

1주차

웹 개론 / git

키워드



Client

Server

Request



Response

인터넷

TCP/IP 통신 프로토콜을 이용해 정보를 주고받는 컴퓨터 네트워크

WWW(World Wide Web)

인터넷을 이용해 접근하며 URL을 이용하여 편리하게
하이퍼텍스트 정보검색을 가능하게 하는 서비스



http / https

WWW에서 Client와 Server 사이에서 하이퍼텍스트를 주고받는 통신규칙

URL

WWW 시스템이 인지 할 수 있는 주소지정방법



← → ↻ 🏠 🔒 <https://www.naver.com>



coupang

- 상품주문을 통해
- 쿠팡에 물건을
- 배송요청 한다
- 주문내용을 받아
- 재고를 확인하고
- 물건을 배송한다.

Client



- URL을 통해
- 네이버 페이지를
- 요청한다. (Request)

Server

NAVER

- URL주소에 따라
- 해당되는 페이지를
- 응답한다.(Response)

여기까지 이해하셨다면
이제 Request 할 때의 두가지
방법를 알아봅시다!



GET

Request할 때 필요한 데이터를 URL의 끝에
이름=값 형태로 전송하는 방식

POST

Request할 때 필요한 데이터를 URL에 노출시키지 않고
request BODY에 데이터를 담아서 전송하는 방식

누구에게? == naver.com

무엇을? == /search or /blog

어떻게 == ?q=likelion or POST

이제 어떻게 구현? - 프로그래밍 언어



프레임워크



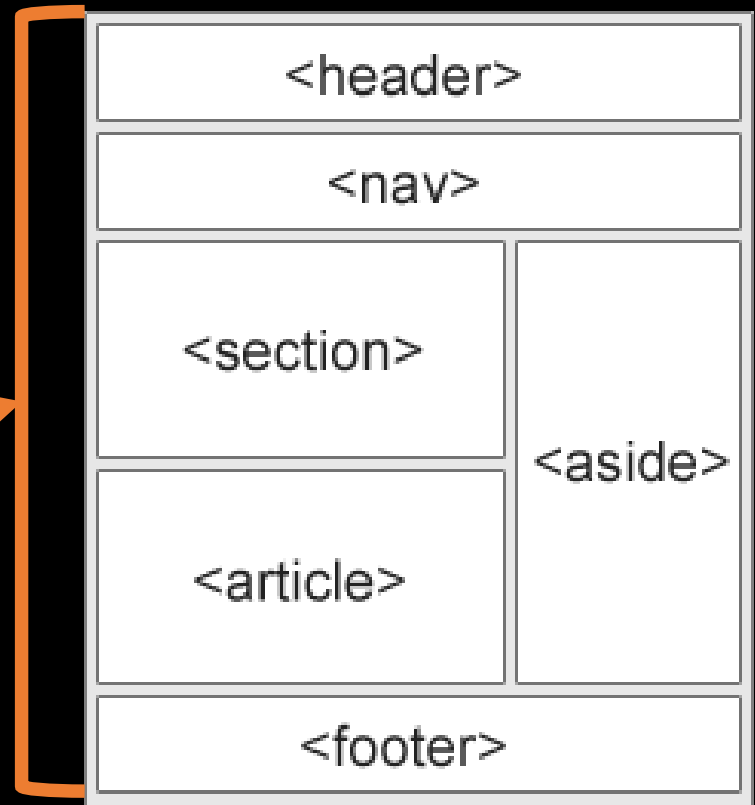
Express JS





생활코딩에서 부족한 부분을 공부해봅시다.

```
1 <html>
2   <head>
3
4   </head>
5
6   <body>
7     
8   </body>
9 </html>
```



공간분할 태그

<div>
block 분할

inline-block 분할

```
1 <html>
2   <head>
3
4   </head>
5
6   <body>
7     <div class="main">
8       <div class="1">
9         1번div
10      </div>
11      <div class="2">
12        2번div
13      </div>
14    </div>
15
16  </body>
17 </html>
```

table 태그

<th> 열 제목
<tr> 행
<td> 열

```
19 <div class="bottom">
20   <table>
21     <th>테이블</th>
22     <th>만들기</th>
23
24     <!-- 첫번째 줄 시작 -->
25     <tr>
26       <td>첫번째 칸</td>
27       <td>두번째 칸</td>
28     </tr>
29     <!-- 첫번째 줄 끝 -->
30
31     <!-- 두번째 줄 시작 -->
32     <tr>
33       <td>첫번째 칸</td>
34       <td>두번째 칸</td>
35     </tr>
36     <!-- 두번째 줄 끝 -->
37   </table>
38 </div>
39
```

Display

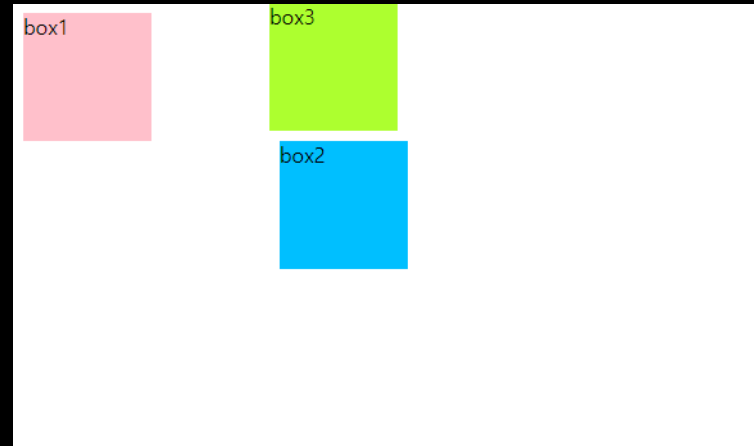
Block : 한줄 한줄 쌓아지는 형식

Inline-block : block으로 잡히지만 inline형태로 배치

None : 숨김

```
<div>  
  <div style="display: block; background: pink;">고양이</div>  
  <div style="display: block; background: deepskyblue;">댕댕이</div>  
  <div style="display: block; background: lawngreen;">펍수</div>  
  <br>  
  <div style="display: inline-block; background: pink;">고양이</div>  
  <div style="display: inline-block; background: deepskyblue;">댕댕이</div>  
  <div style="display: inline-block; background: lawngreen;">펍수</div>  
</div>
```

position



Static : 정적으로 위치, 기본값

Relative : 원래의 static 위치를 기준으로 상대적인 위치에 지정

Absolute : 조상의 position 위치를 기준으로 지정

Fixed : 스크롤의 영향을 받지 않고 고정

Git basic

<디자이너의 흔한 최종 파일>

0912_아이콘_최종.png

0912_아이콘_최종_수정.png

0913_아이콘_최종_수정_2차.png

0916_아이콘_파이널.png

0916_아이콘_진짜_파이널.png

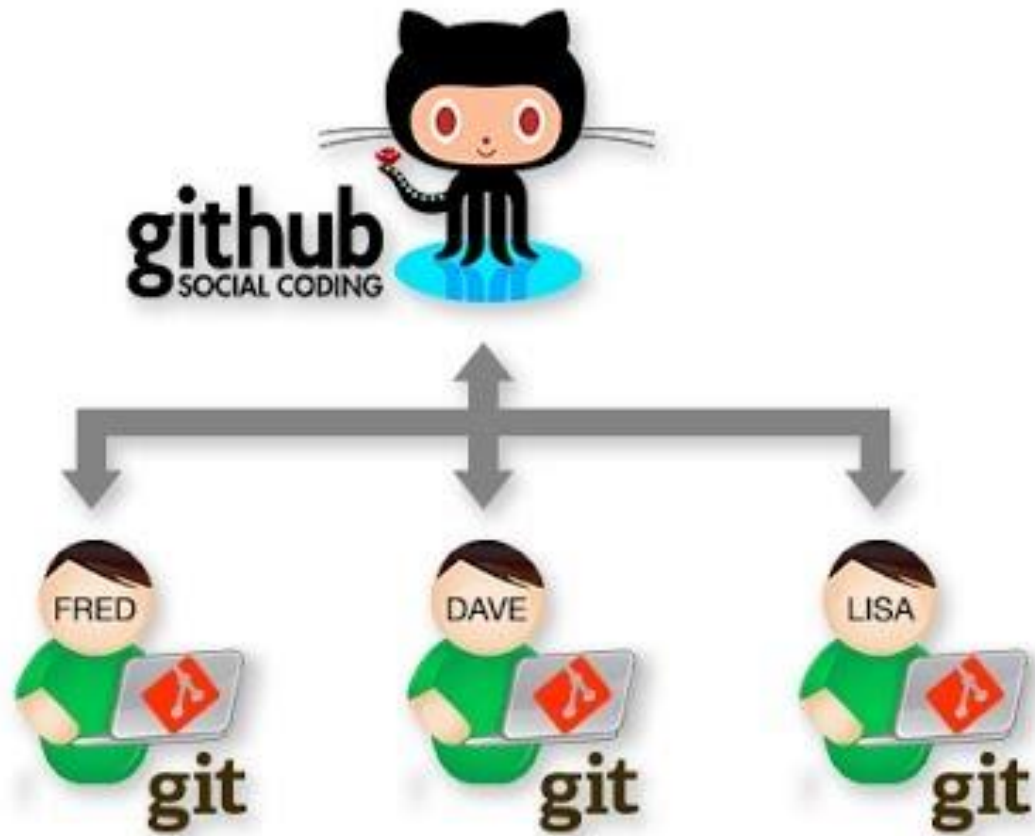
0917_아이콘_확정.png

0919_아이콘_NEW_시안_5종.jpg

버전관리시스템 =

Git with github

= git 호스팅 서비스





회원가입 먼저!!!!

<https://github.com/join>

github repo 생성

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Repository template

Start your repository with a template repository's contents.

No template ▾

Owner

Repository name *

 parksanghyup ▾ / git_practice ✓

Great repository names are short and memorable. Need inspiration? How about [ideal-tribble?](#)

Description (optional)



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ Initialize this repository with a README

This will let you immediately clone the repository to your computer.


Add .gitignore: None ▾

Add a license: None ▾



Create repository

repo를 생성했다면 아래와 같은 페이지를 볼 수 있습니다

 parksanghyup / git_practice

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/parksanghyup/git_practice.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 "# git_practice" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/parksanghyup/git_practice.git
git push -u origin master
```



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

```
git remote add origin https://github.com/parksanghyup/git_practice.git
git push -u origin master
```



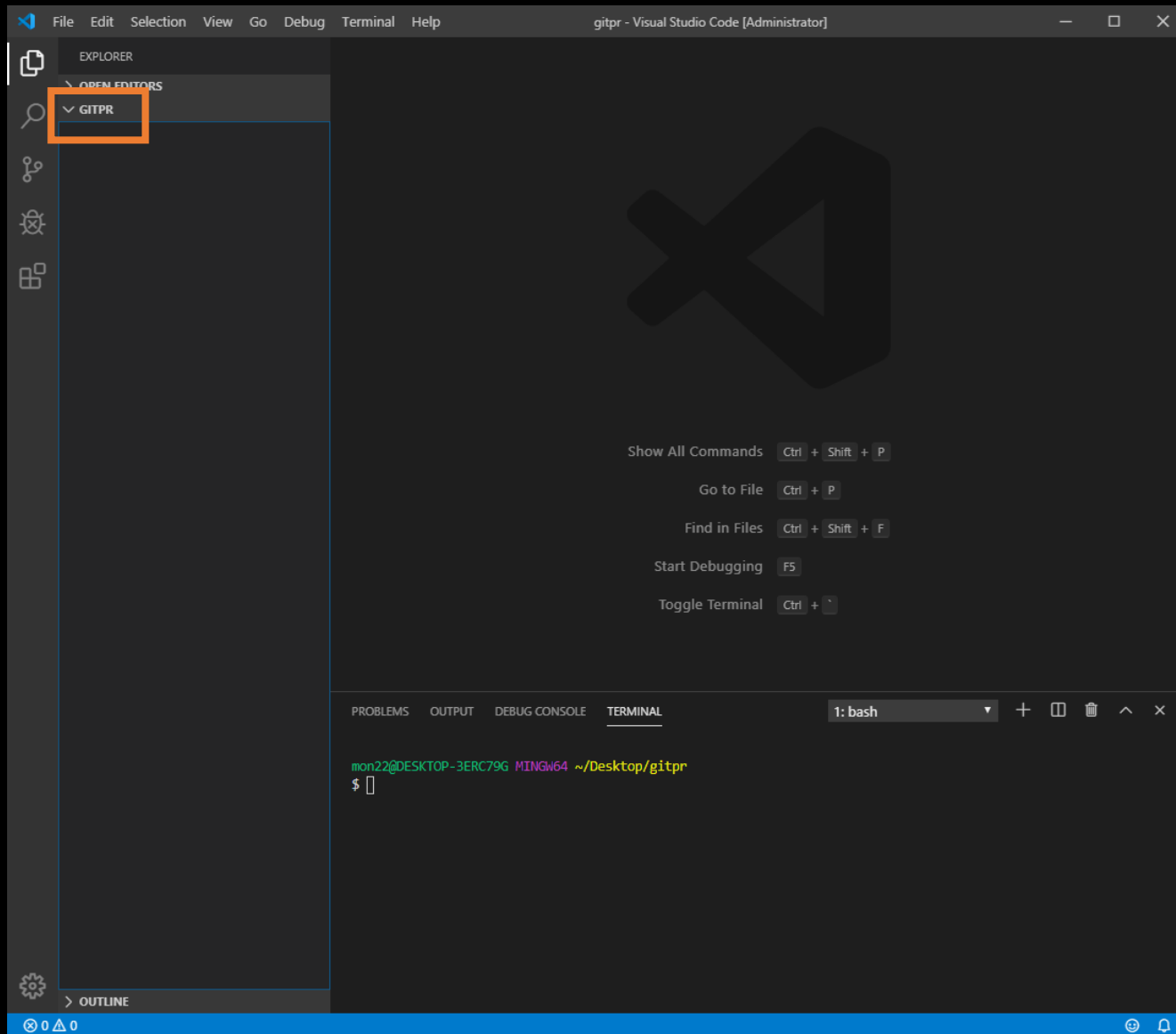
...or import code from another repository

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

Import code

바탕화면에 `gitpr`이라는 폴더를 만들어주세요

그리고 `vscode`로 폴더를 열어주세요



git init

작업 경로 내의 변화를 추적하기 위한 시작 명령
git을 사용하려면 제일 먼저 해주어야 합니다!

```
mon22@DESKTOP-3ERC79G MINGW64 ~/Desktop/gitpr  
$ git init  
Initialized empty Git repository in C:/Users/mon22/Desktop/gitpr/.git/
```

git으로 관리하려는 **폴더**에서 실행해야 합니다!
gitpr **폴더**에서 실행해 주세요!

Git config

```
git config --global user.name "Your name"
```

```
git config --global user.email "Your email address"
```

원격저장소 연결

git remote add origin <remote_addr>

```
mon22@DESKTOP-3ERC79G MINGW64 ~/Desktop/gitpr (master)  
$ git remote add origin https://github.com/hoonihu1/Vue-Study/[]
```

방금 전에 생성한 github의 주소를 입력

origin은 별칭(nickname)을 의미합니다
각자 하고 싶은대로 정해도 되지만 보통은 origin을
제일 많이 사용합니다!!

원격저장소 연결확인

git remote

```
mon22@DESKTOP-3ERC79G MINGW64 ~/Desktop/gitpr (master)
$ git remote
origin
```

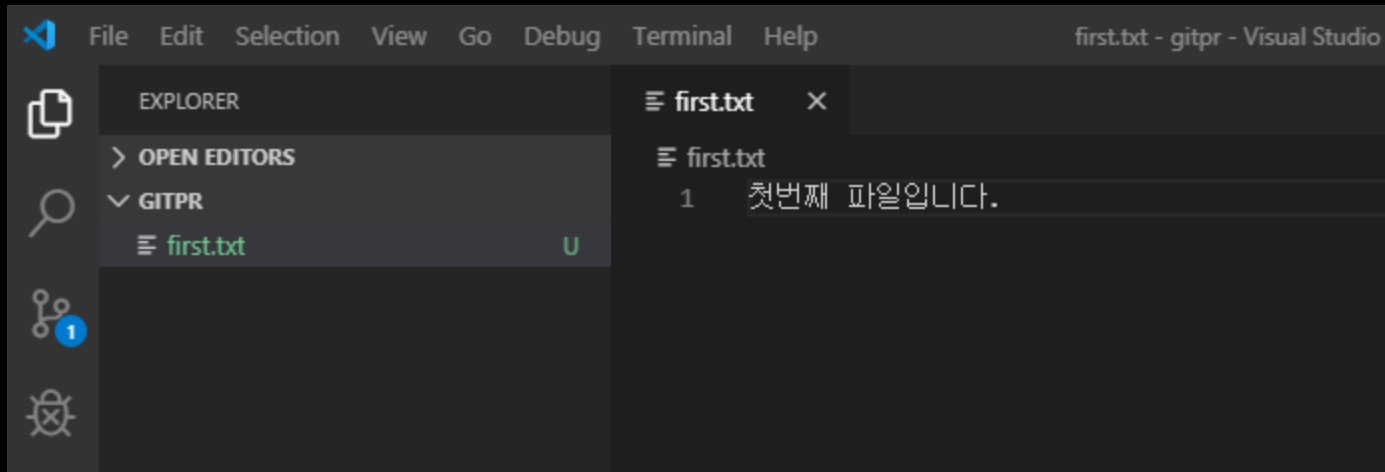
git remote -v

```
mon22@DESKTOP-3ERC79G MINGW64 ~/Desktop/gitpr (master)
$ git remote -v
origin https://github.com/hoonihu1/Vue-StudY (fetch)
origin https://github.com/hoonihu1/Vue-StudY (push)
```

- git status
- git add
- git commit
- git push
- git pull
- git clone
- git diff
- git log

- git branch
- git checkout

파일 하나를 생성해 봅시다



git status

```
mon22@DESKTOP-3ERC79G MINGW64 ~/Desktop/gitpr (master)
```

```
$ git status
```

```
On branch master
```

```
No commits yet
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
first.txt
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

아직 git이 추적하지 않고 있는 파일임을 의미합니다(untracked)

git add <file_name>

git의 파일 추적시스템에 추가해 달라고 명령합니다

```
mon22@DESKTOP-3ERC79G MINGW64 ~/Desktop/gitpr (master)
$ git add first.txt
```

```
mon22@DESKTOP-3ERC79G MINGW64 ~/Desktop/gitpr (master)
$ git status
On branch master
```

```
No commits yet
```

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

git add 로 파일을 추적되게 한 후
git status 명령어로 다시 확인 해 봅니다
(changes to be committed - 커밋될 변경점)

Git add를 통해 등록된 파일만
commit 할 수 있습니다.

git status

자주 사용해주세요!

어떤 파일을 git에 등록하고 커밋할지 자주 확인하는 습관을 가집시다!

git commit

변경점들을 기록에 남기는 과정

commit을 하는 방법

1. git commit 입력 후 vi 에디터로 메시지 작성
2. git commit -m "msg" 로 메시지 작성

git commit -m "<msg>"

```
mon22@DESKTOP-3ERC79G MINGW64 ~/Desktop/gitpr (master)
$ git commit -m '첫번째 커밋'
[master (root-commit) c1a62e9] 첫번째 커밋
1 file changed, 1 insertion(+)
create mode 100644 first.txt
```

#좋은 커밋 메시지

<http://meetup.toast.com/posts/106>

Edit how to return keyboard ...

 jinmay committed on 30 Oct 2017

Add new function to return keyboard in every POST request

git log

내가 작성한 커밋 로그를 볼 수 있어요

```
mon22@DESKTOP-3ERC79G MINGW64 ~/Desktop/gitpr (master)
$ git log
commit c1a62e9e12cf8c193beec4a8e10c0f491edf974d (HEAD -> master)
Author: parksanghyeop <parksanghyup>
Date:   Mon Nov 11 13:26:44 2019 +0900
```

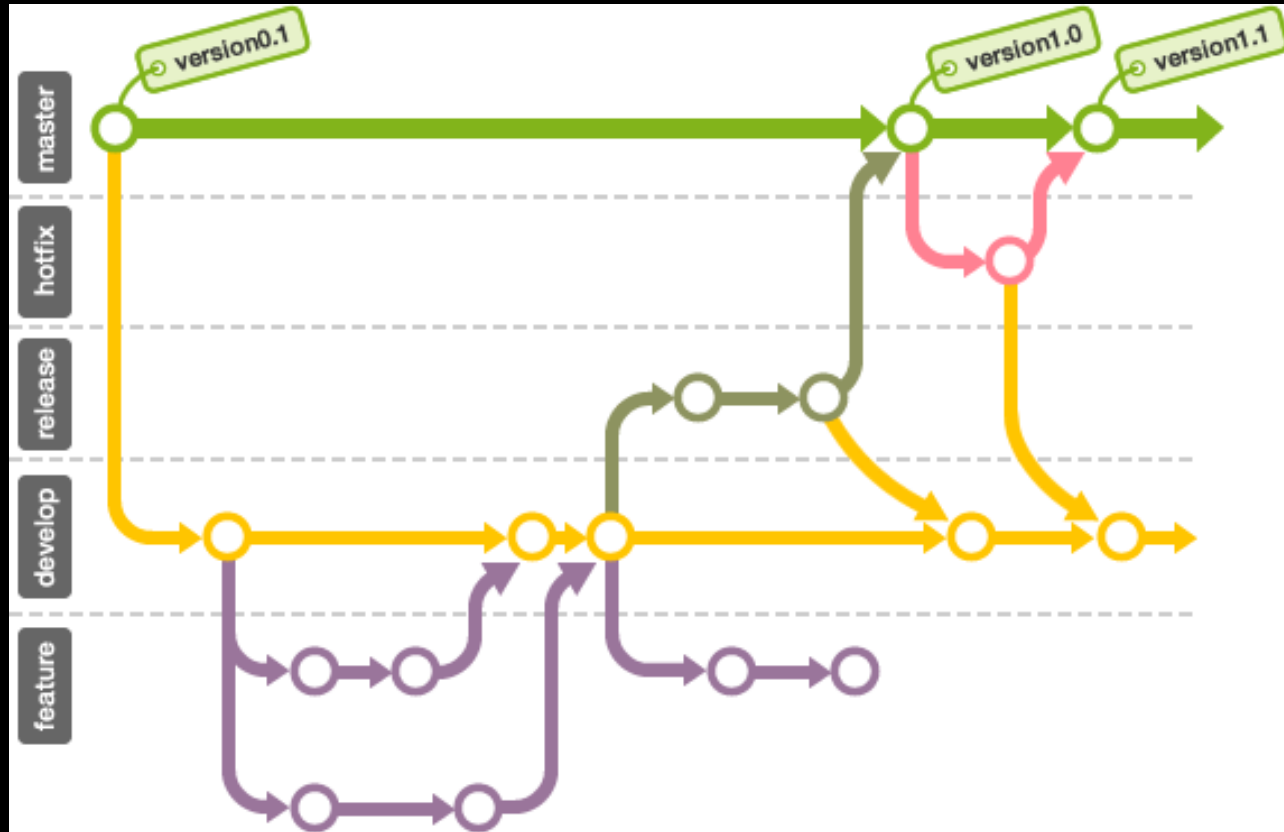
첫번째 커밋

git push

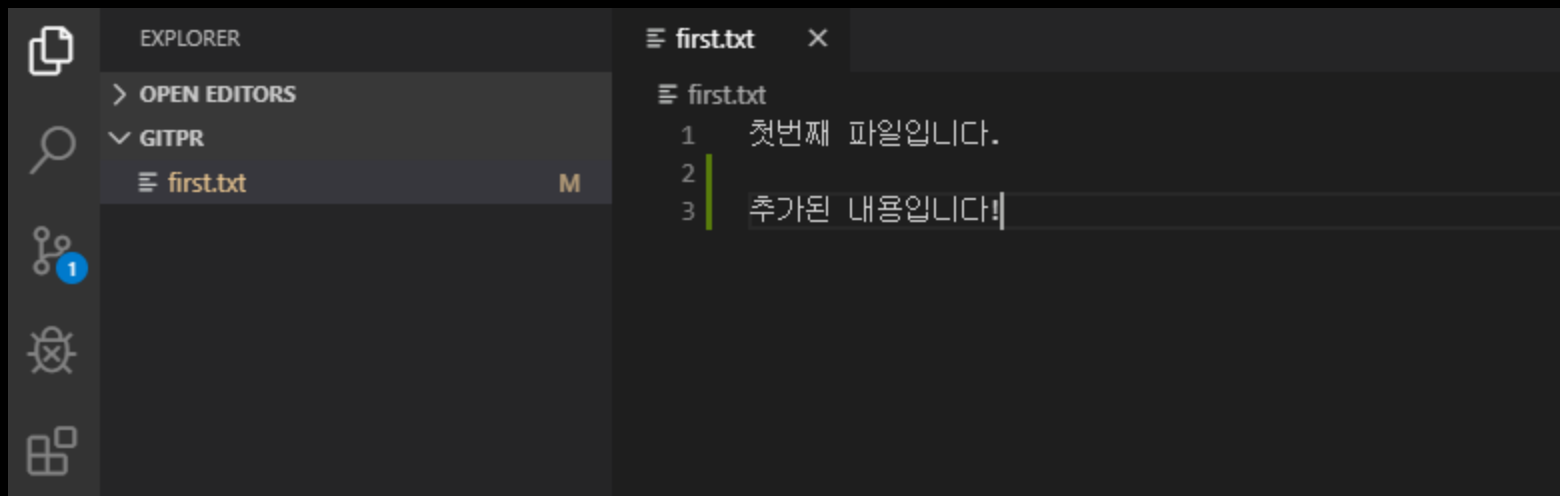
커밋들을 연결된 **github repo**에 올리는 과정

```
mon22@DESKTOP-3ERC79G MINGW64 ~/Desktop/gitpr (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 249 bytes | 249.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/parksanghyup/git_practice
* [new branch]      master -> master
```

아까 설명한 것 처럼
origin은 **github repo**에 대한 별칭입니다



변경점을 만들어봅시다



git diff

```
mon22@DESKTOP-3ERC79G MINGW64 ~/Desktop/gitpr (master)
$ git diff
diff --git a/first.txt b/first.txt
index fd7863f..f4fb532 100644
--- a/first.txt
+++ b/first.txt
@@ -1,3 @@
-첫번째 파일입니다.
\ No newline at end of file
+첫번째 파일입니다.
+
+추가된 내용입니다!
\ No newline at end of file
```

이제는 github repo에서 받아올게요

git clone

Repo 복사

git pull

Commit 갱신

Git clone <https://github.com/parksanghyup/likelion8/>

```
mon22@DESKTOP-3ERC79G MINGW64 ~/Desktop/gitpr (master)
$ git clone https://github.com/parksanghyup/likelion8/
Cloning into 'likelion8'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 9 (delta 0), reused 6 (delta 0), pack-reused 0
Unpacking objects: 100% (9/9), done.
```

git pull origin master

```
mon22@DESKTOP-3ERC79G MINGW64 ~/Desktop/gitpr (master)
$ git pull origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/parksanghyup/likelion8
* branch          master      -> FETCH_HEAD
* [new branch]     master      -> origin/master
Updating 13f3110..9ea4070
Fast-forward
 ...4\353\251\213\354\237\201\354\235\264\354\202\254\354\236\220.txt" | 4 +++-
1 file changed, 3 insertions(+), 1 deletion(-)
```

```
mon22@DESKTOP-3ERC79G MINGW64 ~/Desktop/gitpr (master)
$ git pull origin master
From https://github.com/parksanghyup/likelion8
* branch          master      -> FETCH_HEAD
Already up to date.
```

과 제

공지방에 웹페이지 사진을 올려 드리겠습니다

똑같이 만든 후에 Github repo를 새로 만들고

Push 해주시고 Github 주소를 보내주세요