

Session 0

개발환경 설정 & HTML, CSS 기초

NEXT X LIKELION 김지성

목차

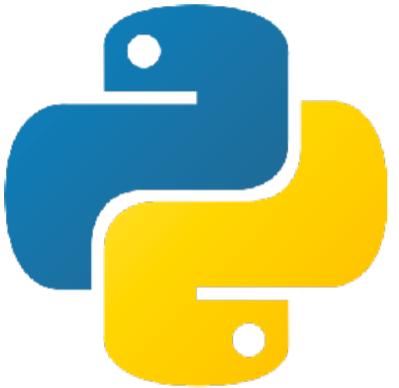
1. 개발 환경 설정
2. 왜 웹개발인가?
3. 웹의 구조
4. HTML, CSS 기초
5. Github 사용해보기

| 11기 NEXT 첫 세션!

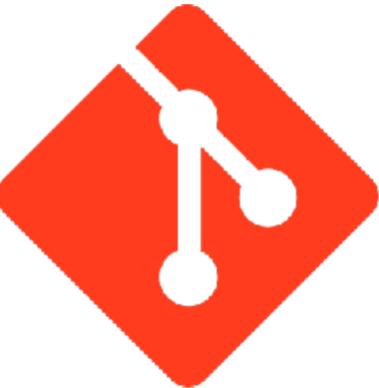


와 개발이당-!

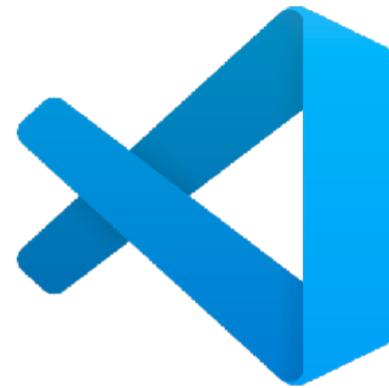
| 개발환경 설정



Python
(쉬운 프로그래밍 언어)



Git
(버전 관리 시스템)



Visual Studio Code
(텍스트 에디터)

| 파이썬 설치

macOS 유저는 **Terminal**, Window 유저는 **PowerShell**에서
`python --version` 으로 설치 유무 검색

안뜨면 설치 에러나면 설치 3.10.10 아니면 설치 ㄱㄱ

아나콘다 깔려있으면 제!발! 삭제하고 설치하세요!

| 파이썬 설치

www.python.org/download

11기 공식 파이썬 버전 == 3.10.10

윈도우 유저는 Add Python 3.10 to PATH 필수

| Git 설치

macOS 유저는 **Terminal**에서 `git --version` 으로 설치 유무 검색.
Window 유저는 왼쪽 하단 검색창에서 **Git Bash** 프로그램 유무 검색.

이미 있으면 설치 x, 없으면 설치 ↗

ONLY Window!
(Mac은 터미널에 이미 내장되어 있음)

| Git 설치

<https://www.git-scm.com/downloads>



Window 유저

Download for Windows

Click here to download the latest (2.35.1) 32-bit version of Git for Windows. This is the most recent maintained build. It was released about 1 month ago, on 2022-02-01.

Other Git for Windows downloads

Standalone Installer
[32-bit Git for Windows Setup.](#)

[64-bit Git for Windows Setup.](#) 

Portable ("thumbdrive edition")
[32-bit Git for Windows Portable.](#)

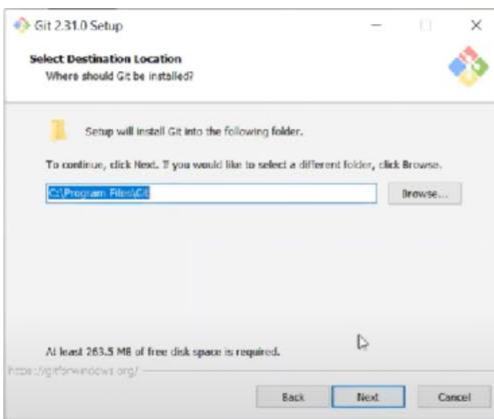
[64-bit Git for Windows Portable.](#)

Using winget tool

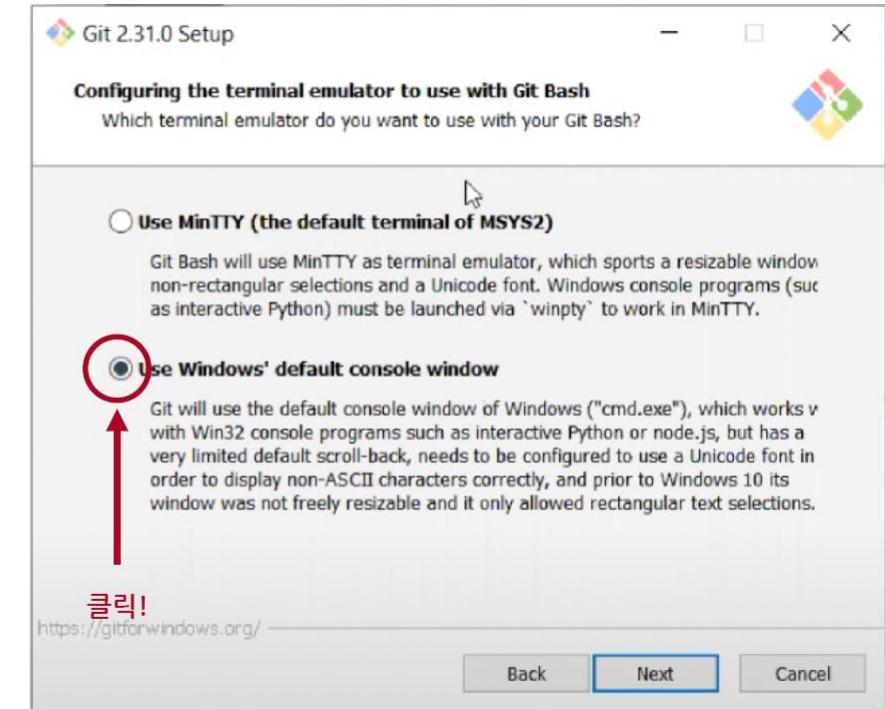
Install [winget tool](#) if you don't already have it, then type this command in command prompt or Powershell.

```
winget install --id Git.Git -e --source winget
```

Git 설치



계속 next
누르다가..



Git 초기 설정

맥은 Terminal,
윈도우는 Git Bash에서

```
-zsh

git config --global user.name ${user.name}
ex) 자신의 깃헙 아이디가 hello123이면
    git config --global user.name hello123

git config --global user.email ${user.email}
ex) 자신이 깃헙을 가입한 이메일이 hello123@korea.ac.kr이면
    git config --global user.email hello123@korea.ac.kr

git config --list
-> git username과 email 확인
```

| VS Code 설치

<https://code.visualstudio.com>



Window 유저

설치 후 git bash에서
code . 쳐서 vscode 뜨면 설치 성공

* code . 의미: vscode를 현재 경로에서 열겠다.
(‘.’은 현재 경로를 의미하고, code는 (vs)code를 가리킴.)



MacOS 유저

다운로드 후 code . 따로 설정 필요!

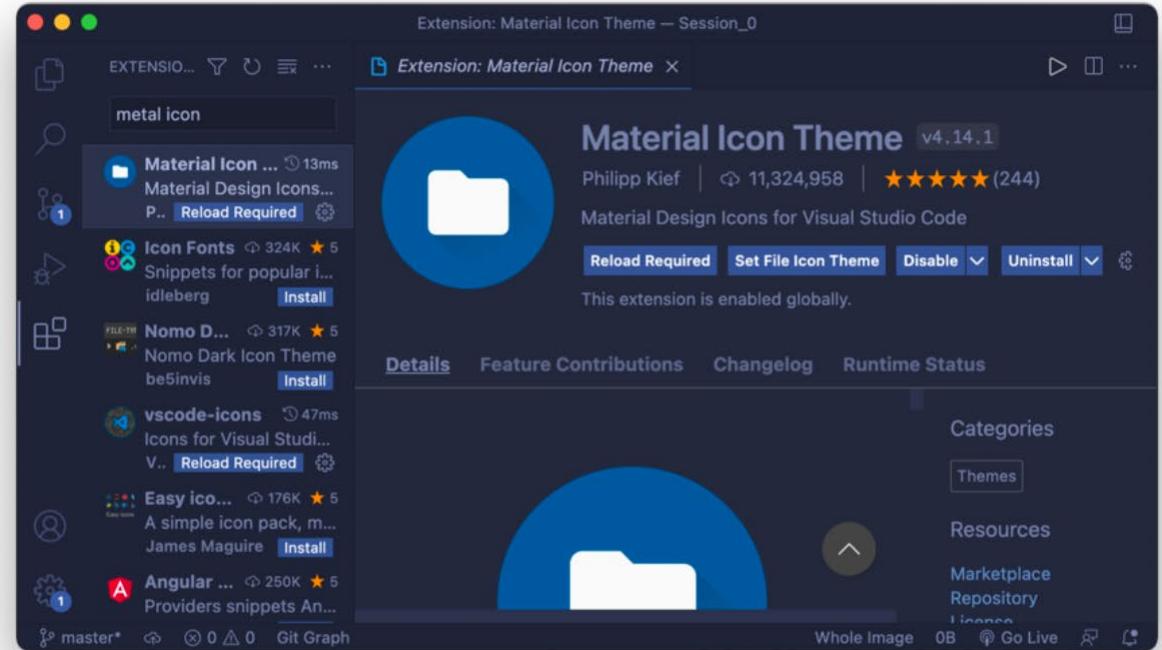
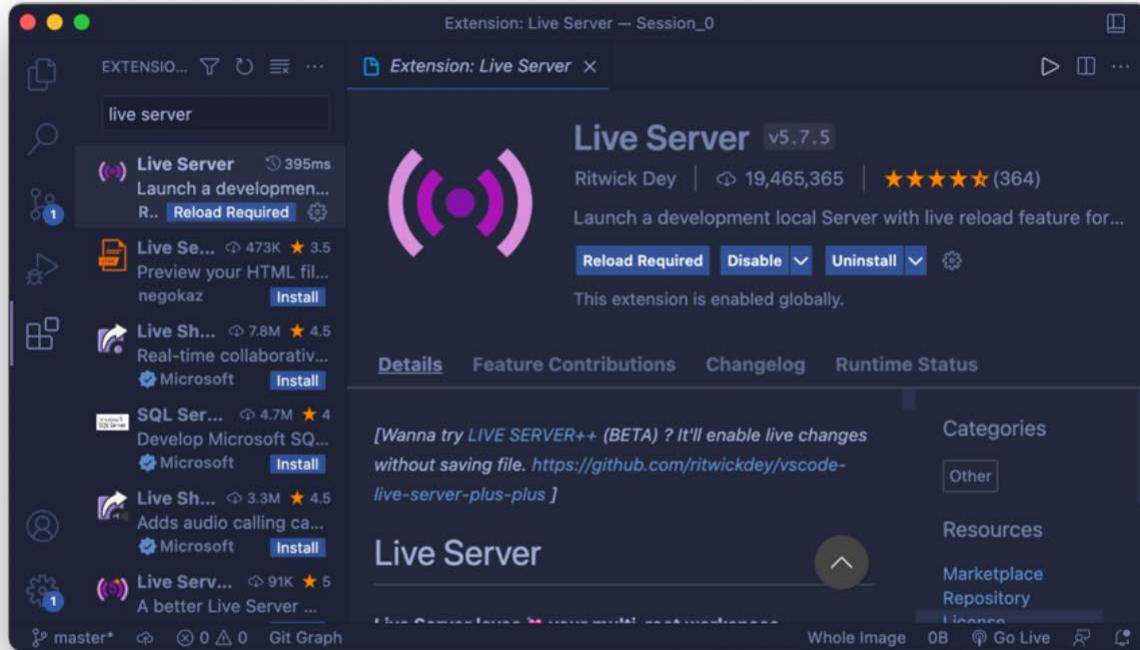


```
>shell command
Shell Command: Install 'code' command in PATH
Shell Command: Uninstall 'code' command from PATH
```

Command Palette ($\text{⇧}\text{⌘}\text{P}$) 연 다음 shell command 검색 & 설치

VS Code 설치

Live Server & material icon theme 설치하기



| 왜 웹개발인가?

웹보다는 앱개발이 대세 아닌가요?



| 왜 웹개발인가?

1. 낮은 획득 비용

| 왜 웹개발인가?

APP

CAC(획득 비용)

\$4.18

WEB

\$2.69

Retention Rate(재방문률)

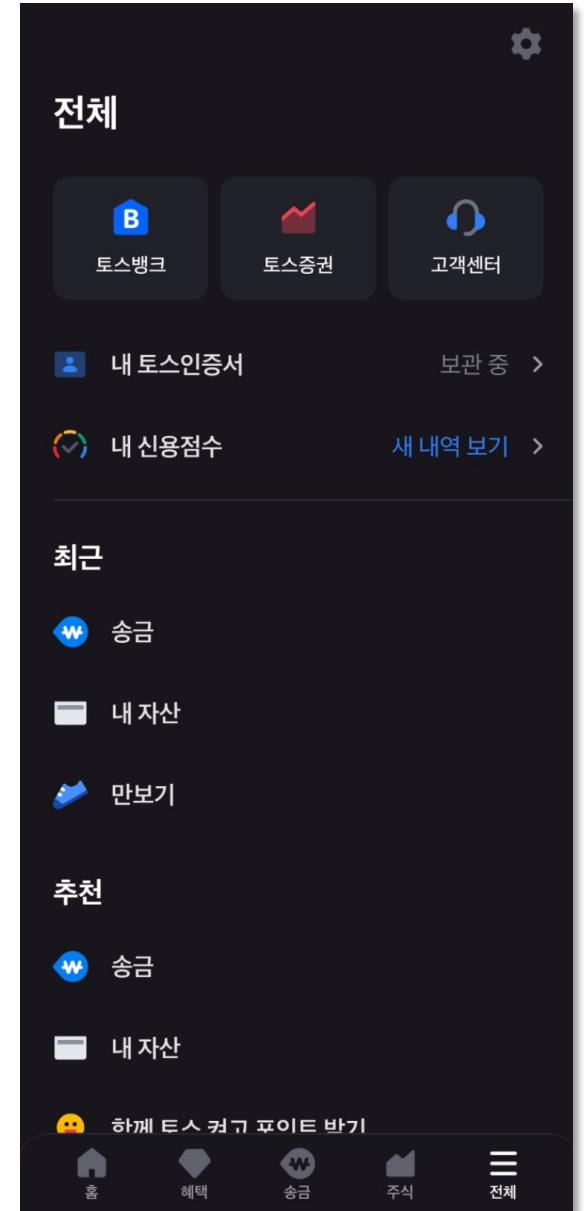
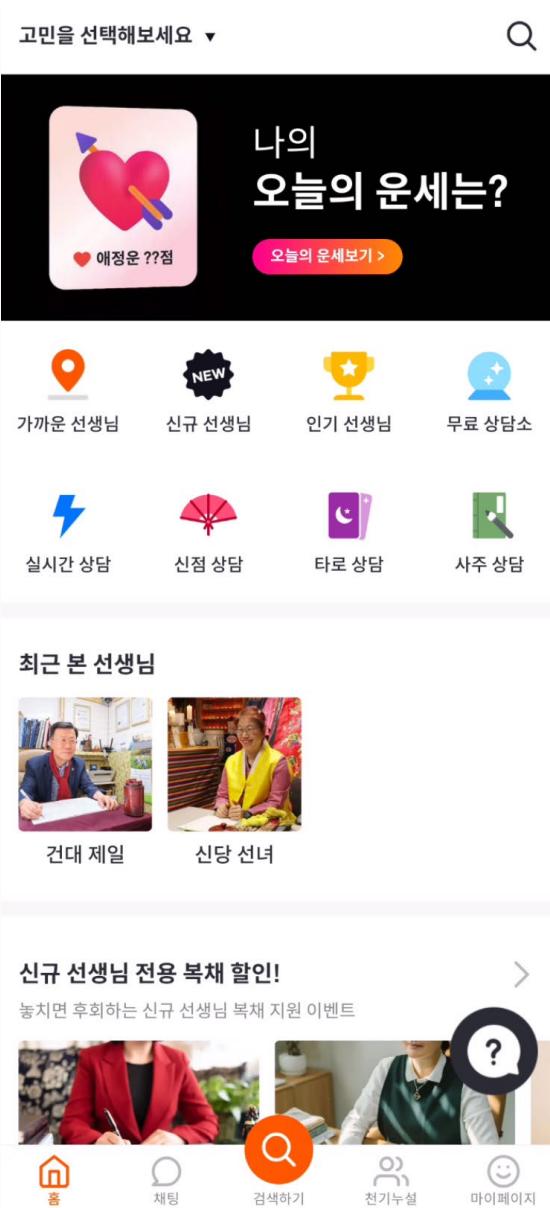
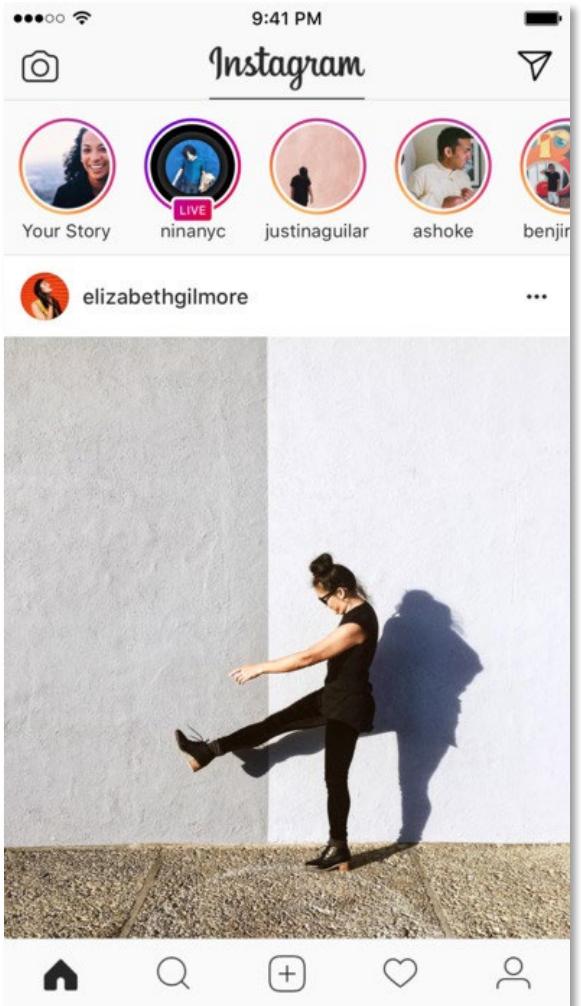


MVP용으로 가장 좋음...!

| 왜 웹개발인가?

2. 뛰어난 확장성

왜 웹개발인가?



| 왜 웹개발인가?

3. 재미

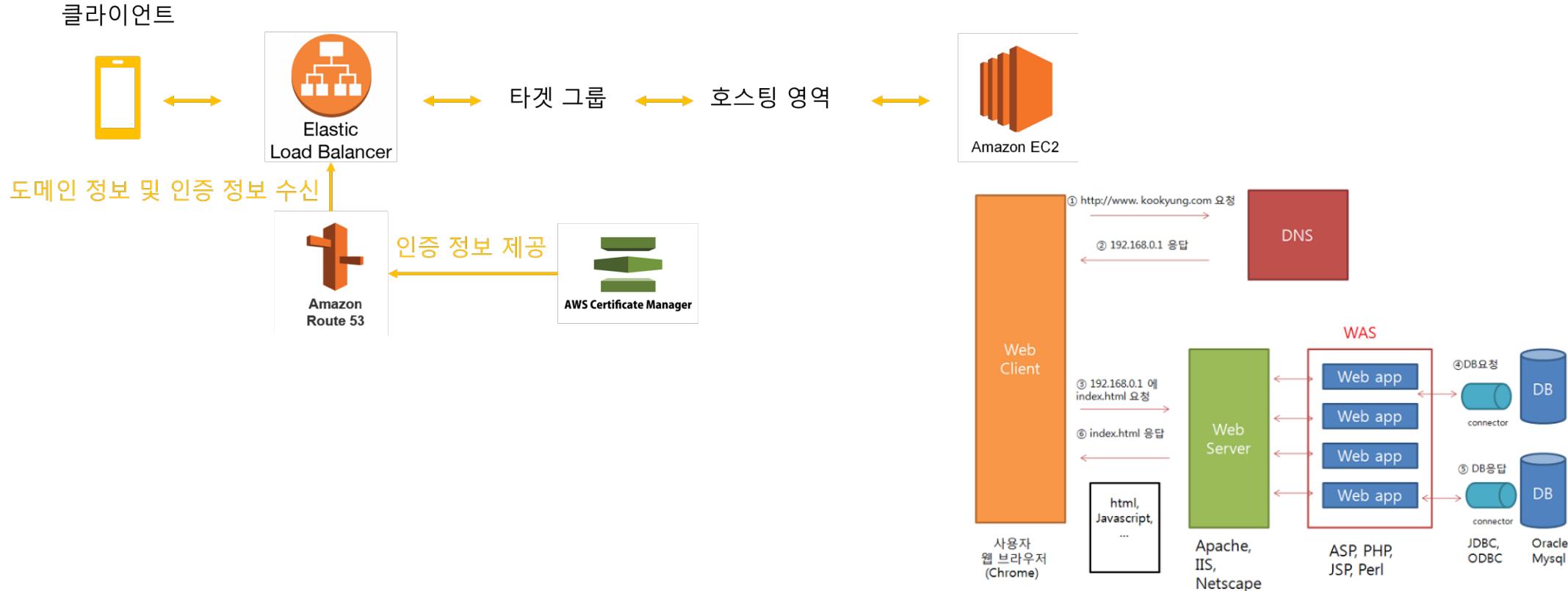
| 왜 웹개발인가?



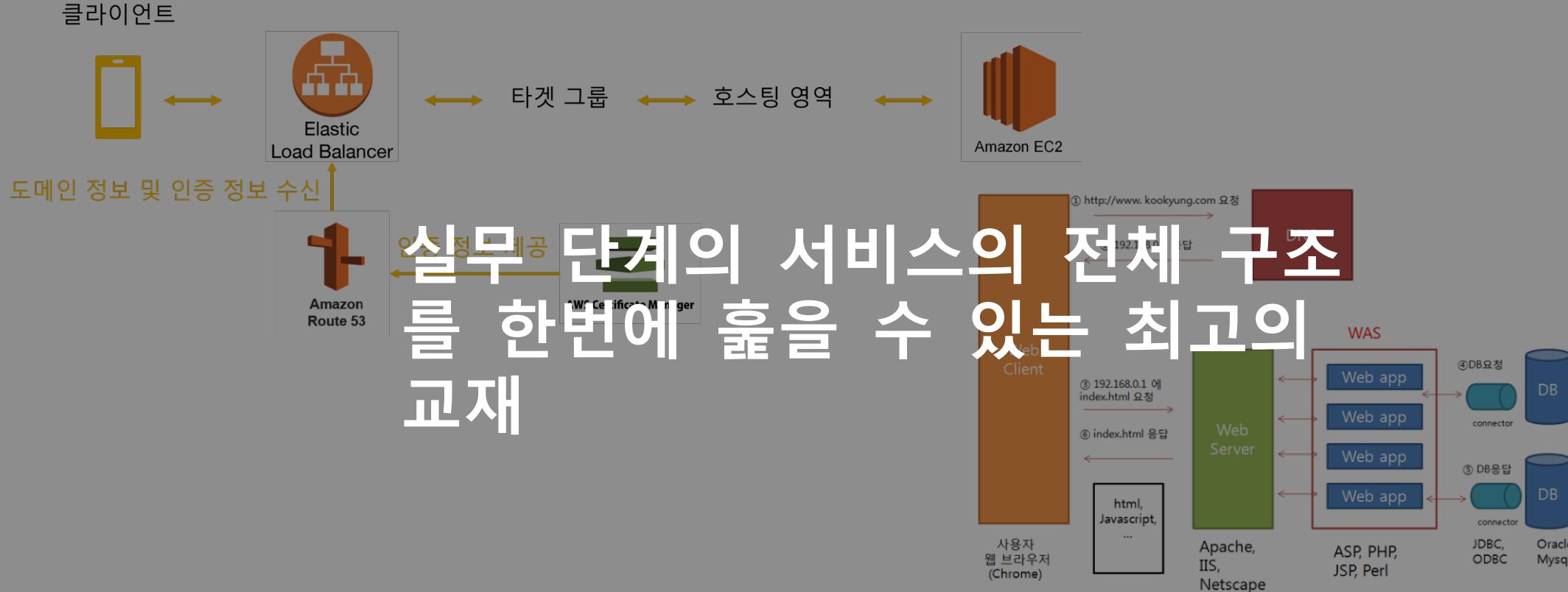
VS



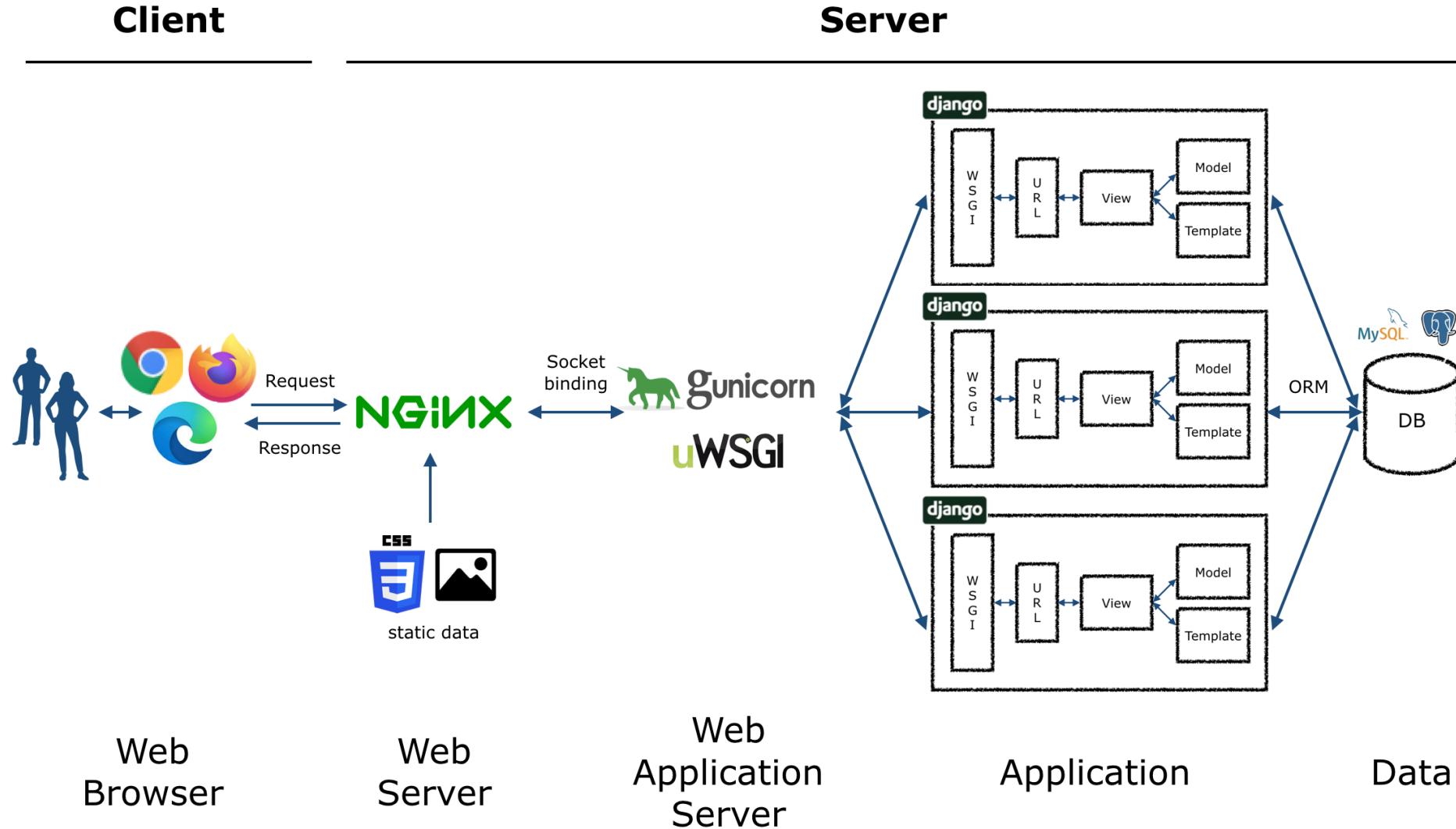
왜 웹개발인가?



왜 웹개발인가?



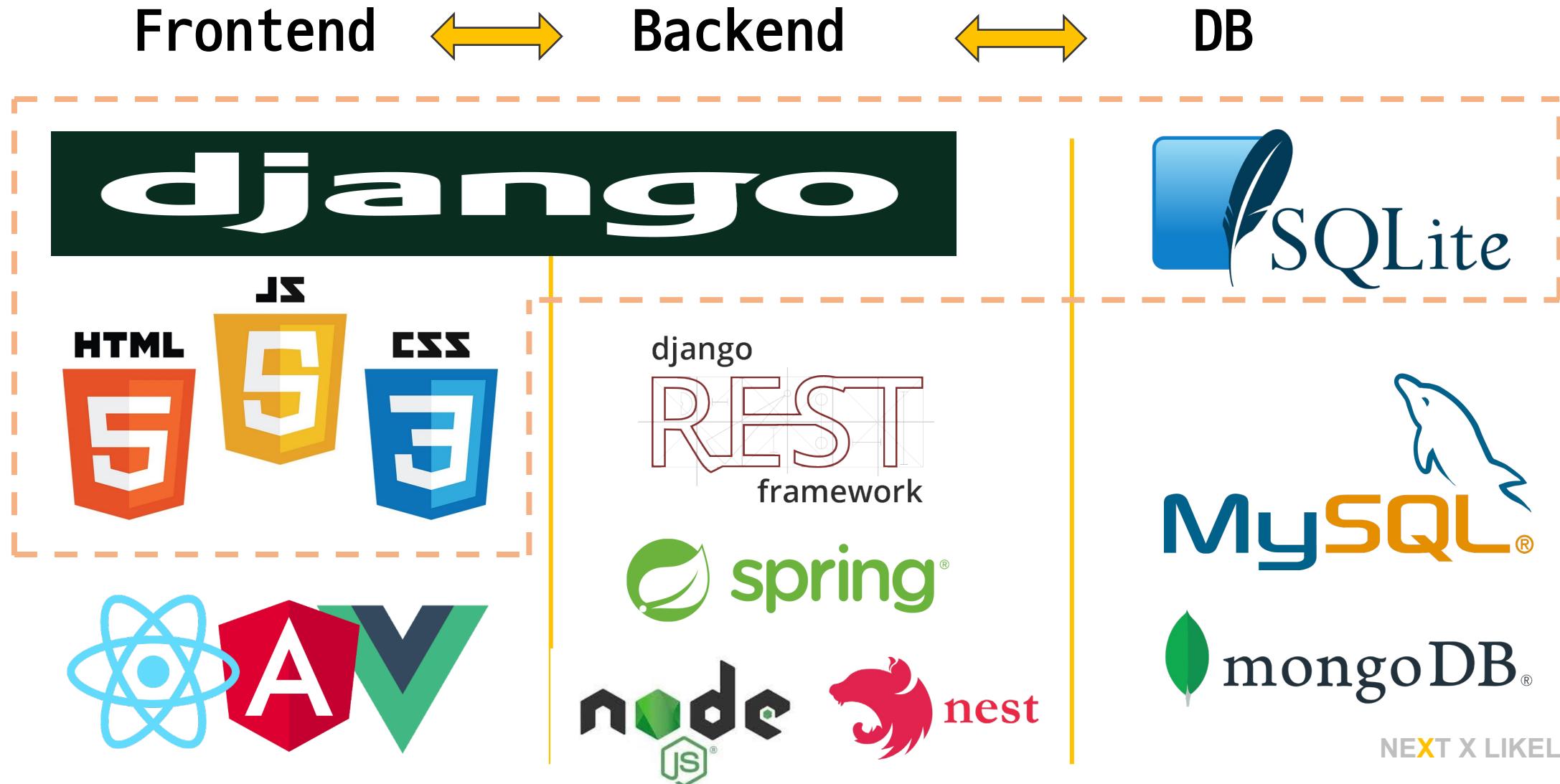
웹 서비스 구조



| 웹 서비스 구조



| 웹 서비스 구조



| 웹 서비스 구조-FE

HTML



뼈



CSS



피부



JS



근육



HTML 기초

| HTML이란?

Hyper Text Mark-up Language

한 문서에서 다른 문서로 즉시 접근할 수 있는,
태그 및 텍스트 기반으로 구조화된 문서

웹 페이지 구조 혹은 데이터 작성을 위한 마크업 언어.

HTML 문서 구조

F12 누르고 네이버 손민수하기

```
<html>
<head>
</head>
<body>
<script>                               </script>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width,
initial-scale=1.0">
<title>Document</title>
</head>
<body>
</body>
</html>
```

HTML 문서 구조

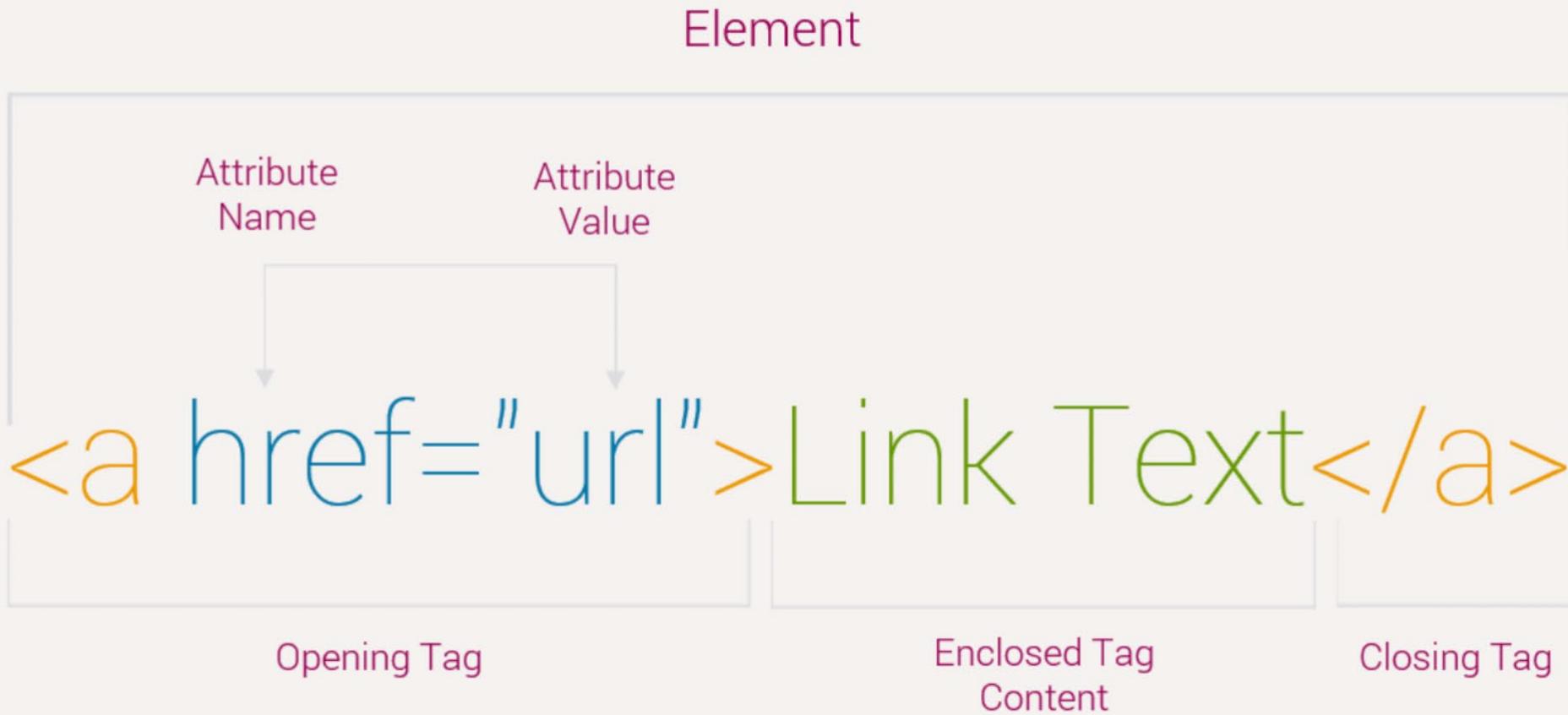
F12 누르고 네이버 손민수하기

네이버 노잼;;

세계 최초 웹사이트 정도는 봐야지;;;

<http://info.cern.ch/>

HTML 태그



HTML 태그 종류

<h1> ~ <h6>

<form>

<p>

<script>

<a>

<div>

:

HTML태그는 절대로 다 외울 수 없습니다!

자주쓰는것만 외우고 다른건 그때그때 검색!

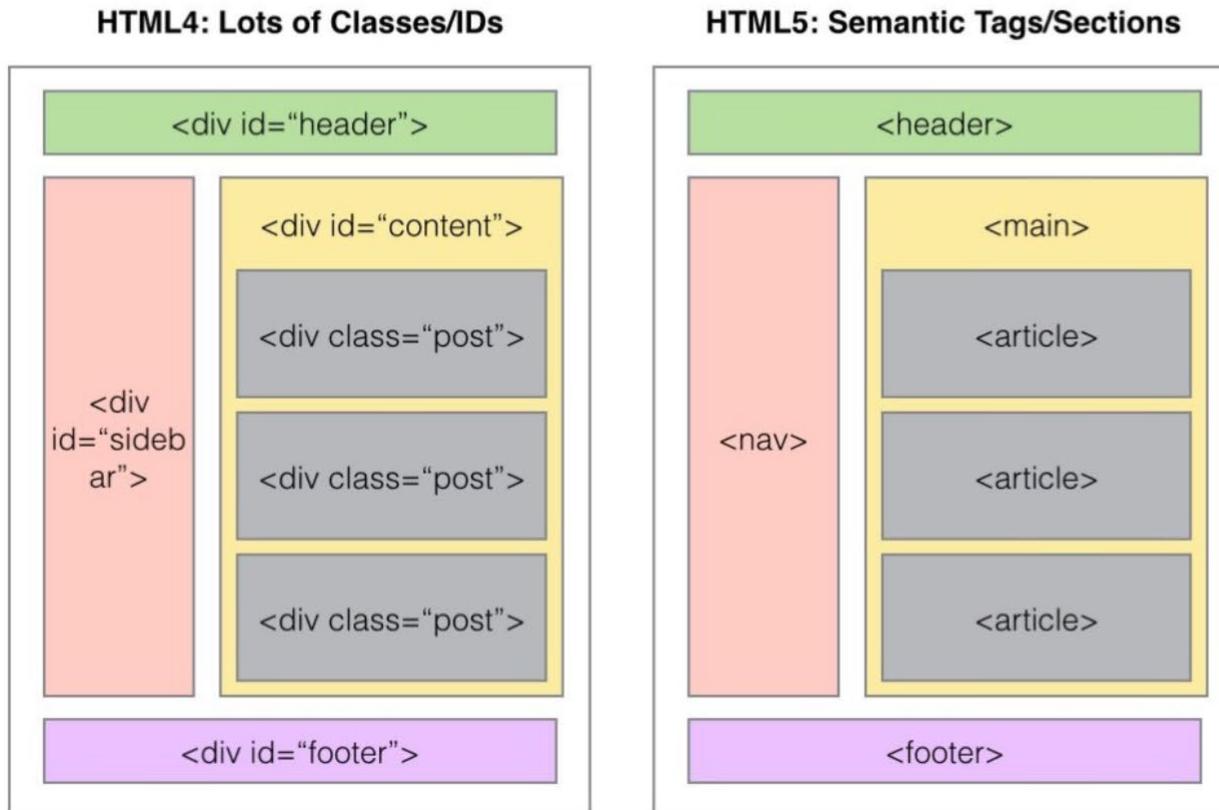
<https://developer.mozilla.org/ko/docs/Web/HTML/Element>

<https://www.w3schools.com/tags/default.asp>

HTML5- Semantic tag

의미를 담은 태그 – 왕위를 계승하는 중입니다, HTML4!

HTML4 vs HTML5 Page Structure on a Blog



<nav>
<main>
<header>
<footer>
<section>
<article>
<details>
<aside>
⋮

Tags:

사진도 넣어봅시다

```
< img src="..../image.png" alt="description" />
```

첨부하려는
이미지 경로 (필수)

오류시 이미지
대체할 수 있는 설명

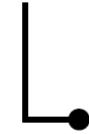
절대경로: 페이지나 파일의 고유 주소(root기준)

상대경로: 현재 위치 기준으로 한 상대적 주소

Tags: <form></form>

모든 입력의 부모 태그

```
< form action="server.html" method = "GET" />
```



폼 전송할 서버쪽
스크립트 파일 지정



폼을 서버에 전송할
http 메소드 지정

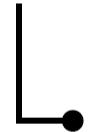
Tags: <input></input>

뭐라도 입력은 받아야죠

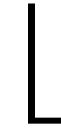
```
< input type="type" name ="username" value ="name" />
```



입력 받을
data type



Form data 이름



Form data 내용

Tags: <a>

링크 걸고 싶으면 a를 기억하세요

< a href=https://naver.com>네이버로 이동



• 이동할 URL

HTML 실습- 자기소개 페이지 만들기

HTML 맛보기

자기 소개 페이지를 만들어보자.

Browser title,

Text

List

Link

:

CSS 기초

Cascading Style Sheets

스타일, 레이아웃 등 사용자에게 문서를 표시하는 방법을 지정하는 언어.

* cascading : 위에서 아래로 흐르는, 상속 또는 종속하는.

css 기본 문법

선택자 {속성: 값;}

```
p {  
    font-size : 1.2em;  
    color : gray;  
}
```

선택자

HTML 요소 선택하기

*

div, p, body
Tag 직접 선택

*

div

{

width : 100%;
box-sizing : border-box;

}

#id
id로 선택

#id {

font-size: 24px;

}

.class
class로 선택

.class {

font-size: 24px;

}

선택자

HTML 요소 선택하기

*

모두 선택

* {

width : 100%;

Selector Hierarchy(선택자 계층) -> Cascading

#id

#id {

font-size: 24px;

div, p, body
Tag 직접 선택

div {

font-size: 12px;

color : black;

.class
class로 선택

.class {

font-size: 24px;

선택자

HTML 요소 선택하기

*

모두 선택

div, p, body
Tag 직접 선택

* {
width : 100%;
box-sizing : border-box;
}

ID와 CLASS
id : 고유한 요소로 한 문서에 하나만 존재할 수 있음.
class : 공통 요소로 다른 요소에 같은 class 반복 사용 가능

div {
font-size: 12px;
color : black;
}

#id {
id로 선택
}

.class {
class로 선택
}

#id {
font-size: 24px;
}
class {
font-size: 24px;
}

| id, class

개발자님께서 멋진 이름을 주셨어요!

```
<a id="naverBtn" class="btn testBtn"  
 href="https://naver.com">네이버로 이동</a>
```

CSS 실습 – CSS 적용하기

파일을 따로 관리합시다

```
<head>  
...  
<link rel = "stylesheet" href=".style.css"/>  
</ head>
```

요소의 크기관리 : box-sizing

요소가 만들어지는 구조. 얘 외워야 웹개발 인생 편해집니다



CSS 실습 : 자기소개 페이지 꾸미기

오늘은 CSS 맛만 보자구요



김지성입니다.

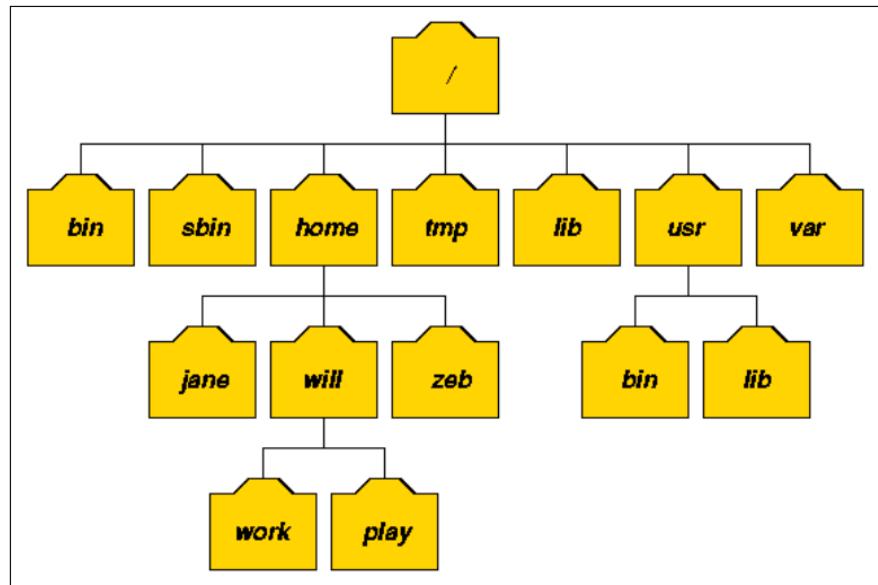
- 고려대학교 영어영문학과였다가
- 고려대학교 산업경영공학부가 되었습니다

사실 이것도 리스트인데
점이 없어요

Git 입문

File System 알아보기

Git 입문 1



디렉토리 구조: 계층 트리 구조

하위 폴더에 접근하려면 현재 경로에서부터 차례대로

ex1) /home/will/play

ex2) /usr/bin

CLI(Command Line Interface) 배우기

Git 입문 2



CLI (Command Line Interface)

텍스트 기반의 명령어 입력으로 시스템 조작

<→ GUI (Graphic User Interface) 버튼, 아이콘, 이미지
등과 같은 운영 체제와 상호 작용할 수 있는 비주얼을 제공

CLI(Command Line Interface) 배우기

Basic Command

[경로 탐색] **pwd**: print (current) working directory

[경로 이동] **cd**: change directory

[파일 확인] **ls**: list segments

ls -a: list segments of all

ls -l: list directory with long listing

ls -al: list directory of all with long listing

[파일 생성] **touch**: create a file

mkdir: make a directory

[파일 제거] **rm**: remove a file

rm -rf: remove forcibly including directory and the files in it

CLI(Command Line Interface) 배우기

Basic Command

pwd

현재 절대 경로를 확인합니다.

```
chunhyerin@iMac Session_0 % pwd  
/Users/chunhyerin/Desktop/Session_0
```

CLI(Command Line Interface) 배우기

Basic Command

cd 원하는_경로

원하는 절대 경로, 상대 경로 주소값으로
이동합니다.

상위 폴더인 Desktop 으로 이동
chunhyerin@iMac Session_0 % cd ..
chunhyerin@iMac Desktop % cd Session_0
chunhyerin@iMac Session_0 %

Session_0 으로 이동

절대경로 - root directory에서 해당 directory까지의 경로
(Pwd 를 사용했을 때 나오는 경로)

상대경로 - 현재 directory에서 해당 directory까지의 경로

“.” - 현재 경로를 의미

“..” - 위의 경로를 의미

CLI(Command Line Interface) 배우기

Basic Command

ls 현재 디렉토리 위치에 내용 나열

ls -a 숨김 파일 (.., .git, .gitignore 등)
포함해서 모두 표시

ls -l 좀 더 자세한 정보 (수정 날짜 등)
포함해서 출력

ls -al ls -a, ls -l 둘 다 같이 섞어서
출력

```
chunhyerin@iMac Session_0 % ls
README.md      Session0.png      Session0.txt
chunhyerin@iMac Session_0 % ls -a
.              ..              .git          README.md      Session0.png      Session0.txt
chunhyerin@iMac Session_0 % ls -l
total 8
-rw-r--r--  1 chunhyerin  staff  103  3 10 10:20 README.md
-rw-r--r--  1 chunhyerin  staff    0  3 10 09:04 Session0.png
-rw-r--r--  1 chunhyerin  staff    0  3 10 09:04 Session0.txt
chunhyerin@iMac Session_0 % ls -al
total 8
drwxr-xr-x  6 chunhyerin  staff  192  3 10 09:32 .
drwx-----+ 7 chunhyerin  staff  224  3 10 09:03 ..
drwxr-xr-x 12 chunhyerin  staff  384  3 10 09:35 .git
-rw-r--r--  1 chunhyerin  staff  103  3 10 10:20 README.md
-rw-r--r--  1 chunhyerin  staff    0  3 10 09:04 Session0.png
-rw-r--r--  1 chunhyerin  staff    0  3 10 09:04 Session0.txt
```

CLI(Command Line Interface) 배우기

Basic Command

`touch 파일_이름`

현재 디렉터리에서 “파일_이름”을
가진 파일을 생성합니다.

```
chunhyerin@iMac Session_0 % touch hello.txt
chunhyerin@iMac Session_0 % ls
README.md          Session0.png      Session0.txt    hello.txt
```

CLI(Command Line Interface) 배우기

Basic Command

rm 파일_이름

현재 디렉터리에서 “파일_이름”을 가진 파일을 삭제합니다.

rm - rf 폴더_이름

현재 디렉터리에서 “폴더_이름”을 가진 파일을 삭제합니다.

```
chunhyerin@iMac Session_0 % ls
README.md      Session0.png    Session0.txt    hello.txt      newFolder
chunhyerin@iMac Session_0 % rm hello.txt
chunhyerin@iMac Session_0 % ls
README.md      Session0.png    Session0.txt    newFolder
chunhyerin@iMac Session_0 % rm -rf newFolder
chunhyerin@iMac Session_0 % ls
README.md      Session0.png    Session0.txt
```

Git 기초 버전관리

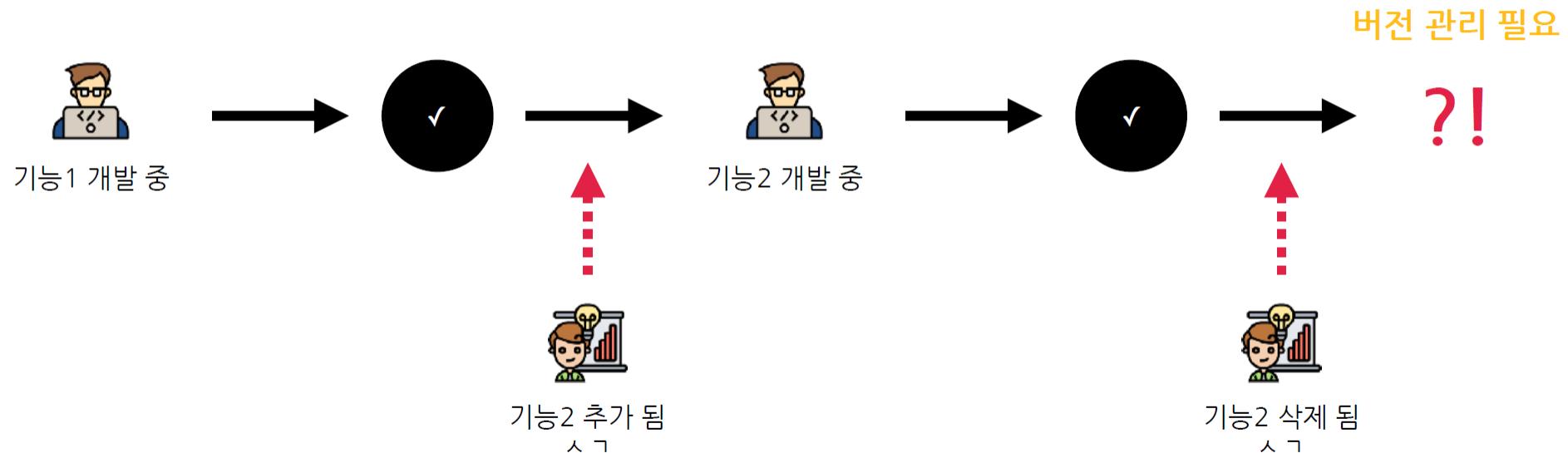
깃이 뭔데. 그거 어떻게 하는건데.

커밋이요? 머지요? 뭐요?



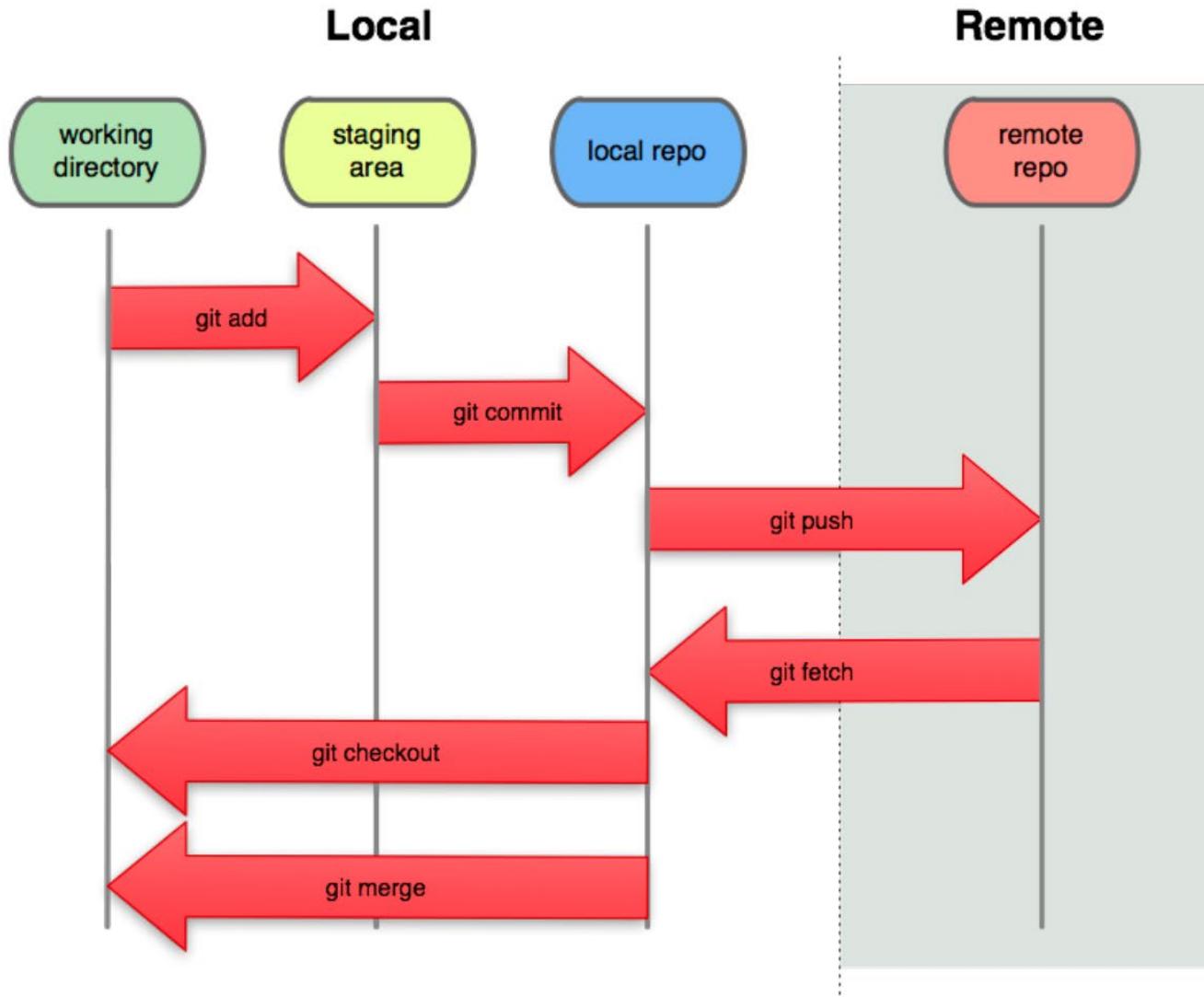
Git 기초 버전관리

그거 왜쓰는데



Git 기초 버전관리

Git 기능 정리



Git 기초 버전관리

아무것도 안했을 때 초기 상태



Git 기초 버전관리

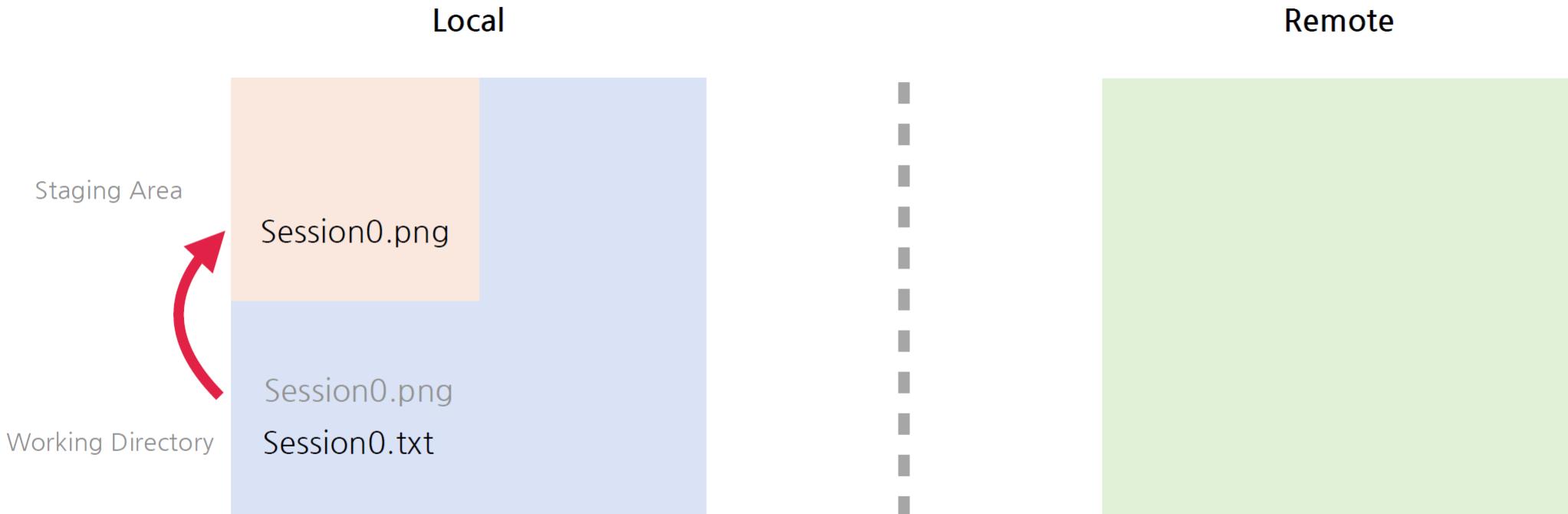
스테이징 영역 활성화 및 추적 시작



`git init`

Git 기초 버전관리

스테이징 영역에 파일 추가

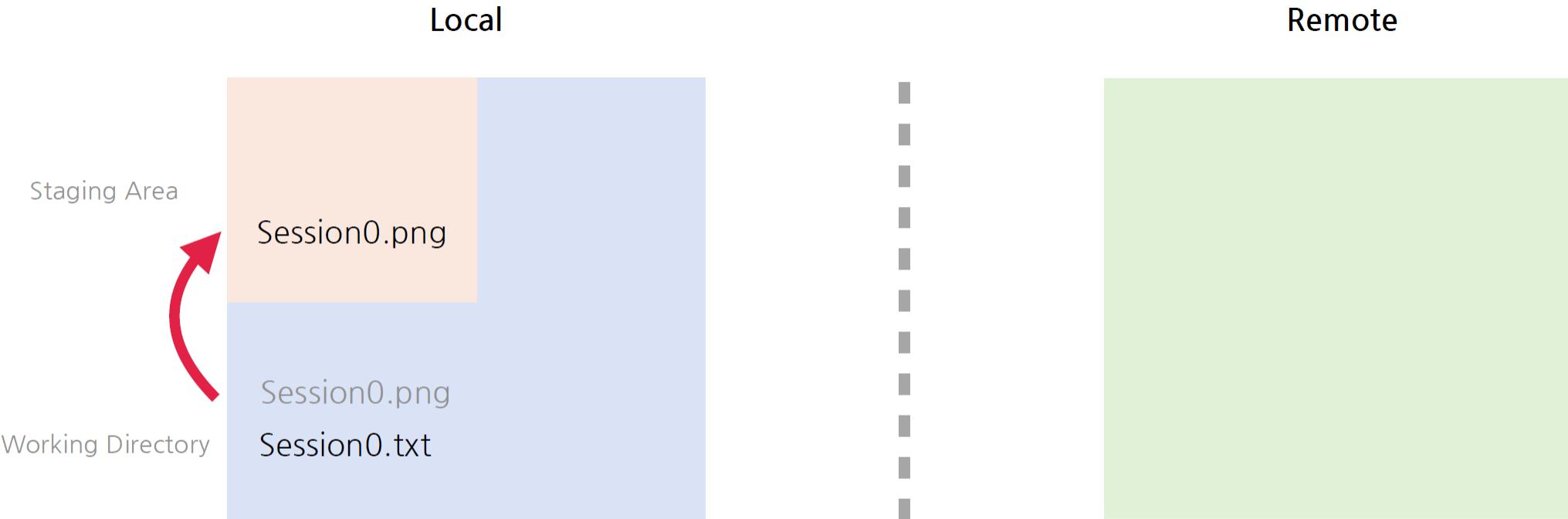


git add \${file_name}

`git add index.html` `git add .`

Git 기초 버전관리

Local Repository에 확정



git commit -m "\${커밋메세지}"

git commit -m "first commit"

Git 기초 버전관리

Git 기록 관리

```
chunhyerin@iMac Session_0 % git log
commit c3f15a64ed342864b782469bfb2ebae2525a3eba (HEAD -> master)
Author: deborah91298 <deborah91298@korea.ac.kr>
Date:   Thu Mar 10 09:34:09 2022 +0900
```

첫 세션 푸시

git log

Git 기초 버전관리

Git 기록 관리



```
git add .  
git commit -m "second commit"
```

“.” - 현재 경로를 의미.
즉, 현재 경로에 있는 모든
파일 및 폴더를 포함함

Github에 푸시하기

Github 그게 뭔데. 그거 어떻게 하는건데.



Github

협업 지원, 오픈 소스의 성지, 코드 관리 페이지

Github에 푸시하기

Remote Repository 현재 상태

만들어져 있는데 연결이 안 됐네?

Local

Staging Area

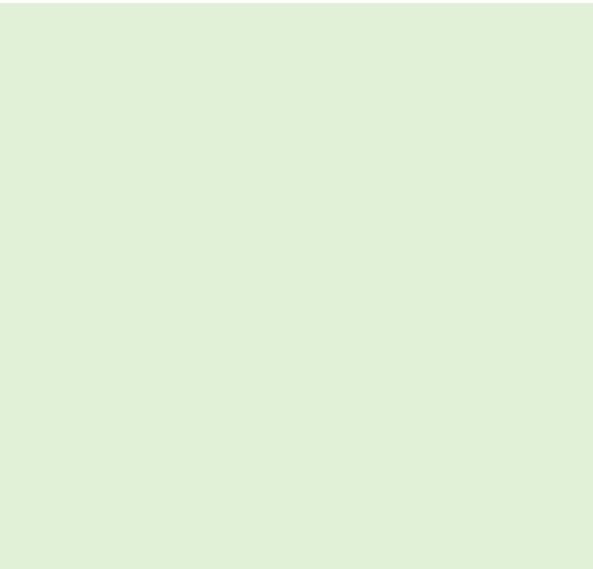
README.md

Session0.txt

Session0.png

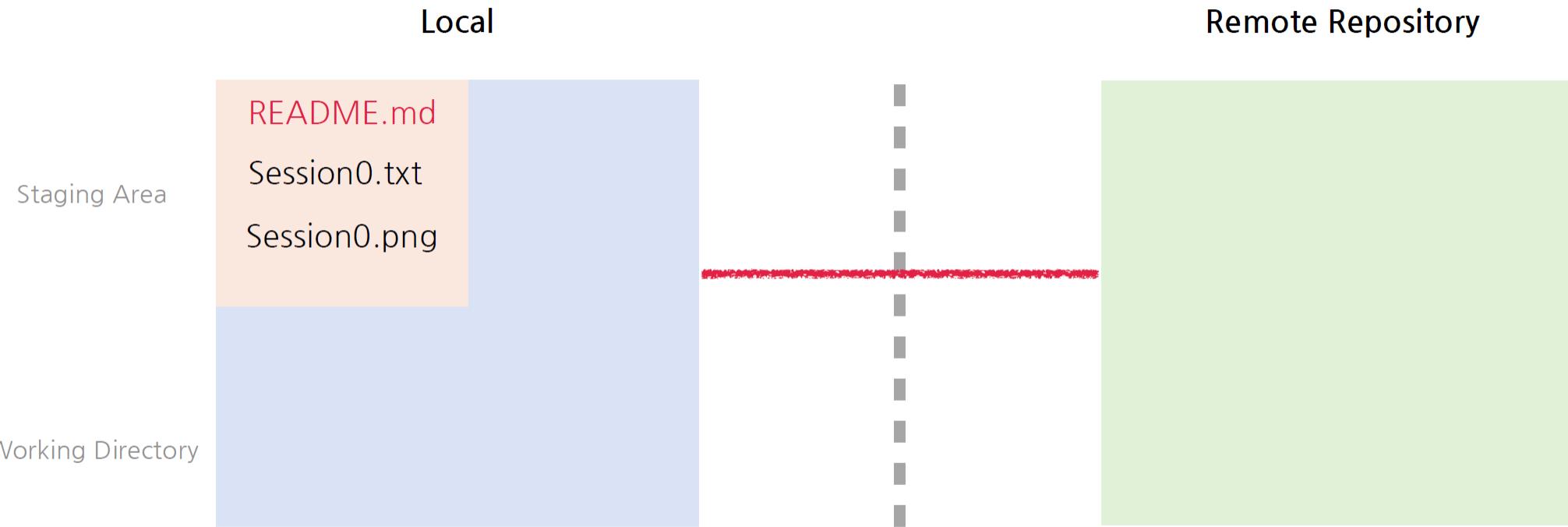
Working Directory

Remote



Github에 푸시하기

Local Repository와 Remote Repository 연결하기



`git remote add oriain “보이 Repository 주소”`

Github에 푸시하기

Local Repository와 Remote Repository 연결 확인

```
chunhyerin@iMac Session_0 % git remote -v
origin  https://github.com/deborah91298/Next_Likelion.git (fetch)
origin  https://github.com/deborah91298/Next_Likelion.git (push)
```

git remote -v

Github에 푸시하기

Github Repository에 접근 가능한 Personal Access Token 생성

My profile > Settings > Developer Settings > Personal access tokens > Generate new token

The screenshot shows the GitHub Developer Settings page under the Personal access tokens section. The sidebar on the left has three options: GitHub Apps, OAuth Apps, and Personal access tokens, with Personal access tokens being the active tab. The main area displays a table of generated tokens. One token is listed:

| repo — admin:repo_hook, delete_repo, gist, repo, user | Last used within the last week | Delete |
|---|--------------------------------|--------|
| ⚠ This token has no expiration date. | | |

Below the table, a note states: "Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#)".

At the bottom, there is a footer with links: © 2022 GitHub, Inc., Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, About, and a GitHub logo.

Github에 푸시하기

Github Repository에 접근 가능한 Personal Access Token 생성

Note
 repo

What's this token for?

Expiration
This token has no expiration date. To set a new expiration date, you must [regenerate the token](#).

Select scopes
Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

| | |
|--|---|
| <input checked="" type="checkbox"/> repo | Full control of private repositories |
| <input checked="" type="checkbox"/> repository_status | Access commit status |
| <input checked="" type="checkbox"/> repository_deployment_status | Access deployment status |
| <input checked="" type="checkbox"/> public_repo | Access public repositories |
| <input checked="" type="checkbox"/> repository_invitations | Access repository invitations |
| <input checked="" type="checkbox"/> security_events | Read and write security events |
| <input type="checkbox"/> workflow | Update GitHub Action workflows |
| <input type="checkbox"/> write_packages | Upload packages to GitHub Package Registry |
| <input type="checkbox"/> read_packages | Download packages from GitHub Package Registry |
| <input type="checkbox"/> delete_packages | Delete packages from GitHub Package Registry |
| <input type="checkbox"/> admin_org | Full control of orgs and teams, read and write org projects |
| <input type="checkbox"/> write_org | Read and write org and team membership, read and write org projects |
| <input type="checkbox"/> read_org | Read org and team membership, read org projects |
| <input type="checkbox"/> admin_public_key | Full control of user public keys |
| <input type="checkbox"/> write_public_key | Write user public keys |
| <input type="checkbox"/> read_public_key | Read user public keys |
| <input checked="" type="checkbox"/> admin_repo_hook | Full control of repository hooks |
| <input checked="" type="checkbox"/> write_repo_hook | Write repository hooks |
| <input checked="" type="checkbox"/> read_repo_hook | Read repository hooks |
| <input type="checkbox"/> admin_org_hook | Full control of organization hooks |
| <input checked="" type="checkbox"/> gist | Create gists |
| <input type="checkbox"/> notifications | Access notifications |

user Update ALL user data
 read:user Read ALL user profile data
 user:email Access user email addresses (read-only)
 user:follow Follow and unfollow users

delete_repo Delete repositories

write_discussion Read and write team discussions
 read_discussion Read team discussions

admin_enterprise Full control of enterprises
 manage_runners_enterprise Manage enterprise runners and runner-groups
 manage_billing_enterprise Read and write enterprise billing data
 read_enterprise Read enterprise profile data

admin_gpg_key Full control of public user GPG keys (Developer Preview)
 write_gpg_key Write public user GPG keys
 read_gpg_key Read public user GPG keys

[Update token](#) [Cancel](#)

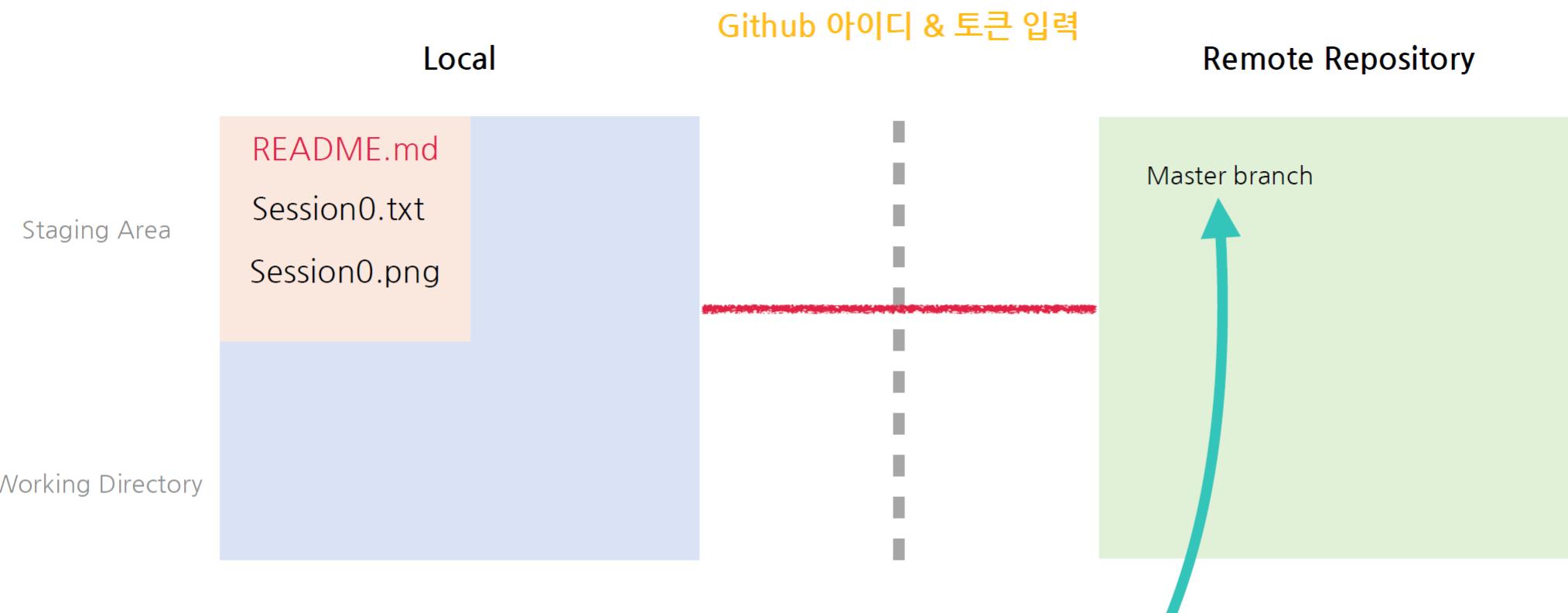
Delete personal access token

Any applications or scripts using this token will no longer be able to access the GitHub API. You cannot undo this action.

[Delete this token](#)

Github에 푸시하기

Github Repository에 코드 올리기



git push origin “branch 이름”

ex) git push origin master

과제 (3월 9일 전까지)

교수님 살려주세요

오늘 실습하면서 만든 자기소개 페이지를

예쁘게 꾸며서

Github 과제용 레포지토리에 푸시하세요!

+ 도메인 투표도 참여해주세요! (추후 안내)