

# 1주차: Git과 Github, HTML 기초

계명대학교 멋사 13기 프론트엔드 2팀

# Git과 Github

## 1. Git, Github란?

- Git이란?
- Git을 쓰는 이유
- Github란?

## 2. Git 주요 기능

- init과 clone, 그리고 add와 status
- commit과 push, 그리고 pull
- branch와 remote 저장소
- fork 그리고 pull request

## 3. Q & A

# 1. Git, Github란?

# Git이란?

Git이란 컴퓨터 파일의 변경 사항을 **추적**하고 파일들의 작업을 조율하는 분산 버전 관리 시스템(VCS)입니다.  
즉, 소프트웨어 개발에서 코드를 **관리**하고 **기록**하고 버전 관리를 해주므로 체계적인 개발이 가능하도록 도와주는  
무료 공개 **소프트웨어**입니다.

상황 1.

## 버전 관리가 너무 힘들어요..

내 코드 혹은 다른 개발자의 코드가 변경된 이력을 쉽게 확인할 수 있고, 특정 시점에 저장된 버전과 비교, 특정 시점으로 돌아갈 수도 있습니다.



상황 2.

저 코드, 믿어도 되는거죠?

내 코드와 다른 사람의 코드를 합치는 게 쉽고 내 코드와 다른 사람의 코드가 충돌 시 경고 메시지를 통해 어떤 부분에서 충돌이 났는지 알려줍니다.

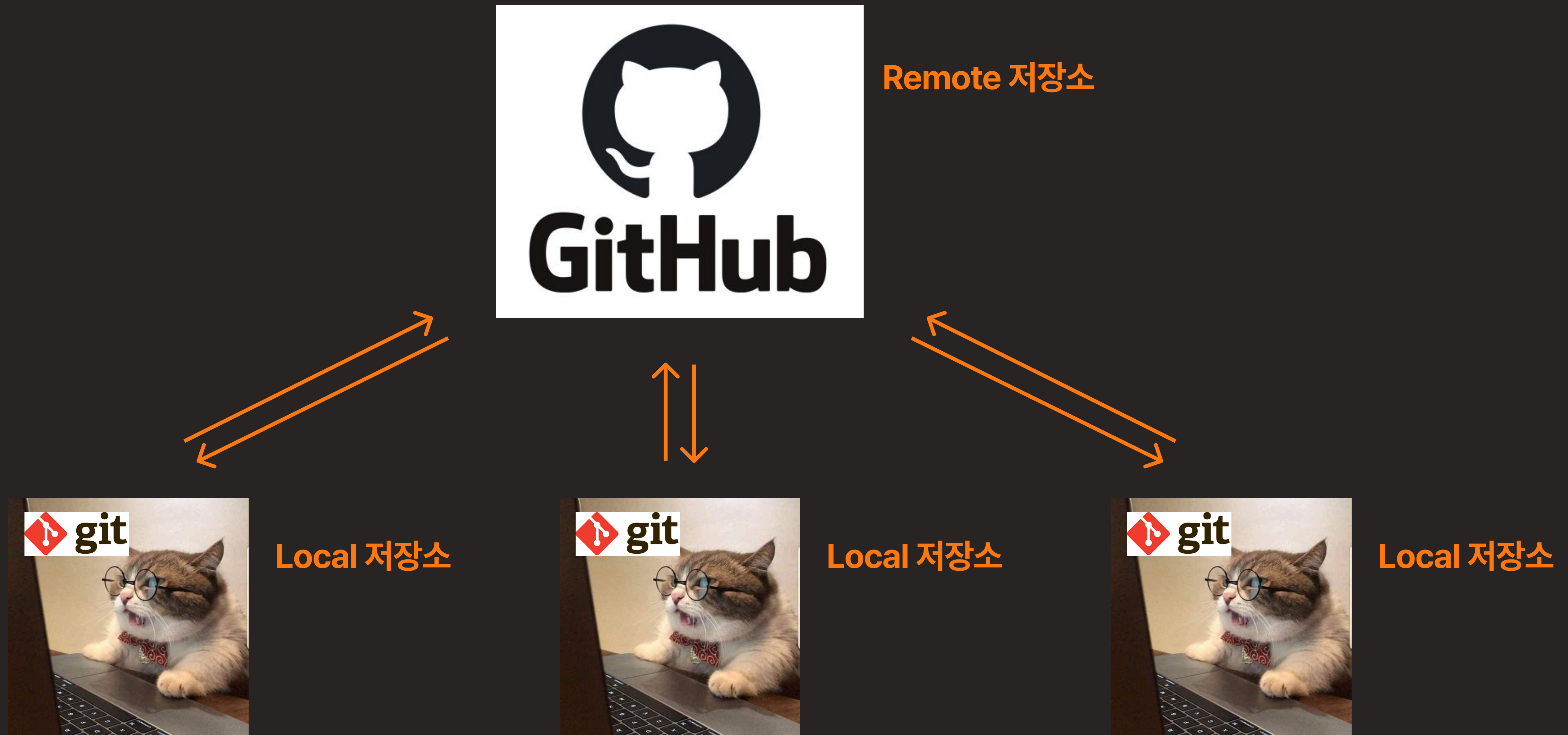


아이 뭐 그런거는 믿음으로 가는거지

# Github 란?

Github는 **Git을 사용하는 프로젝트**를 지원하는 **웹호스팅 플랫폼**으로, 언제, 어디서나 협업 프로젝트를 쉽게 진행할 수 있도록 돕는 **역할**을 합니다!

# Github 란?





# Git과 Github의 차이점



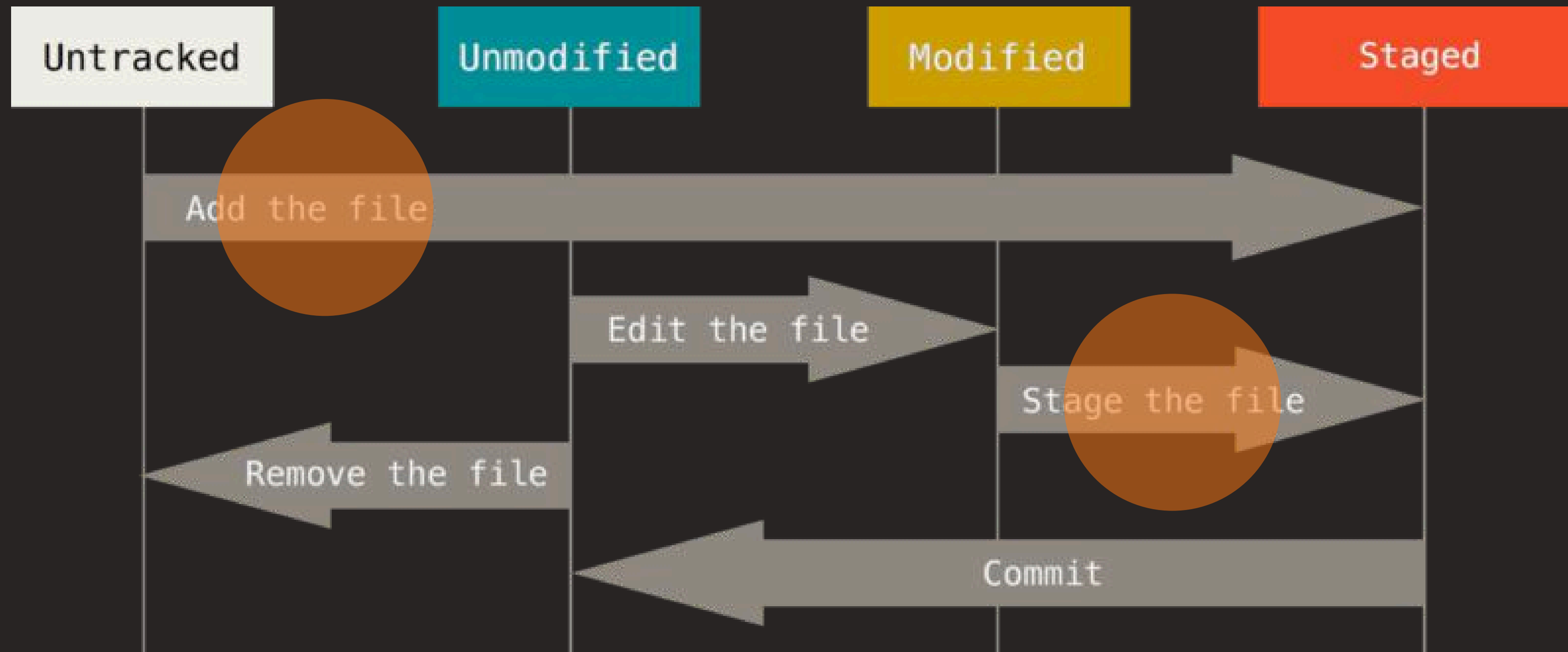
**Git: 소프트웨어**



**Github: 웹 서비스**

## 2. Git의 주요 기능

# git add와 status



# git add와 status

- git add

작업 디렉토리 상의 변경 내용을 스테이징 영역에 추가하는 명령어



- git status

작업 디렉토리와 파일의 스테이지 영역의 상태를 나타내는 명령어

```
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md
        new file:   untracked2.txt

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

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        untracked.txt
```



# commit과 push, 그리고 pull

- git commit

스테이지 된 파일들을 로컬 레포지토리에 반영하는 명령어



- git push

로컬 레포지토리의 변경 사항을 Remote 저장소에 반영하는 명령어

POV : When developers celebrate Diwali in office 🤔🔥



# git init과 clone

- git init

Git 레포지토리를 직접 생성하는 명령어



- git clone

Remote 저장소에서 Git 레포지토리를 로컬로 복제하는 명령어





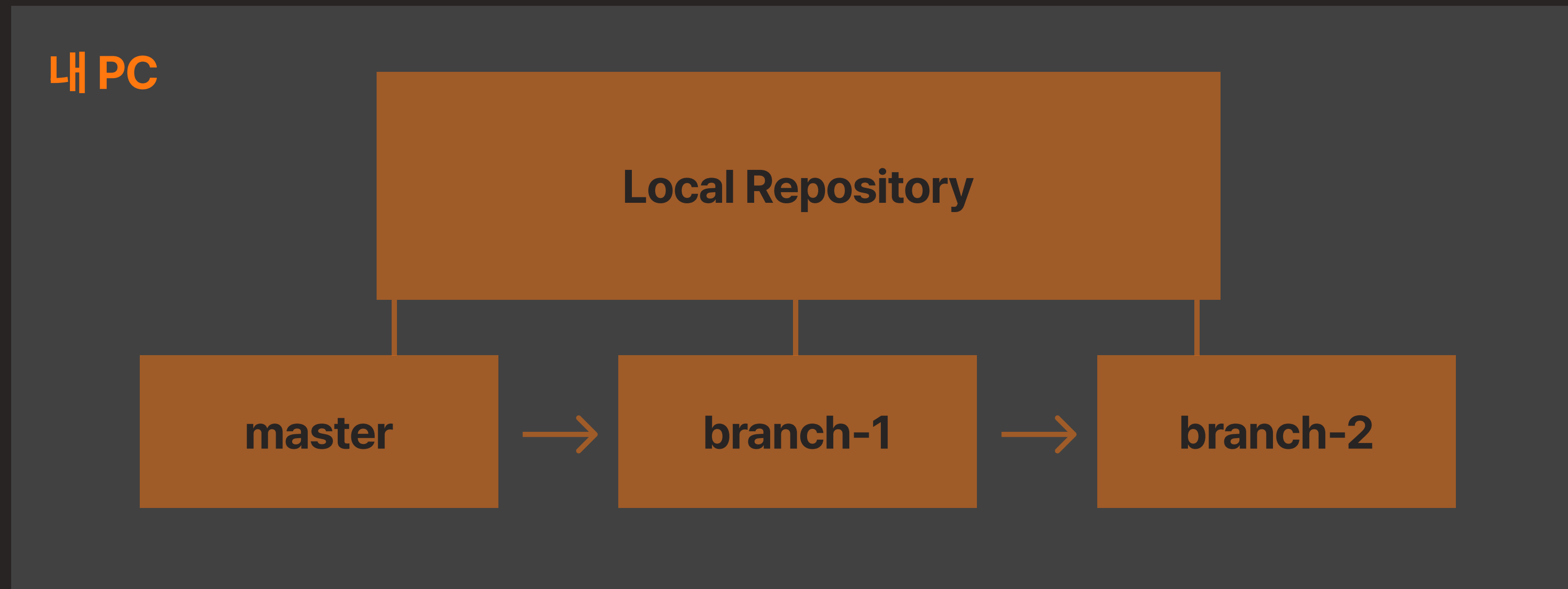
# commit과 push, 그리고 pull

- git pull

Remote 저장소에 있는 모든 내용을 가져와 로컬의 작업 내용과 **합치**는 명령어



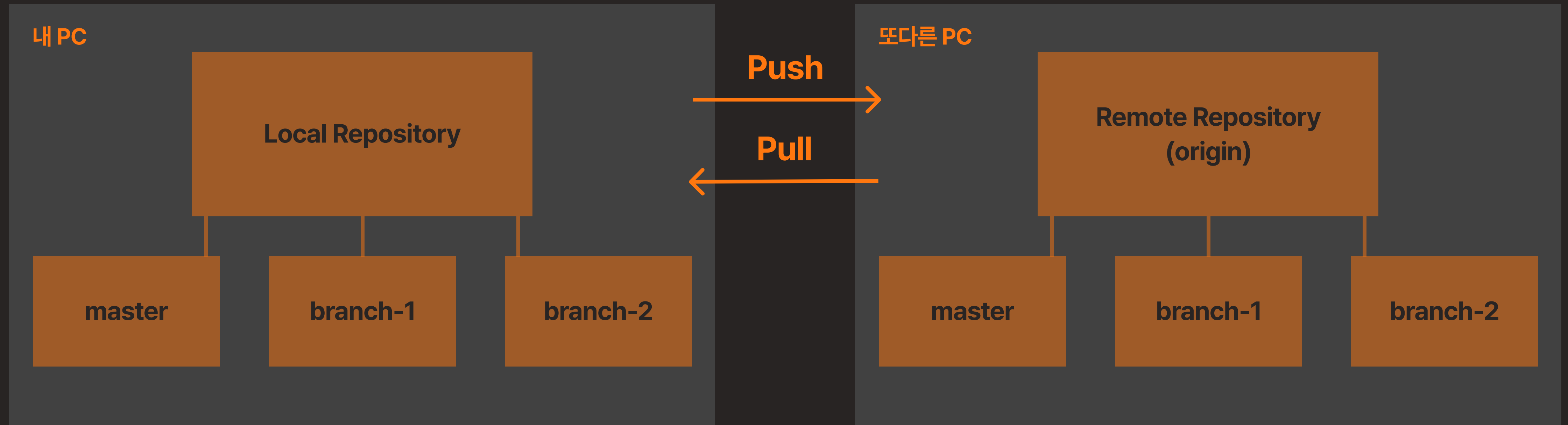
# branch와 remote 저장소



**branch = 버전, 작업 흐름**



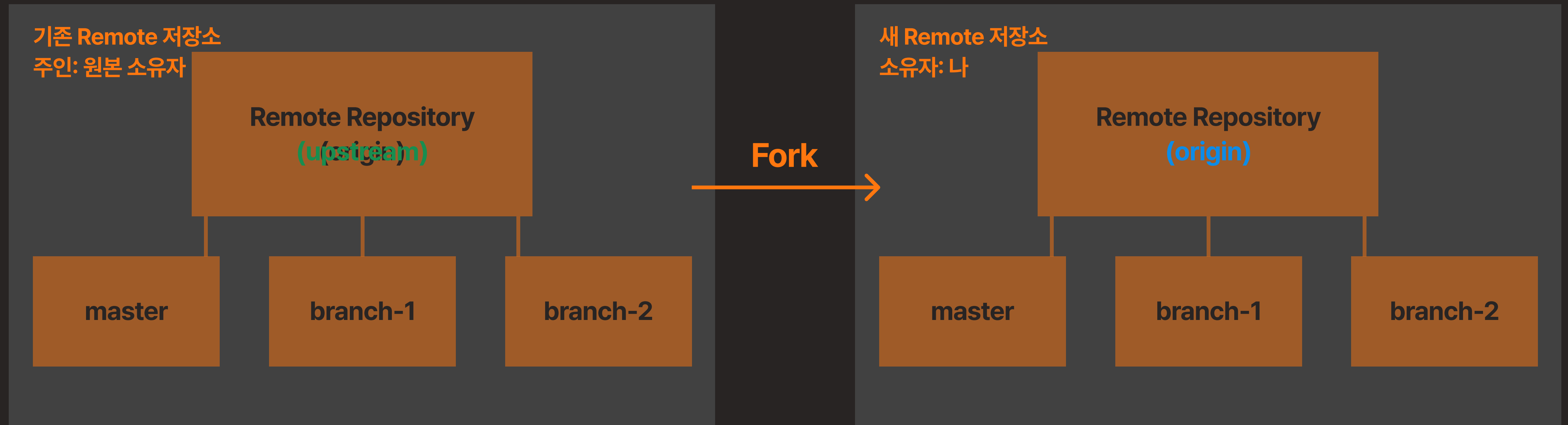
# branch와 remote 저장소



remote 저장소 = 서버

# fork 그리고 pull request

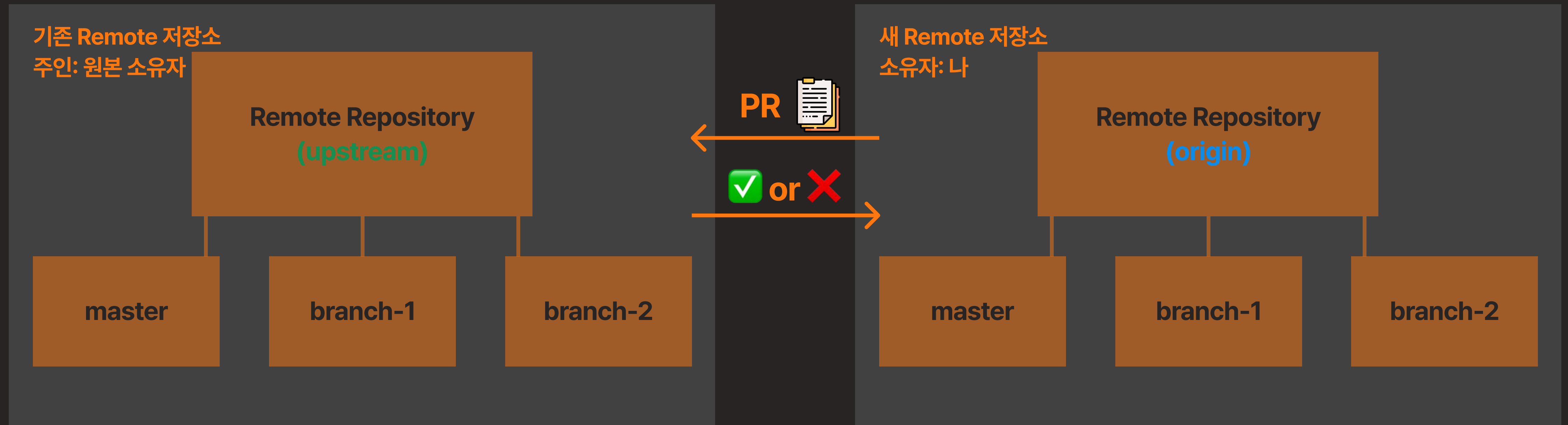
※ Github에서 제공하는 기능



fork = clone의 Github버전

# fork 그리고 pull request

※ Github에서 제공하는 기능



pull request =



### 3. Q & A

# HTML 기초

## 1. HTML이란?

- HTML이란?
- 왜 배울까?
- CSS란?

## 2. HTML 요소는?

- HTML 디렉토리
- 글자 태그
- list
- link
- 그 외 element

## 3. Form과 Field

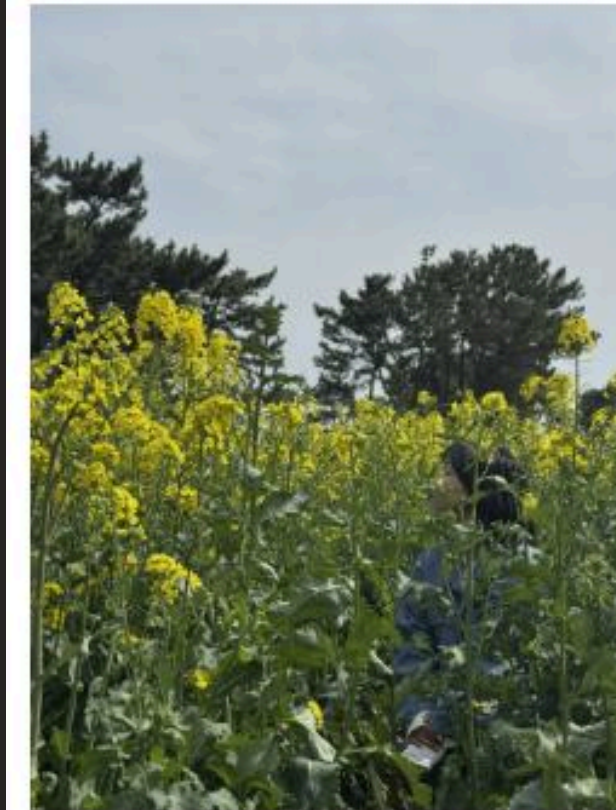
- Form이란?
- Field란?

# 오늘 최종 실습

## About Me

**KimJinA**

hello! my name is Kim JinA.



## interesting

- Listening to music
- Watching movies
- Writing a blog

[It's a song video that I've been listening to recently.](#)

## Role model

1. Lee Jae-hyun, Baseball player
2. Koo Ja-wook, Baseball player

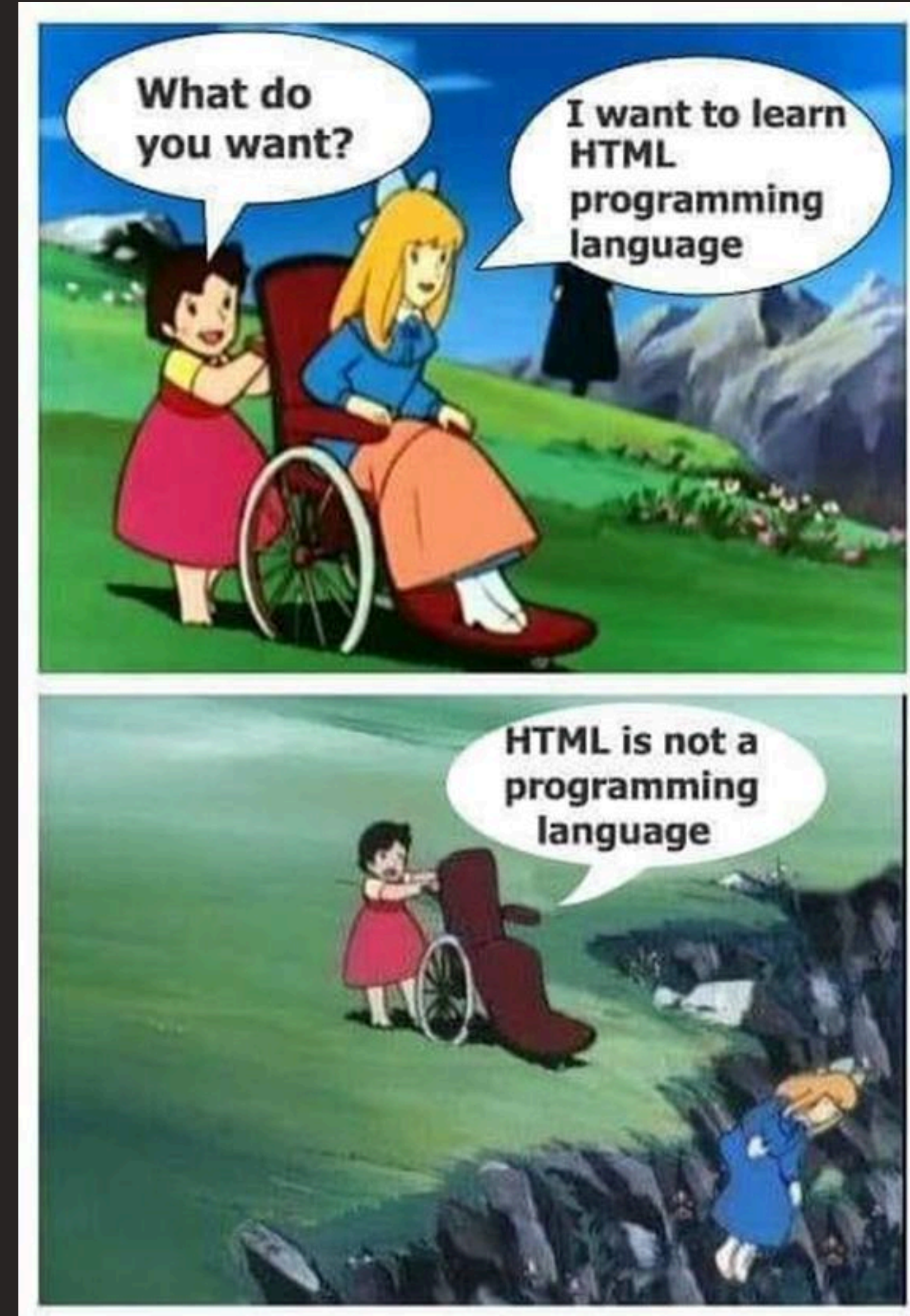
# 1. HTML이란?



# HTML 이란?

HyperText Markup Language (HTML)

제목, 단락 및 목록과 같은 문서의 다양한 구성요소를 식별하고 설명하는 마크업 언어입니다.  
현재는 HTML5까지 나와있습니다.





# 내용에 의미와 구조를 추가하기 위한 목적

내용이 어떻게 표시되어야 하는지에 대한 지침을 제공하기 위한 것이 아닙니다.  
HTML은 웹 페이지의 구조를 짜고 그 구조에 의미를 부여하기 위해 사용합니다

그러므로, 콘텐츠(내용)를 마크업 할 때는 가장 적합한 내용 및 요소에 대한 설명을 제공해야 합니다.  
요소들 간의 관계를 파악하거나 상호작용을 위해 의미와 구조를 추가하기 위해 사용합니다.

# CSS란?

Cascading Style Sheets (CSS)

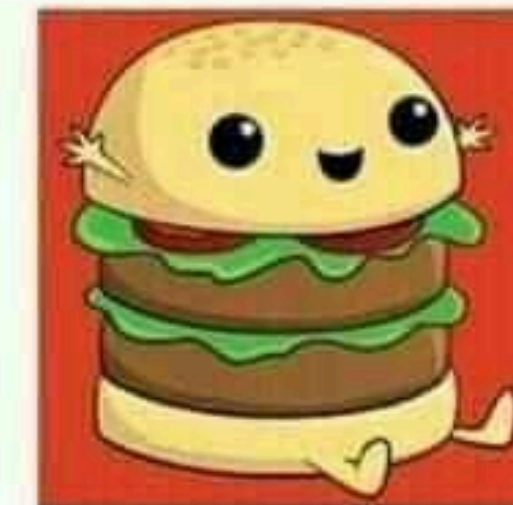
내용의 모양, 글꼴, 색상, 배경 이미지, 줄 간격, 페이지 레이아웃 등을  
설명하는데 사용되는 스타일 시트 언어 (구조물)



**HTML**



**HTML + CSS**

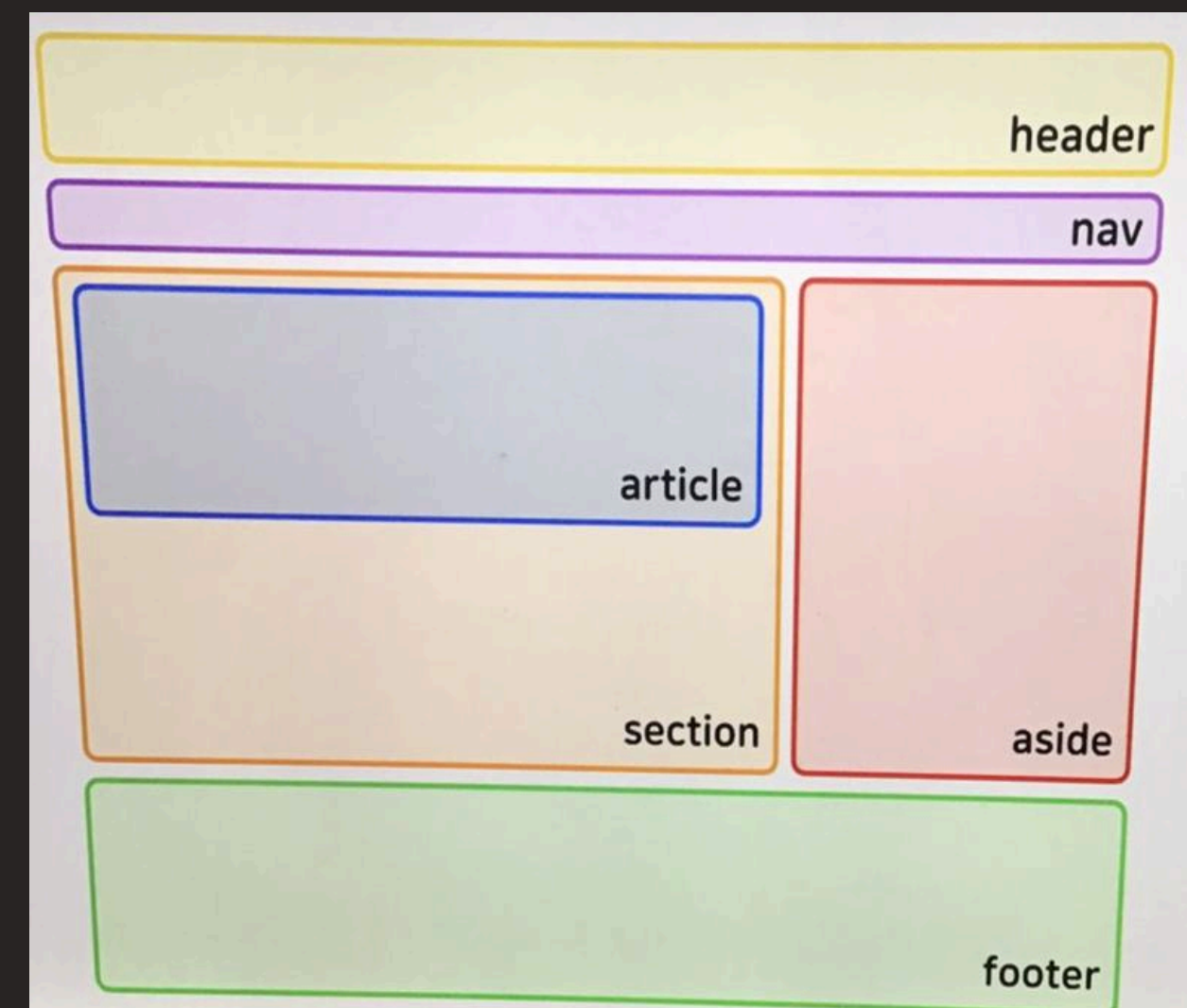
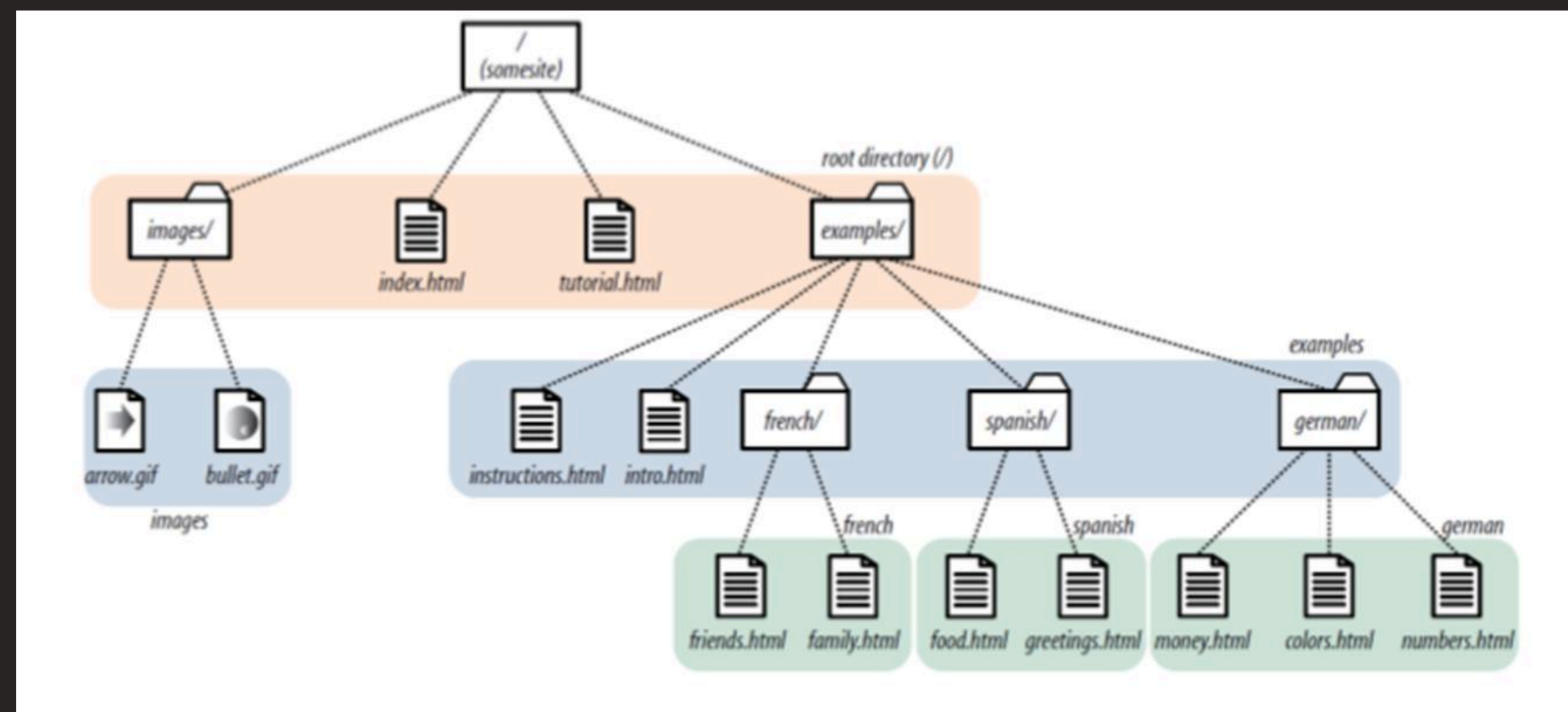


**HTML + CSS + JAVA SCRIPT**

## 2. HTML요소는?

# HTML 디렉토리

- 대규모 웹 사이트에서는 사이트의 각기 다른 섹션에 대한 페이지를 새 폴더에 배치하여 코드를 구성하는 것이 좋습니다
- 웹 사이트의 폴더를 디렉토리라고 합니다
- HTML5는 섹션에 의미를 부여하는 새로운 요소를 도입하였습니다.



# 섹션 태그

- **header**

웹페이지의 시작을 표현하거나 기사나 섹션의 맨 위에 표시  
주로 페이지, 섹션, 기사의 시작을 표현할 때 사용

- **nav**

사이트 탐색을 위해 사용되는 요소  
흔히 메뉴라고 불리는 요소

- **main**

사이트 내 메인 콘텐츠를 담는 태그  
문서에는 둘 이상의 <main> element가 없어야 한다.

- **aside**

광고, 추가들 설명 등 여분의 정보를 담는 태그  
주로 배경정보나 관계되는 링크, 사이드바, 로그인 정보등

- **section**

요소를 그룹화 하는데 주로 사용  
전체 페이지를 주요 섹션으로 나누거나 주제 섹션을 식별하는  
등 광범위한 용도로 사용

- **article**

잡지나 신문기사, 블로그 게시물, 댓글 또는 기타 항목에 사용  
article은 section안에 포함되는 경우가 많음.

- **footer**

footer 요소는 정보의 유형을 나타내는데 사용된다  
footer에도 <nav>태그를 추가하는 것이 가능하다.

# 글자태그

- `<p></p>`

단락을 나타내는 텍스트 문서의 가장 기본적 요소  
단락에는 텍스트, 이미지 및 기타 인라인 요소가 포함

- `<h1></h1>`

제목을 나타내는 태그로 h1~h6까지 6개 레벨의  
headings가 존재.  
`<h1>` 제목부터 시작하는 것이 가장 좋은 방법

# List

- **ul**

특별한 성격이나 요청이 나타나지 않는 항목 모음. 보통 `orderlist`나 일반 `list`의 항목을 묶는 용도로 사용.

- **ol**

항목의 순서가 중요한 목록. 앞에 번호표시가 있음

- **dl**

용어와 정의에 국한되지 않는 목록 표현

# Link

- `<a></a>`

└→ opening link tag

`[<a href = "https://www.google.com">]`구글로 이동!`[</a>]`



이동할 사이트 주소



텍스트 클릭하면 주소로 이동



# 그 외 text element

-<b>...</b>

키워드나 텍스트를 강조하는 강조 텍스트

-<i> ... </i>

이탤릭 체

-<s> ... </s>

취소줄 표시

-<u> ... </u>

밑줄 표시

- <small> ... </small>

글씨체를 보다 작게 표시

- <q> ... </q>

인용구 표시

이것은 **b tag** 입니다.

이것은 *i tag* 입니다.

이것은 ~~s tag~~ 입니다.

이것은 u tag 입니다.

이것은 small tag 입니다.

이것은 "q tag" 입니다.

# 3. Form과 Field

## Form이란?

-구매, 비행기 표 예약, 청원서 서명, 사이트 검색, 트윗 게시 등 이러한 모든 상호작용은 양식에 의해 처리됩니다

-HTML 양식은 사용자로부터 데이터를 수집하는 집합을 제공합니다.

-Form 은 두가지 방법으로 작동합니다.

-GET, POST

- HTML 마크업을 사용하여 작성된 페이지 자체에 표시되는 양식 (버튼, 입력 필드 및 드롭 다운 메뉴는 폼 컨트롤이라고 통칭합니다. )

-정보를 처리하는 서비스의 응용프로그램 또는 스크립트 양식에 의해 수집되고 적절한 응답을 반환합니다.

# Form Element

- Form element는 양식(from)의 모든 콘텐츠를 담는 그릇입니다..
- Form controls는 텍스트 입력필드 및 버튼과 같습니다.
- 블록 요소(ex: h1, p 및 목록)도 포함 할 수 있습니다.
- 다른 form element 를 중복으로 포함해서는 안됩니다.
- Form 요소에는 서버에서 form을 처리하고 상호작용하기 위해 필요한 몇가지 속성이 있습니다.

# Form Tag 속성

-Form tag attribute 에는 name, action, method, target등이 있다. 이러한 폼 속성을 이용하여 전송할 때 어디로 보내야 하는지, 또 어떤 방법으로 보낼지를 정하게 됩니다.

Form tag attribute는 다음과 같습니다.

-action :form을 전송할 서버 쪽 응용 프로그램 또는 스크립트의 위치(URL)을 지정합니다.

-name : form 을 식별하기 위한 이름을 지정합니다.

-target : action 에서 지정한 스크립트 파일을 현재 창이 아닌 다른 위치에 열도록 지정합니다.

-method : 서버로 보내야하는 정보의 전송 방법을 지정한다. 인코딩 된 데이터를 전송하는 방법은 POST, GET 두가지가 있습니다

## text entry Fields

- 비밀번호 입력 필드 <input type="password">
- 검색 필드 : <input type= "search">
- 이메일 주소 작성 필드 : <input type = "email">
- 전화번호 작성 필드 : <input type = "tel">
- URL 작성필드 : <input type=" url " >

수고하셨습니다 :)

