

Week 5

협업 & git

목차

1. 협업에 대하여
2. Git?
3. Git 사용해보기
4. 왜 에러가 날까?

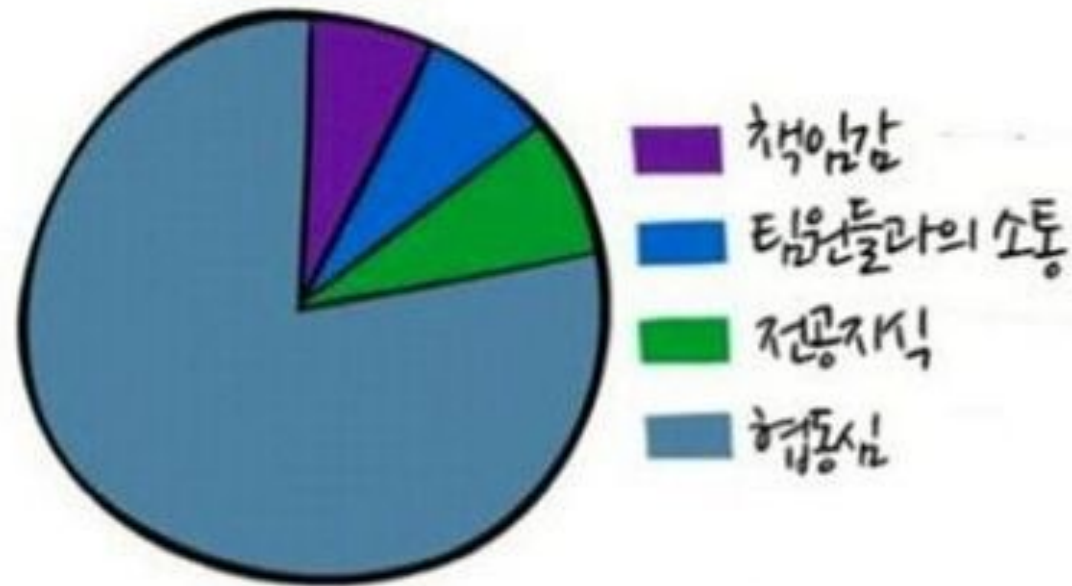
1. 협업에 대하여

a to c

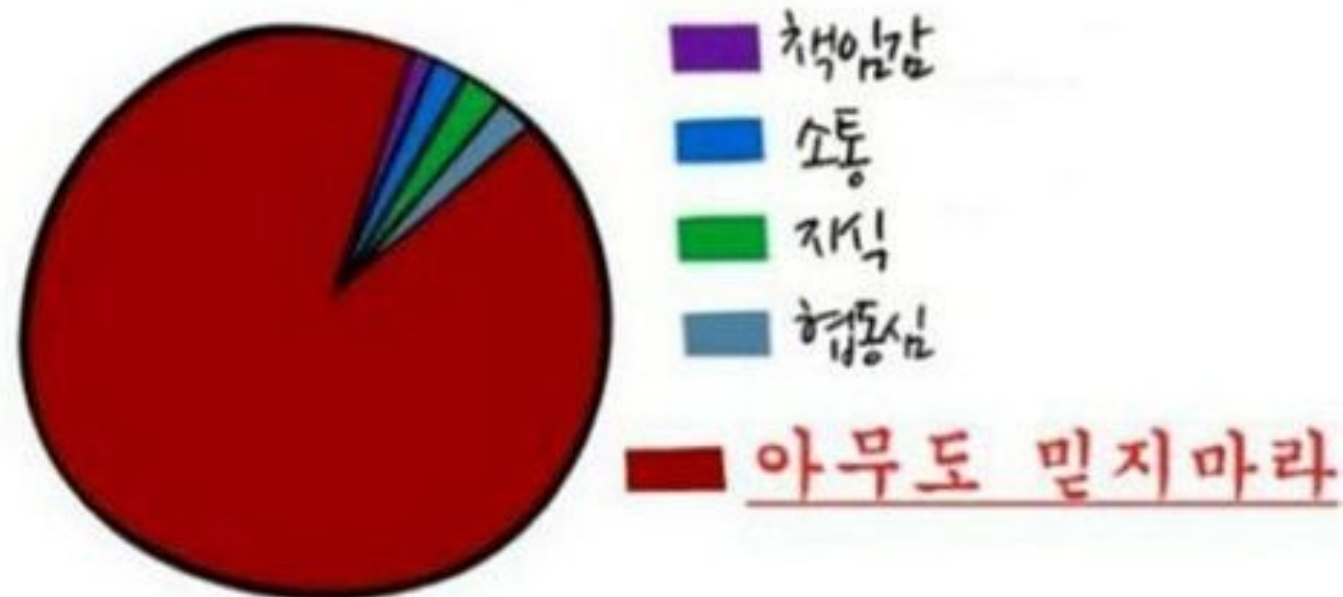
협업이란?

모두 일하는, 협력하는 것
공동작업을 가리키는 말
= collaboration

교수님이 생각하셨던 팀워크의 균형



우리가 실제로 알게 되는 것



기획자

서비스
아이템
선정

서비스
프로토타입
제작

서비스명 /
개발환경
결정

Admin
설계

개발자

Admin
개발

디자이너

Admin
검수

콘텐츠
등록



UI
개발

배포



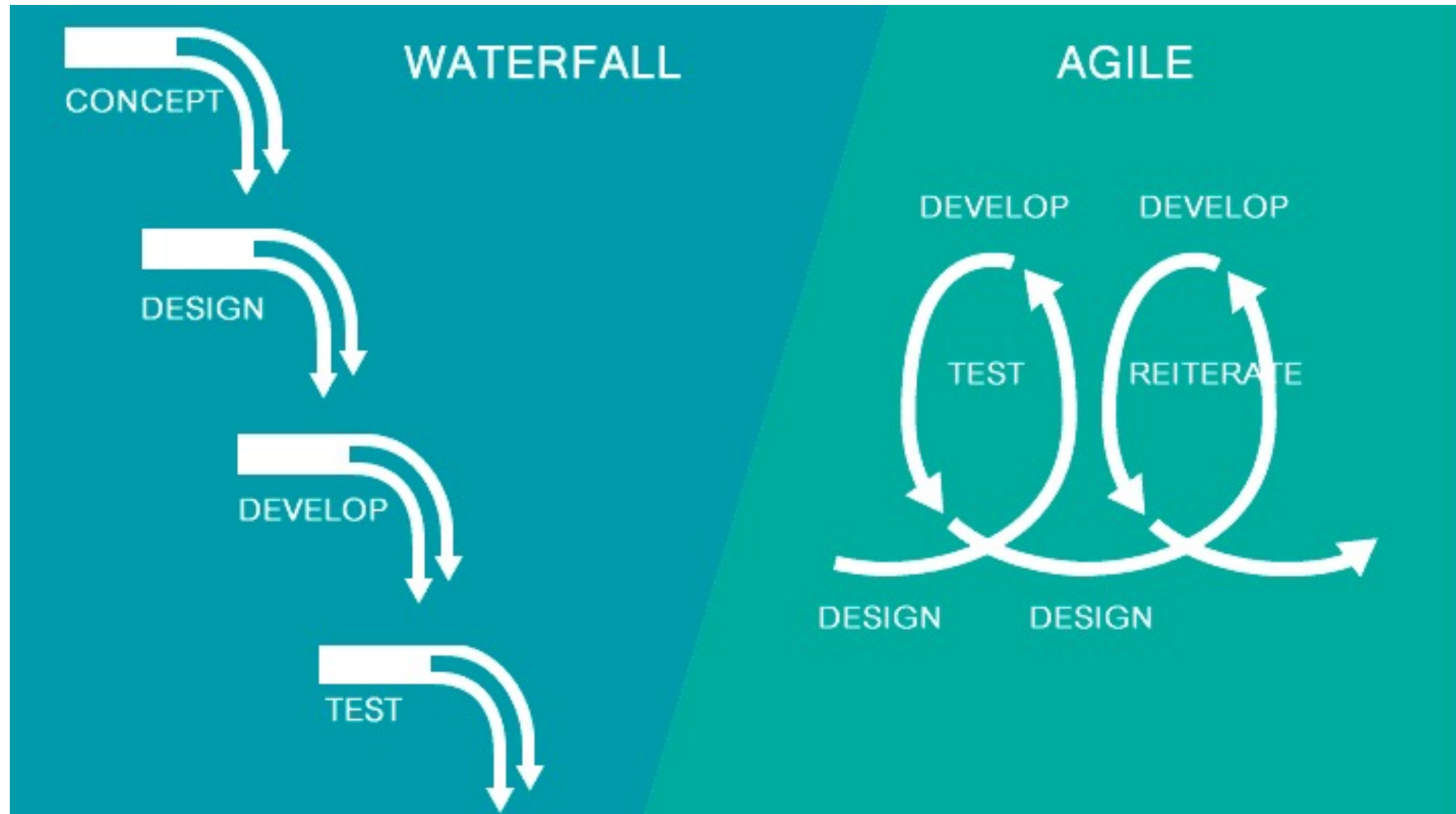
UI 디자인

UI 스타일
가이드

로고 디자인



그래서 어떻게 협업하는 건데?



멋쟁이들은
어떻게 했을까?
- PrettyLion -

서비스 아이템 선정
서비스 프로토타입 제작
서비스명 / 개발환경 결정
Admin 설계
Admin 개발
Admin 검수
콘텐츠 등록
UI 디자인
UI 스타일 가이드
로고 디자인
UI 개발

협업이란?

공동의 목표를 이루기 위해
커뮤니케이션하면서 작업하는 것

2. Git?

Git

파일의 변경사항을 추적하고
여러 명의 사용자들 간에
해당 파일들의 작업을 조율하기 위한
분산 버전 관리 시스템

Git

~~파일의 변경사항을 추적하고
여러 명의 사용자들 간에
해당 파일들의 작업을 조율하기 위한
분산 버전 관리 시스템~~

GitHub

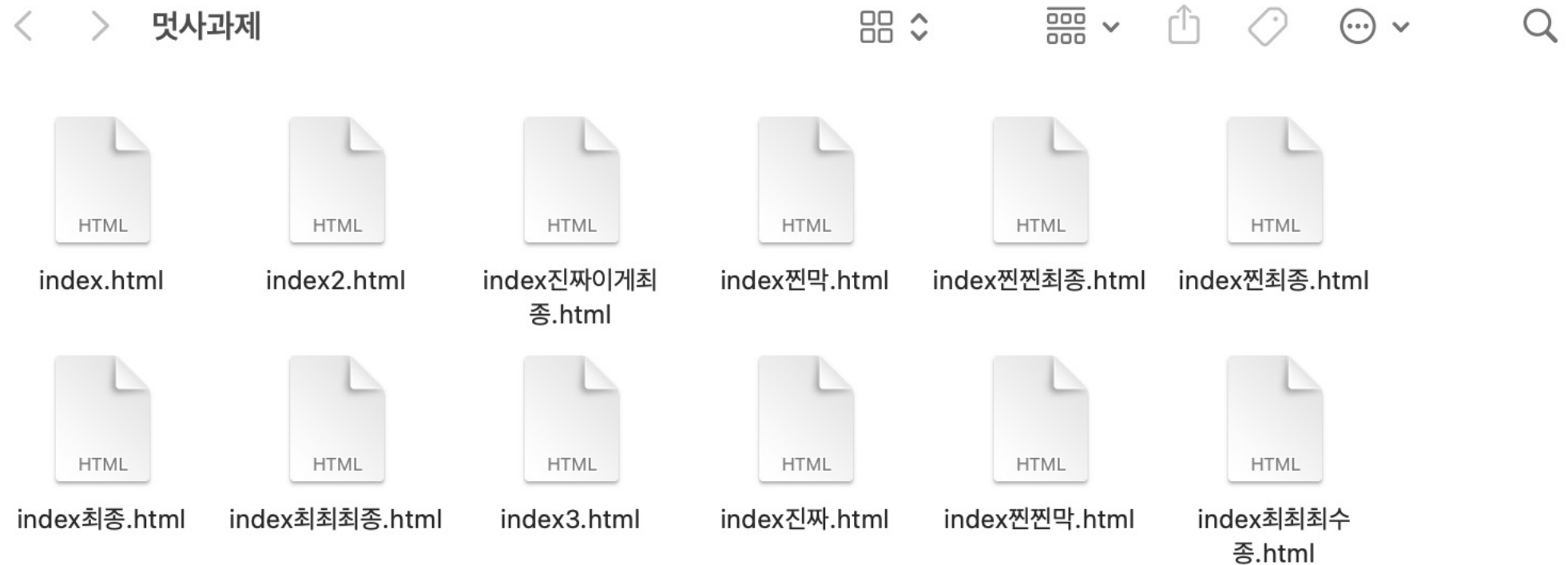
분산 버전 관리 툴인 깃 저장소 호스팅을
지원하는 웹 서비스

Git Bash

GNU 프로젝트를 위해
브라이언폭스가 작성한 유닉스 셸



본격적으로 시작하기 전에! 비유를 통해 감을 잡아볼까요?



vscode를 열어서 새 파일을 생성합니다.

내용을 추가하고 저장을 합니다.

이름을 붙여줍니다.

멋사과제 폴더를 엽니다.

폴더에 파일을 업로드합니다.

3. Git 뽐셔보자!

먼저 몇가지 명령어를 배워보아요~

```
git config --global user.name "[firstname lastname]"
```

```
git config --global user.email "[valid-email]"
```

```
git config --global init.defaultBranch main
```

```
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
```

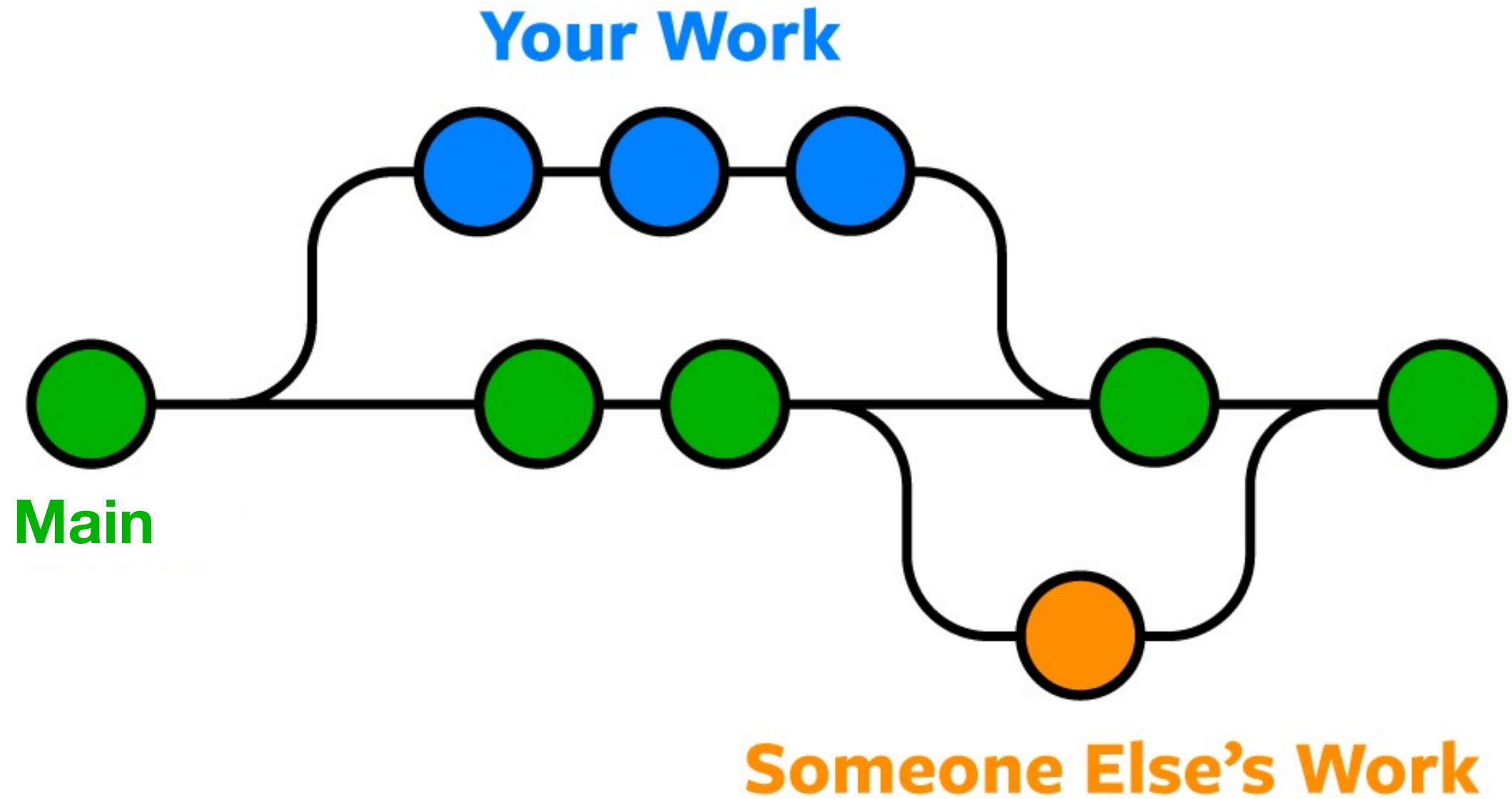
```
git config --list
```

먼저 몇가지 명령어를 배워보아요~

<code>git init</code>	: <code>vscode</code> 를 열어서 새 파일을 생성합니다.
<code>git add [file]</code>	: 내용을 추가하고 <code>저장</code> 을 합니다.
<code>git commit -m “ 커밋 내용 ”</code>	: <code>이름</code> 을 붙여줍니다.
<code>git remote add [alias][url]</code>	: 멋사과제 폴더를 <code>연</code> 니다.
<code>git push [alias][branch]</code>	: 폴더에 파일을 <code>업로드</code> 합니다.
<code>git pull [alias][branch]</code>	

실습

엥? 이게 협업...?
아직 깃에 대해 다 알았다고 하면 오산!
BRANCH가 뭔지는 알아줘야~



먼저 몇가지 명령어를 배워보아요~

```
git clone [url]
```

```
git branch
```

```
git branch [branch]
```

```
git checkout [branch]
```

```
git checkout -b [branch]
```

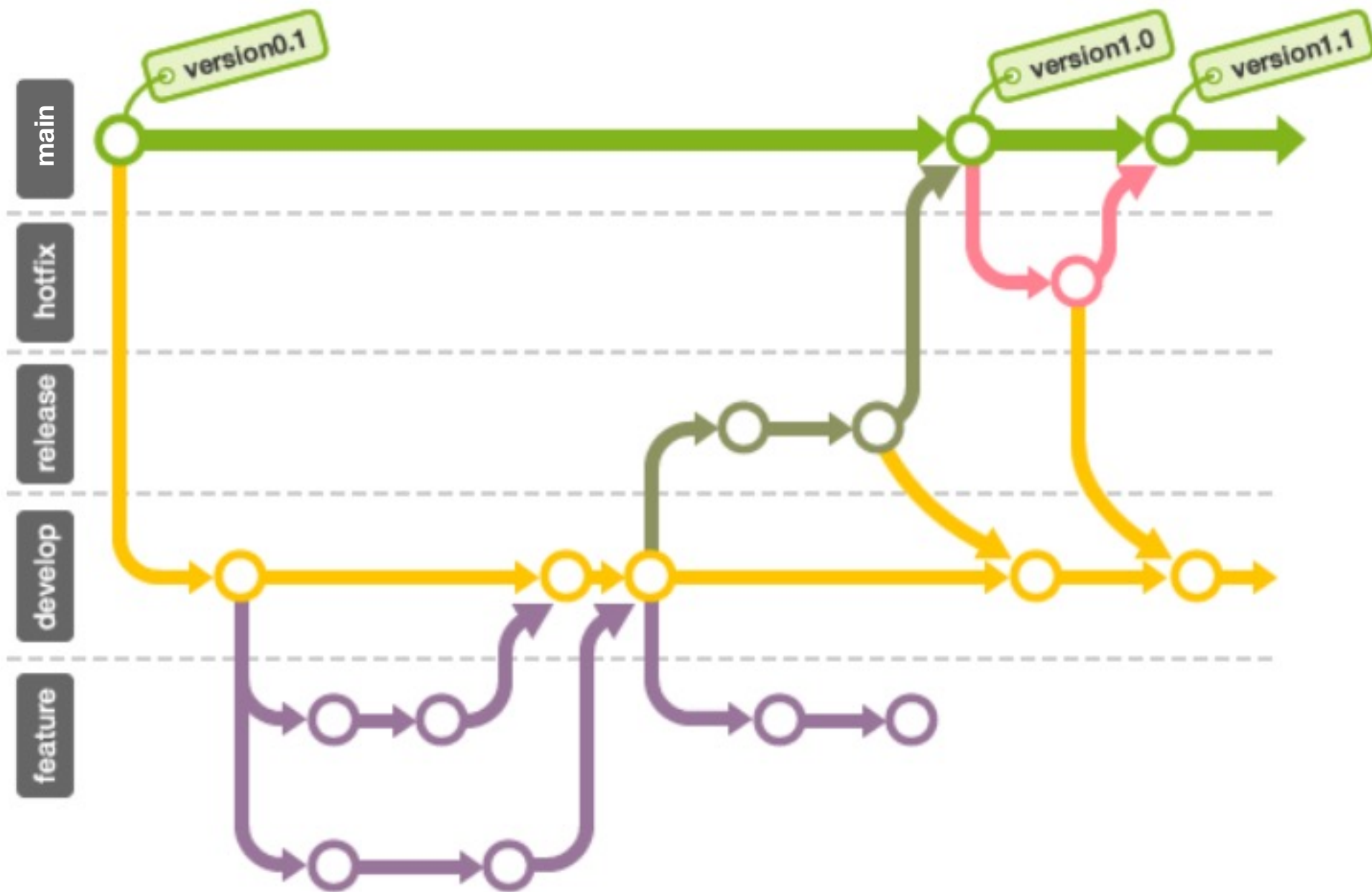
```
git branch -d [branch]
```

```
git branch -m [old branch][new branch]
```


실습

Branch 네이밍

- (1) main branch
- (2) develop branch
- (3) feature branch
- (4) release branch
- (5) hotfix branch



`git commit -m “커밋 내용”`

Commit 메시지 컨벤션

Feat	새로운 기능을 추가할 경우
Fix	버그를 고친 경우
Design	CSS 등 사용자 UI 디자인 변경
!BREAKING CHANGE	커다란 API 변경의 경우
!HOTFIX	급하게 치명적인 버그를 고쳐야하는 경우
Style	코드 포맷 변경, 세미 콜론 누락, 코드 수정이 없는 경우
Refactor	프로덕션 코드 리팩토링
Comment	필요한 주석 추가 및 변경
Docs	문서를 수정한 경우
Test	테스트 추가, 테스트 리팩토링(프로덕션 코드 변경 X)
Chore	빌드 테스트 업데이트, 패키지 매니저를 설정하는 경우(프로덕션 코드 변경 X)
Rename	파일 혹은 폴더명을 수정하거나 옮기는 작업만인 경우
Remove	파일을 삭제하는 작업만 수행한 경우

4. 왜 에러가 날까?

- 동일한 파일명 사용
- 동일한 파일 안에서 여러명이 커밋 푸시
 - pull 안 받고 진행

이제 파일 drag&drop 멈춰~! 🖐️

과제

- (1) 본인 레포 clone 받기
- (2) branch 파기(이름은 자유)
- (3) week5.html 만들기
- (4) 자유롭게 코딩
- (5) add, commit, push, PR 해보기
(머지는 운영진이 할 예정입니다~!)

공지

<중소커톤>

- 5/11 웹페이지 클론코딩
- 영권 + 선영 : 6명 1팀, 나머지 조는 4명 1팀
- 2인 1조 Pair Coding → 2인 1page. 총 2page(영선 팀은 3page)
- Pair Coding
→ 애자일 개발 방법론 중의 하나로 하나의 개발 가능한 PC 에서 두 명의 개발자가 함께 작업하는 것을 말합니다. 네비게이터(navigator)가 전략을 제시하고 드라이버(driver)가 실제 코드를 작성하며, 이 역할을 각자 번갈아가며 수행합니다.
- 구현해야할 페이지는 당일 공개
- github는 반드시 활용. 팀 별로 레포지토리 생성.
- README.md 작성
- git 명령어를 활용하여, branch 나누어서 진행