가리킨다.
- `git checkout 브랜치명` 을 통해 branch를 변경할 수 있다.

1. branch 생성(branchName -> addFunction)

```

정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/snack_project (master)
$ git branch
* master

정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/snack_project (master)
$ git checkout -b addFunction
Switched to a new branch 'addFunction'

정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/snack_project (addFunction)
$ git branch
* addFunction
  master

```

2. 작업 후 파일의 상태, addFunction branch에서 commit

```

정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/snack_project (addFunction)
$ git status
On branch addFunction
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   author.php

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   snack.php

```

```

정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/snack_project (addFunction)
$ git commit -am "second-commit:Create button which shows author_data"
[addFunction 14d1d0f] second-commit:Create button which shows author_data
2 files changed, 28 insertions(+)
create mode 100644 author.php

정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/snack_project (addFunction)
$ git log
commit 14d1d0f4cb427ad04953da335fb9b763685e5560 (HEAD -> addFunction)
Author: HyunseokJung <ghi02124@gmail.com>
Date: Sun Aug 4 19:22:50 2019 +0900

    second-commit:Create button which shows author_data

commit 85287bb67350130b9c1e8af49eb4b2b4ed62ce27 (master)
Author: HyunseokJung <ghi02124@gmail.com>
Date: Sun Aug 4 18:15:06 2019 +0900

first-commit

```


e. merge

- branch로 나뉜 commit 작업들을 합치는 것
- merge시에 conflict(충돌)이 발생할 수 있다.
 - conflict(충돌)
 - merge 과정에서 파일의 이름이 같으면 충돌이 발생한다.
 - 파일이 다르면 무조건 자동으로 합쳐준다.
 - 파일이 같아도 수정한 부분이 다르다면 자동으로 합쳐준다.

1. addFunction branch를 master branch에 merge

```
정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/snack_project (master)
$ git merge addFunction
Updating 85287bb..14d1d0f
Fast-forward
 author.php | 25 +++++
 snack.php  | 3 +++
 2 files changed, 28 insertions(+)
 create mode 100644 author.php

정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/snack_project (master)
$ git log --decorate --graph
* commit 14d1d0f4cb427ad04953da335fb9b763685e5560 (HEAD -> master, addFunction)
 | Author: HyunseokJung <ghi02124@gmail.com>
 | Date: Sun Aug 4 19:22:50 2019 +0900
 |
 | second-commit:Create button which shows author_data
 |
 * commit 85287bb67350130b9c1e8af49eb4b2b4ed62ce27
 | Author: HyunseokJung <ghi02124@gmail.com>
 | Date: Sun Aug 4 18:15:06 2019 +0900
 |
 | first-commit
```

2. 필요 없는 branch 삭제

```
정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/snack_project (master)
$ git branch -d addFunction
Deleted branch addFunction (was 14d1d0f).

정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/snack_project (master)
$ git log --decorate --graph
* commit 14d1d0f4cb427ad04953da335fb9b763685e5560 (HEAD -> master)
 | Author: HyunseokJung <ghi02124@gmail.com>
 | Date: Sun Aug 4 19:22:50 2019 +0900
 |
 | second-commit:Create button which shows author_data
 |
 * commit 85287bb67350130b9c1e8af49eb4b2b4ed62ce27
 | Author: HyunseokJung <ghi02124@gmail.com>
 | Date: Sun Aug 4 18:15:06 2019 +0900
 |
 | first-commit
```

f. push

- local repository에 있는 데이터를 remote repository로 업로드시킬 수 있다.

- push 하기 전 remote repository와 local repository를 연결해야한다.

```
git remote add origin [local경로(pwd) or 도메인주소]
```

- repository에 올리는 코드는 아래와 같다.

```
git push -u origin master(branch명)
```

1. 원격 repository 지정

```
정 현 석 @DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/snack_project (master)  
$ git remote add origin https://github.com/HyunSeokJung-student/snack_project.git
```

2. push(origin에 master 등록)

```
정 현 석 @DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/snack_project (master)  
$ git push -u origin master  
Enumerating objects: 15, done.  
Counting objects: 100% (15/15), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (15/15), done.  
Writing objects: 100% (15/15), 6.78 KiB | 578.00 KiB/s, done.  
Total 15 (delta 3), reused 0 (delta 0)  
remote: Resolving deltas: 100% (3/3), done.  
To https://github.com/HyunSeokJung-student/snack_project.git  
* [new branch]      master -> master  
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

3. push 이진 github

- + github 링크: https://github.com/HyunSeokJung-student/snack_project

[Tests](#)
[Issues](#)
[Marketplace](#)
[Explore](#)

[HyunSeokJung-student / snack_project](#)

Unwatch 1
Star 0
Fork 0

Code
Issues 0
Pull requests 0
Projects 0
Wiki
Security
Insights
Settings

Quick setup — if you've done this kind of thing before

Set up in Desktop
or
HTTPS SSH
https://github.com/HyunSeokJung-student/snack_project.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```

echo "# snack_project" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/HyunSeokJung-student/snack_project.git
git push -u origin master

```

...or push an existing repository from the command line

```

git remote add origin https://github.com/HyunSeokJung-student/snack_project.git
git push -u origin master

```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

ProTip! Use the URL for this page when adding GitHub as a remote.

4. push 이후 github

[HyunSeokJung-student / snack_project](#)

Unwatch 1
Star 0
Fork 0

Code
Issues 0
Pull requests 0
Projects 0
Wiki
Security
Insights
Settings

No description, website, or topics provided.
Edit

Manage topics

2 commits
1 branch
0 releases
0 contributors

Branch: master
New pull request
Create new file
Upload files
Find File
Clone or download

HyunseokJung
second-commit:Create button which shows author_data
Latest commit 14d1d0f 19 minutes ago

| | | |
|------------|---|----------------|
| snackF | first-commit | 1 hour ago |
| author.php | second-commit:Create button which shows author_data | 19 minutes ago |
| snack.php | second-commit:Create button which shows author_data | 19 minutes ago |

Help people interested in this repository understand your project by adding a README.
Add a README



g. pull

- remote repository에 있는 데이터를 local repository로 다운로드 시킬 수 있다.

```
git pull origin master
```

+ origin을 master에 내려받음을 뜻함

1. 새로운 gitfth 폴더 생성

```
정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/gitfth
$ git init
Initialized empty Git repository in C:/Users/정현석/study_programming/BCSD/documents/gitfth/.git/

정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/gitfth (master)
$ git remote add origin https://github.com/HyunSeokJung-student/snack_project.git

정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/gitfth (master)
$ ls -al
total 4
drwxr-xr-x 1 정현석 197609 0 8월 4 19:57 ./
drwxr-xr-x 1 정현석 197609 0 8월 4 19:55 ../
drwxr-xr-x 1 정현석 197609 0 8월 4 19:57 .git/
```

2. gitfth에 github에 있는 내용 내려받기

```
정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/gitfth (master)
$ git pull origin master
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 15 (delta 3), reused 15 (delta 3), pack-reused 0
Unpacking objects: 100% (15/15), done.
From https://github.com/HyunSeokJung-student/snack_project
 * branch            master       -> FETCH_HEAD
 * [new branch]      master       -> origin/master

정현석@DESKTOP-CBRNN22 MINGW64 ~/study_programming/BCSD/documents/gitfth (master)
$ ls -al
total 16
drwxr-xr-x 1 정현석 197609 0 8월 4 19:57 ./
drwxr-xr-x 1 정현석 197609 0 8월 4 19:55 ../
drwxr-xr-x 1 정현석 197609 0 8월 4 19:57 .git/
-rw-r--r-- 1 정현석 197609 814 8월 4 19:57 author.php
-rw-r--r-- 1 정현석 197609 3761 8월 4 19:57 snack.php
drwxr-xr-x 1 정현석 197609 0 8월 4 19:57 snackF/
```

\$ mkdir gitfth

\$ cd gitfth

\$ ls -al : 현재 폴더의 파일 목록을 보여주는 명령문

workspace 지정 : (clone) or (init)

버전관리를 할 폴더로 이동한다.

init을 통해 현재 작업할 장소를 git에게 알려줌 : git init

+ .git이 생성됨 -> 버전관리정보를 저장하는 곳

file을 관리 대상으로 등록하기(git은 기본적으로 새로운 파일을 관리하지 x)

vim f1.txt (file을 생성함과 동시에 편집하게 됨)

i를 누르면 insert가 창에 나오면서 file을 편집할 수 있게 됨

esc를 다시 누르면 vim 명령어 치는 모드로 바뀜
편집한 내용을 파일에 쓰고 싶으면 :w , 작업을 종료하고 싶으면 q를
입력하고

enter

- + vim 명령어
- + i : insert mode
- + h,j,k,l : 방향키
- + o : 개행 후 insert mode 진입
- + shift + o : 이전 행에서 개행 후 insert mode 진입
- + shift + h : 문서의 처음으로 커서 이동
- + shift + l : 문서의 마지막 라인의 처음으로 커서 이동

file의 내용을 읽고 싶을 때

cat f1.txt

*status

- 위에 vim으로 생성해도 아직 버전관리 대상이 되지 않음 (= Untracked files)
- git add f1.txt (git 의 관리 대상이 됨)
- 임시적으로 필요한 파일들은 (ex - 테스트용) 관리 대상으로 지정하지 않음

commit(버전 만들기)

- + version: 의미있는 변화단계
(어떤작업이 있으면 작업이 완결된 상태를 version이라고함)
- + 자신이 만든 버전이라고 명시하기 위해 이름 setting
 - + git config --global user.name nickname
 - + git config --global user.email egoing@naver.com

git commit 이라고 git bash 창에 입력

-> vim이 실행

version message 입력

- + git log 버전 만든사람과 만든 시간 나옴
- file을 수정할 경우 status가 modified 됨.
 - 해당 파일로 version 생성하기 위해서 git add f1.txt 를 다시 해야함
 - 왜 이과정이 있는가?
 - 변경된 사항이 여러 개일 때, 선택적으로 file을 commit해서보다 더 버전을 세부적으로 관리할 수있게됨

버전관리를 하는 효용은 수정된 내용을 추적해서 문제해결을 하는데 이용하기 위해서이다.
또다른 효용은 과거의 상태로 돌아갈 수 있다는 점이다.

-> 버전간의 차이점을 확인하는 방법

- git log -p

- + 현재 버전에서 일어난 일 , - 이전 버전에서 일어난 일
 - git log (commit_id) : commit_id에 해당되는 버전과 이전의 버전 간의 차이 출력
 - git diff commit_id1..commit_id2
 - commit_id1과 commit_id2 사이에 생긴 일을 추적해서 출력해줌
 - 수정된 내용이 있는 경우 commit하기 전에 git diff 로 작성자가 수정한 내용이 이전 버전과 비교해서 잘못 된 경우가 있는지 확인하게 해줌
- > 과거의 버전으로 돌아갈수 있다 (reset)
- reset
 - git reset version_id_3 --hard
 - version이 id_3,2,1이 남아있음을 확인할 수 있다.
 - 버린 것처럼 보이지만 남아있어서 복구도 가능하다.
 - 원격으로 공유하기 전에만 할 것
 - revert
 - commit을 취소하고 새로운 commit을 만들
- + git 명령어
- + commit , push, pull, clone, checkout, add, branch, log, diff, fetch, merge, init,
 - + status, reset, tag, rebase, rm, show, bisect, grep, mv
- + git commit --help
- + option 확인
- 내부원리
 - add의 원리
 - git은 object로 내용을 저장하고 메모리주소를 보여주는데 이때 내용이 동일하면 똑같은 메모리주소를 사용(중복으로 인한 메모리 낭비를 방지)
 - ex) cp f1.txt f3.txt -> 이 때 f1과 f3가 저장된 메모리주소가 동일함
 - 내용의 값을 sha1으로 해쉬값을 만들기 때문에 같은 주소값을 가지게 만들 수 있다.
 - commit의 원리
 - tree(commit한 시점) / parent(이전commit)
 - objects -> blob(filename,filecontent) or tree(commit시점의 file들) or commit(tree+ parent+ ..)
 - status의 원리
 - index와 commit objects의 차이가 존재하면 수정된 값이 있음을 알 수 있음

working directory - index(stage:commit 대기상태), staging area, cache - repository

merge 시에 conflict

충돌

- merge 과정에서 파일의 이름이 같으면 충돌이 발생한다.
- 파일이 다르면 무조건 자동으로 합쳐준다.
- 파일이 같아도 수정한 부분이 다르다면 자동으로 합쳐준다.
(버전관리를 사용하는 정말 중요한 이유중의 하나)

branch merge push pull

repo