

차세대 챗봇 Lab I

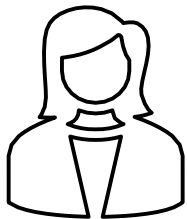
Lecture-1

Sungja Choi,
School of Computing, Gachon University
2023-1



Lecute-1. Introduction to Lecture

차세대 챗봇 Lab I



Information of Lecture

- **Professors** : 최성자 / 송근실 (Team Teaching)
- **Affiliation** : School of Computing, Gachon University
- **Lecture room** : AI-411
- **Phone** : 010-4054-0774 / 010-3665-4065
- **Email** : artchoi0g@gachon.ac.kr / junrub@gachon.ac.kr
- **Webex URI** : <https://gachon.webex.com/meet/artchoi0g>
<https://gachon.webex.com/meet/junrub>

Orientation

KEA (Kakao Enterprise Academy)



- 실무형 SW Engineer 전문인력 육성을 위한 고등교육 플랫폼

교육과정 : 가천 카카오엔터프라이즈 SW아카데미

○ 참여회사 : KaKao Enterprise, DKTechin, ExnTu

교육내용

과정명	교육시간	교육내용
(필수1) 현장미러형 프로젝트	총 336시간 (3과목, 9학점)	◆ 학기 중 기업제공 집중프로젝트 교육 - PBL 방식으로 운영
(필수2) 기업실무 프로젝트	총 320시간	◆ 방학 중 8주간 기업실무 집중프로젝트 교육
(선택) 전공보충 교육과정	총 300시간 (3과목, 9학점)	◆ 학기 중 AI•SW학부 전공생 외 대상 보충교육과정 - 7개 지정 과목 중 3 과목 이수

Orientation

KEA (Kakao Enterprise Academy)

학위수여 : 가천 카카오엔터프라이즈 SW아카데미

- 가천대 재학생 : 가천대 소속 학과 학사 학위 + 마이크로디그리(MicroDegree)
- 가천대 및 타대학 졸업생 : 가천대 마이크로디그리(MicroDegree)

교육생 특전

- 교육지원비(선발 후, 공지) 지급
- 경진대회 참여 지원
- 우수 교육생 시상
- 개인별 진로상담 및 취업지원
- 채용연계형 인턴십 기회 제공

Orientation

차세대 챗봇 Lab 1



- 현장미러형 프로젝트 교육과정
- 소프트웨어 개발 프로세서에 기반한 엔지니어링 능력을 배양
- 프론트엔드 / 백엔드의 최신 기술 습득

- 본 강좌는 현장 미러형 프로젝트 과목으로 기업이 제시하는 프로젝트 중심(PBL)의 수업입니다.
- 산업현장에서 즉각 성과 도출이 가능한 SW 엔지니어의 '지속적, 현실적 양산 모델' 창조하도록 합니다.
- 이를 위해, 응용 시스템 개발 프로세스 학습 및 개발 프로세스에 기반한 응용 개발 실습을 진행합니다.
- 실생활에서 문제를 컴퓨터가 어떻게 해결하는지 그 방식을 이론을 통해 학습하고, 실습을 통해 문제해결 능력을 키우고자 한다.
- 프로젝트 집중교육(PBL)을 통해 당장 현업에 투입할 수 있는 우수한 인력을 양성하여 채용을 위해 실무 능력이 요구되는 전문교육과정으로 현장에 특화된 인력 양성을 목표로 합니다.

Orientation

Course Goal



- 응용 시스템 개발 프로세스 학습 및 개발 프로세스에 기반한 응용 개발 실습
- React.js(또는 Vue.js) 프론트엔드, Spring Boot 백엔드, MySQL을 이용한 설문 시스템 개발
- 클라우드 네이티브 환경의 웹 서비스 개발 (**설문 서비스 / 챗봇 기반의 설문 서비스**)

다음에 기술한 Feature들을 수행하는 응용 시스템을 개발하고 Web을 통해서 서비스한다.

☞ 요구 사항

- [설문 관리자] 설문 관리

설문 생성, 설문 제작; 다양한 유형(찬부식, 선다식, 서열식 등)으로 구성된 설문 작성, 설문 수정/복사 등 관리기능,
설문 발송 대상자 관리, 설문 발송(Email, SMS 등)- [설문관리자] 설문 결과 분석

- [피설문자] 설문 응답

※ 상용 솔루션

구글 Form; https://www.google.com/intl/ko_kr/forms/about/

카카오워크 설문; <https://survey.kakaowork.com/participated>

Orientation

How the class works

PBL(Project Based Learning) 방식

- 과목 운영(가천대 교수진 진행), 수업시간 정규 강의 및 특강(가천대 교수진 진행), 멘토링(카카오엔터프라이즈 실무진 진행),
 - 프로젝트 결과 발표 및 평가(카카오엔터프라이즈 진행)
 - 팀 구성 (5인 1조) / 토론 및 퀴즈 진행
- ※ 기초 및 기본 과목을 이수하지 않은 교육생들을 대상으로(부/복수 전공 학생 등) 개인 맞춤형 보충 교육 진행(AI 소프트웨어학부 교수진 진행 예정)

Orientation

Assessment

- 프로젝트 결과
<SW 개발 프로세스>의 각 단계마다, 교육생들이 '문서화'된 결과 자료 제출 및 발표
☞ 카카오프라이즈와 가천대 교수진이 함께 평가 (60~70 %)
- 그 외 출석, 주차별 과제(수업시간 중 안내 및 진행), 시험(퀴즈 형태로 진행 가능)
☞ 가천대 교수진이 평가 (30 %)
- 필기 유형의 기말평가
- 액티브 러닝

※ 교과과정에 대한 제반사항은 운영상 변경될 수 있음을 공지합니다.

Orientation

Detailed Assessment

- 출 석 : 20%

결석 4회 이상인 경우 F로 처리

- 과 제 : 30%

SW개발프로세스 각 단계의 도출 문서 및 개별과제, 멘토링 점수 반영

- 기말 평가 : 20%

필기시험

- 최종프로젝트 : 30%

최종 프로젝트 보고서 및 발표, 멘토링 점수 반영

※ 교과과정에 대한 제반사항은 카카오 측의 요청 및 운영상의 이슈로 변경될 수 있음을
공지합니다.

Orientation

Weekly Schedule

Week	Contents	Note
1	교과목 소개 및 수업 진행 방법 안내(오리엔테이션) · 설문 시스템 개발 프로젝트 개요 소개 · 사전 코딩 테스트 및 설문조사	도구 및 개발 환경 설정
2	프론트엔드 · 리액트 기본 구조 및 컴포넌트	
3	프론트엔드 · 리액트 Hooks & 상태관리(Recoil)	요구사항 분석
4	프론트엔드 · 리액트 HTTP 통신(Axios), Router Dom, Practice example	구조설계 Active Learning (Maven vs Gradle)
5	백엔드 · Spring Boot를 이용한 RESTFUL API	상세설계 멘토링(1)

Orientation

Weekly Schedule

Week	Contents	Note
6	백엔드 · Spring Boot를 이용한 챗봇-1	
7	백엔드 · Spring Boot를 이용한 챗봇-2	
8	📁 코딩테스트 및 프로토타입	설문 및 챗봇 서비스 멘토링(2) 테스트 점수 미달 학생 재 코딩테스트
9	백엔드 · MSA 설계	Active Learning (Docker vs Kubernetes)
10	백엔드 · MSA와 데이터 관리	최종 프로젝트 요구사항 분석

Orientation

Weekly Schedule

Week	Contents	Note
11	백엔드 · 로드밸런스 및 게이트웨이	구조 및 상세설계
12	백엔드 · 통합 및 테스트	통합 및 테스트
13	백엔드 · 프로젝트 배포 및 운용	배포 및 형상관리
14	프로젝트 완성	멘토링(3)
15	최종 프로젝트 결과 발표 및 평가(카카오엔터프라이즈) 교과목 성적 평가(가천대 교수진)	

Orientation



What do you gain from that process?

- ① 최신 기술 학습 및 학점 수여
- ② 포트폴리오가 도출됩니다. (기본, 심화)
- ③ 실무에서 사용하는 팁 및 개발기간 단축을 위한 훈련
- ④ 3년~5년의 신입사원의 기간 단축
- ⑤ 고액 연봉을 받을 수 있도록 준비

“The Cloude Engineer of Excellent”

Intro

본 학습에 들어가기 전 웹프로그래밍의 기초학습을 진행합니다.

HTML

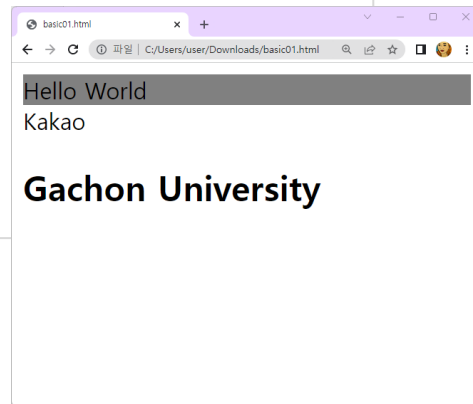


“자~ 이제 웹
브라우저의
개발자도구를
활용해서 디버깅
과정을 살펴봅시다.”

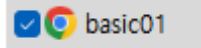
HTML 페이지 작성

- 노트패드를 열어서 아래 코드를 작성하고 작업 폴더에 basic01.html로 저장

```
1 <html>
2 <head>
3 </head>
4 <body>
5 <div id='root'>
6 <div class ="container" style="background-color:gray;
  align-item:center;" >
7   Hello World </div>
8   <div class ="container"> Kakao </div>
9   <h2> Gachon University </h2>
10 </div>
11 <script type="text/javascript" />
12 </body>
13 </html>
```



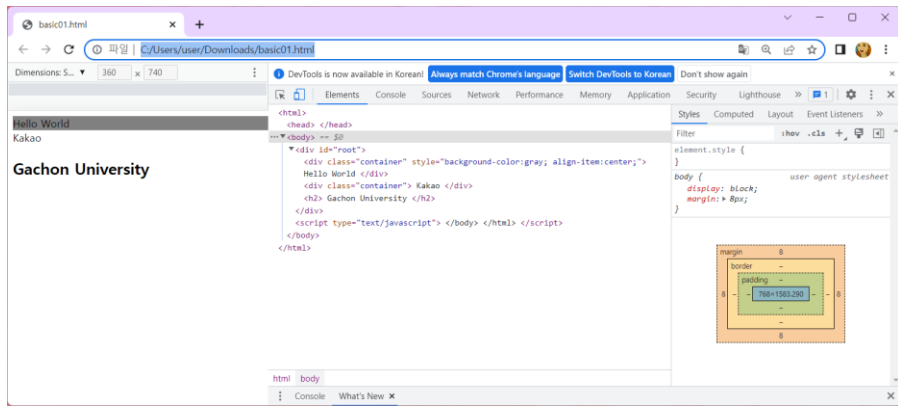
- 오른쪽 마우스 – 열기 (혹은 연결프로그램 Chrome)



Debugging

크롬 개발자 도구 (F12)

- 웹페이지 디버깅 과정을 위해 크롬의 개발자 도구 사용
- 단축키(F12)를 사용해서 개발자 도구 불러옴
- 소스 탭에서 라인 번호 사용
- Add breakpoint 클릭하면 파란색 박스가 표시
- 결과 브라우저에서 단축키(F5) 화면 리플레쉬
- 소스의 변수 위치에서 오른쪽 마우스 Add to watch
- 오른쪽 화면의 Watch에 변수 등록되어 실시간 확인

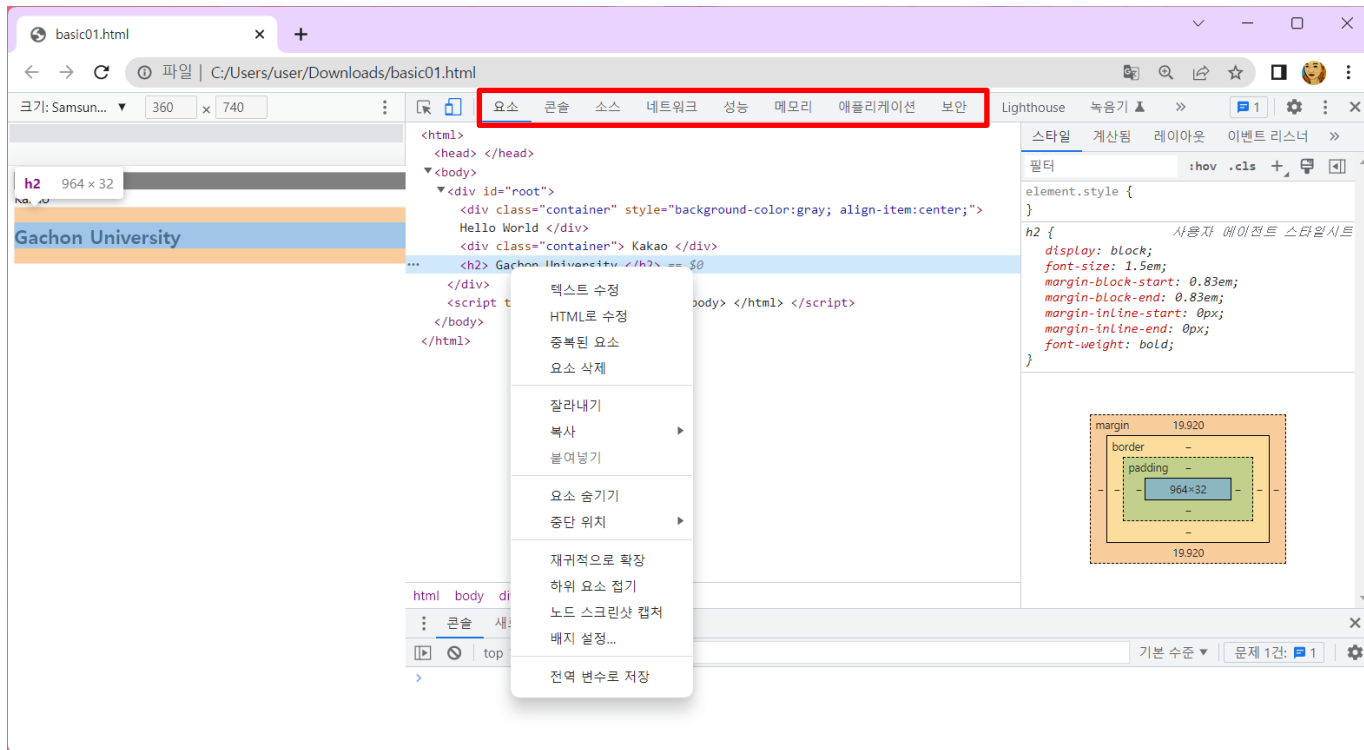


Debugging

- 탭 설명
- 텍스트수정
- 중복된 요소
- 스타일 변경
- HTML로 수정

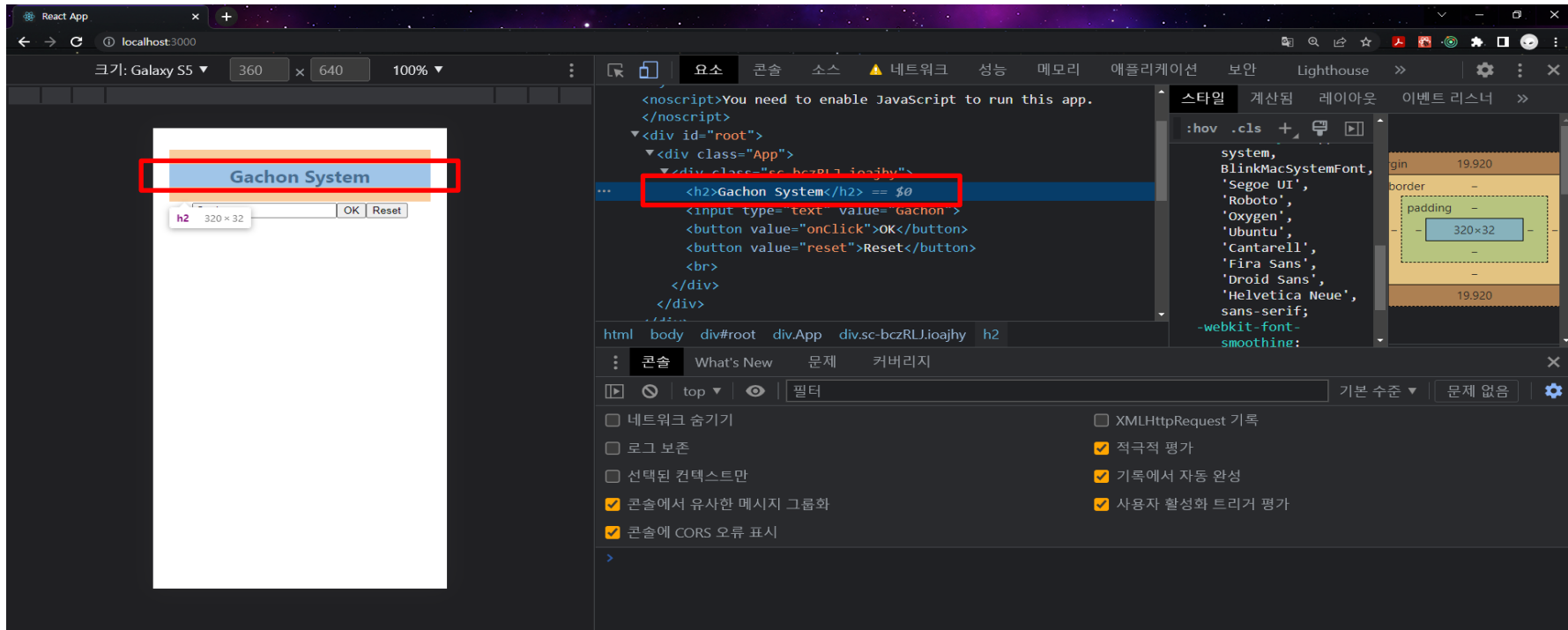
* F/키는 환경설정

크롬 개발자 도구 (F12)



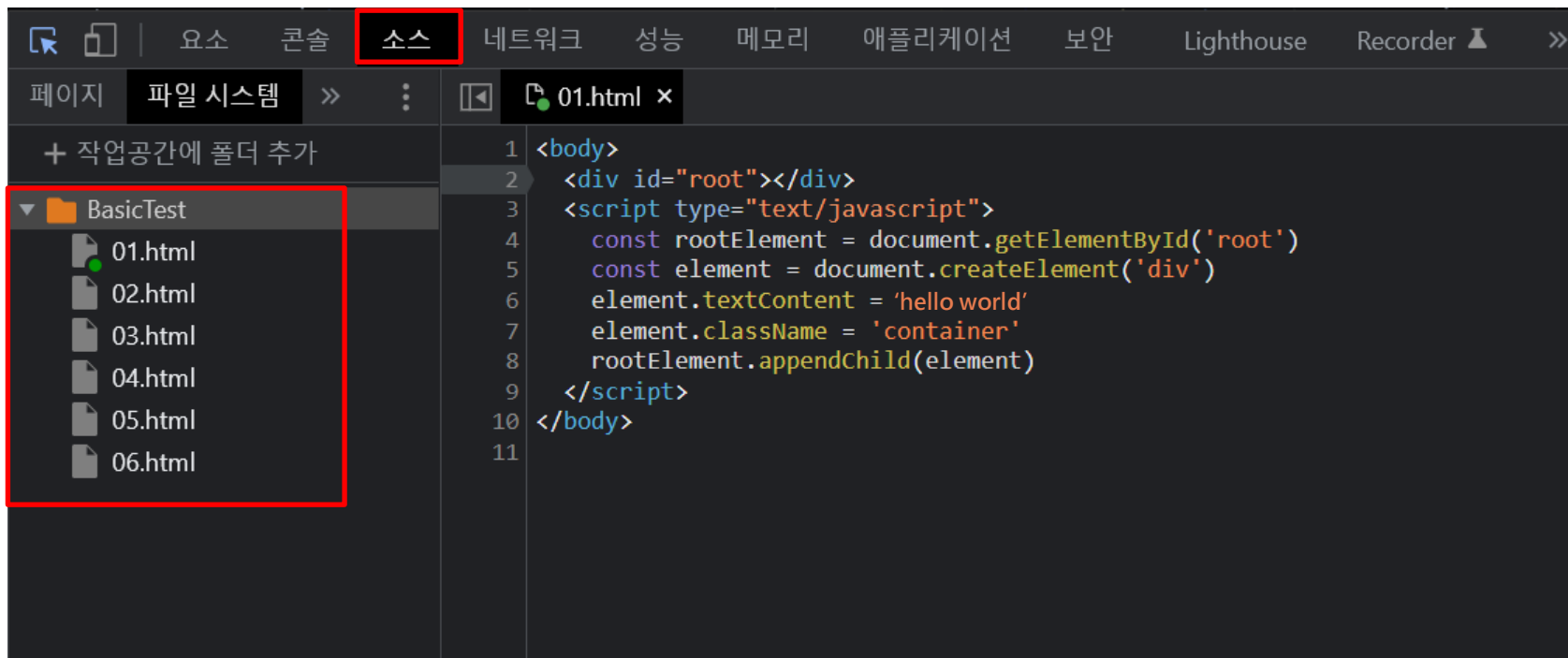
Debugging

요소디버그



Debugging

소스수정



Debugging

콘솔 활용

```
> console.log("hello");  
hello  
◀ undefined  
> function hello(){  
  console.log("fn hello");  
}  
◀ undefined  
> hello();  
fn hello  
◀ undefined  
> for (let i = 0; i < 5; i++) {  
  console.log("숫자", i);  
}  
숫자 0  
숫자 1  
숫자 2  
숫자 3  
숫자 4
```

```
> prompt("Your University?");  
◀ 'Gachon'
```

localhost:3000 내용:

Your University?

확인

취소

```
> console.log(document);  
▼ #document  
  <!DOCTYPE html>  
  <html lang="en">  
    ▶ <head>...</head>  
    ▶ <body>...</body>  
  </html>
```

* 콘솔창을 활용해서 자바스크립트
적용 및 코드 결과 테스트

개발환경 및 IDE 설치

(프론트 엔드 / 백 엔드 개발환경 구축 및 빌드, 배포를 진행)

Node.js

“프론트 엔드의
자바스크립트
서버”

Installation

🖱️ <https://nodejs.org/en/download/>

Downloads

Latest LTS Version: 18.14.1 (includes npm 9.3.1)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS Recommended For Most Users	Current Latest Features	
 Windows Installer node-v18.14.1-x64.msi	 macOS Installer node-v18.14.1.pkg	 Source Code node-v18.14.1.tar.gz
Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit / ARM64	
macOS Binary (.tar.gz)	64-bit	ARM64
Linux Binaries (x64)	64-bit	
Linux Binaries (ARM)	ARMv7	ARMv8
Source Code	node-v18.14.1.tar.gz	



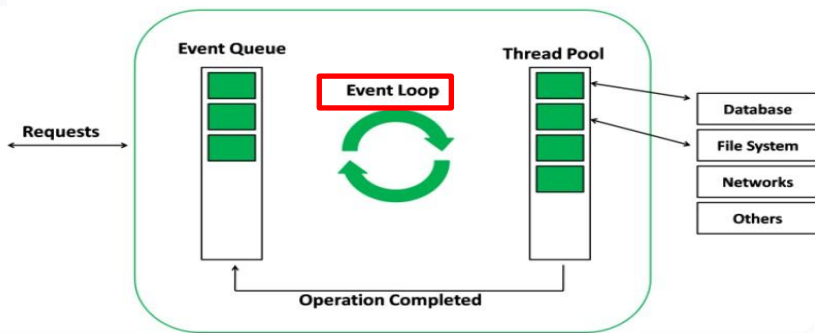
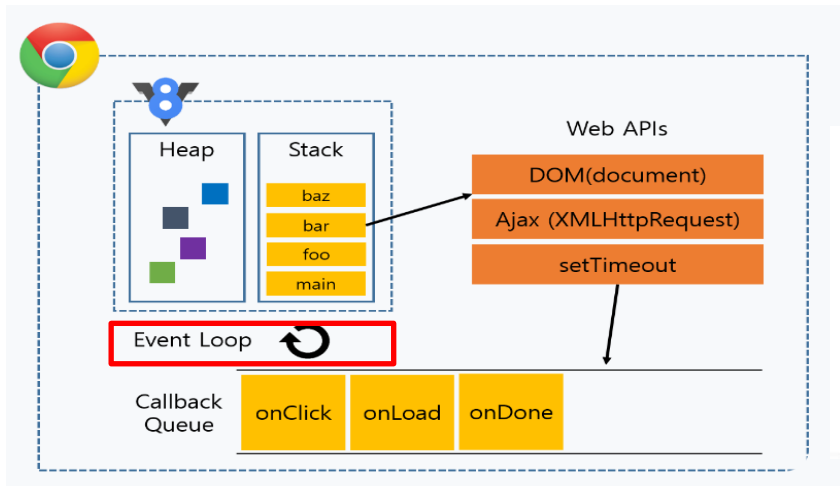
```
C:\Users\user>node -v
v18.14.1

C:\Users\user>npm -v
9.3.1
```

- 환경에 맞는 버전을 다운로드 후 설치를 진행
- Node.js 설치 후 콘솔창에서 버전을 확인합니다.
- Npm 자동 설치 됩니다.
- Npm은 노드 패키지 매니저로 프로젝트에서 요구되는 다양한 패키지를 내장하고 있습니다.

JavaScript Engine

☞ 구글 크롬의 자바스크립트 엔진은 V8의 비동기처리 과정입니다.
해당 엔진이 Node.js 서버모듈과 통신하여 웹서비스를 제공합니다.



- (1) Web API DOM 객체, Ajax 요소, Timer 등의 객체를 스택에 등록하고 (2) 이벤트 콜백 함수를 등록합니다.
- (3) 요청이 들어오면 onClick, onLoad, onDone 등의 (4) 이벤트가 발생하고 이벤트 루프가 작동합니다.
- (5) 이벤트 루프에서는 쓰레드 풀에서 이벤트가 발생하면 이를 이벤트 큐에 순서대로 넣고 그 이벤트를 처리합니다.

VSCode IDE


“코드 작성과 빌드,
테스트, 배포를 위한
개발도구”


Installation


🔗 <https://nodejs.org/en/download/>

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.


↓ Windows
Windows 8, 10, 11


↓ .deb
Debian, Ubuntu
↓ .rpm
Red Hat, Fedora, SUSE


↓ Mac
macOS 10.11+

User Installer 64 bit 32 bit ARM
System Installer 64 bit 32 bit ARM
.zip 64 bit 32 bit ARM

.deb 64 bit ARM ARM 64
.rpm 64 bit ARM ARM 64
.tar.gz 64 bit ARM ARM 64
Snap Store

.zip Universal Intel Chip Apple Silicon

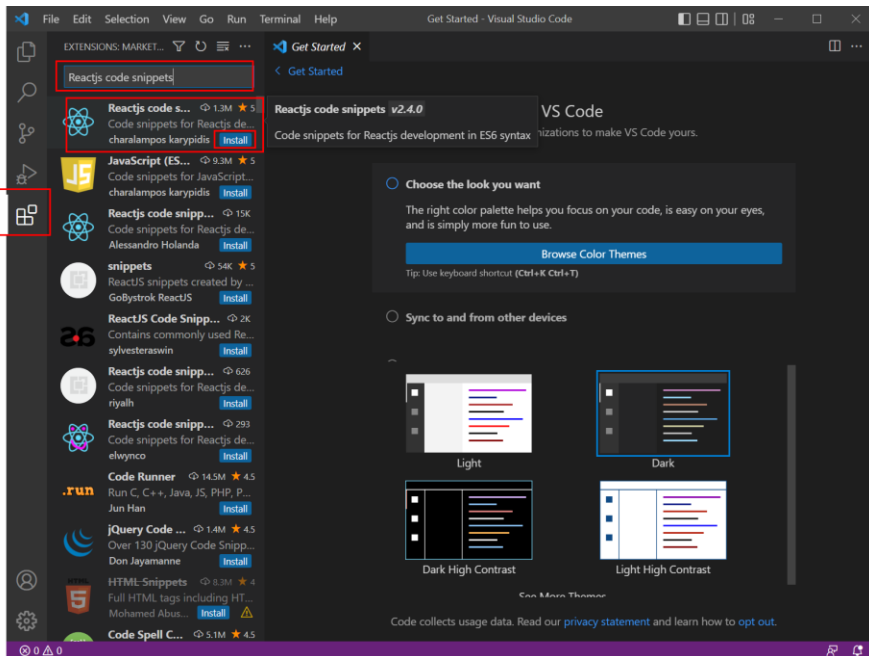


Using Extensions

Install Reactjs code snippets



버튼을 클릭하면 VSCode에 패키지를 설치할 수 있습니다.
Reactjs code snippets는 자동으로 코드를 완성해 주는 도구입니다.



Practice 1

프론트 엔드의 리액트 개발환경을 구축하고
프로젝트를 만들어 도커 이미지를 배포해봅니다.

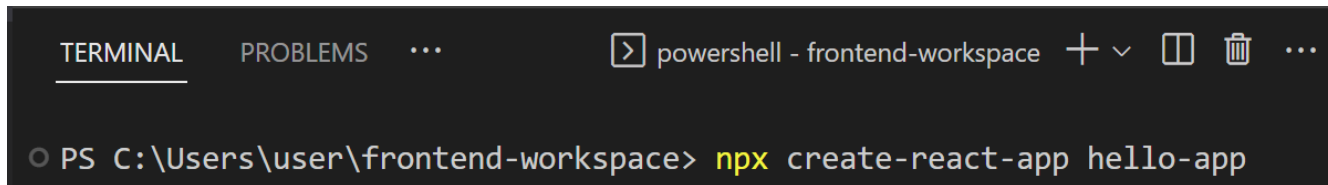
Practice1

hello-app

Create Project Env.

👉 프로젝트 폴더 생성 및 터미널 활용

- FrontEnd 폴더를 생성한 후
- VSCode의 메뉴에서 파일 - 폴더열기를 하여 VSCode 창 오픈
- 하단의 터미널 창 활용 혹은 Ctrl+Shift+` (역따옴표) 혹은 메뉴에서 [터미널 - 새 터미널]로 새 터미널 창 생성
- `npx create-react-app hello-app` 의 명령어를 터미널에 입력



The screenshot shows a VS Code terminal window with a dark theme. The title bar at the top reads "powershell - frontend-workspace". Below the title bar, there are tabs for "TERMINAL" and "PROBLEMS". The terminal content shows a PowerShell prompt "PS C:\Users\user\frontend-workspace>" followed by the command "npx create-react-app hello-app". The "npx" part of the command is highlighted in yellow.

```
TERMINAL PROBLEMS ... powershell - frontend-workspace + - [X] ...  
PS C:\Users\user\frontend-workspace> npx create-react-app hello-app
```

Practice1

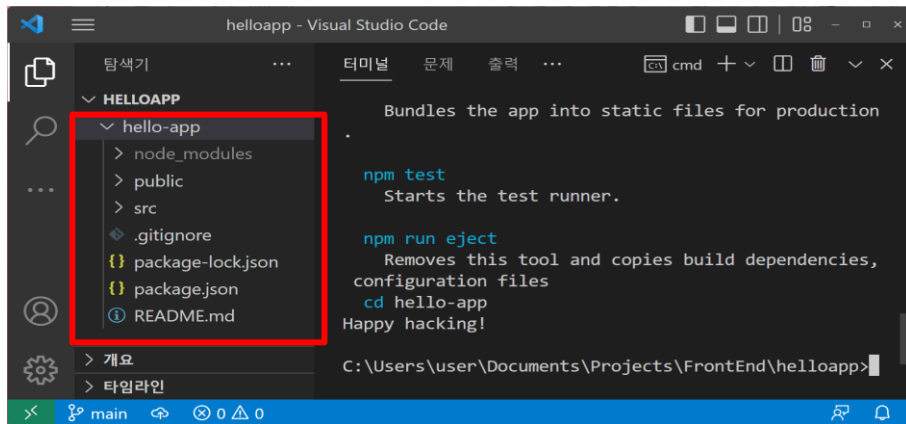
hello-app

Create Project Env.

☞ 프로젝트 폴더 생성 및 터미널 활용

프로젝트 폴더와 파일들이 생성

- **node_modules** 는 자바스크립트 서버관련 모듈을 모아 놓은 폴더
- **public** 폴더는 정적 페이지 작업을 위한 것
- **package.json** 파일은 라이브러리 및 버전, 빌드 등에 대한 정보를 관리



Practice1

hello-app

Code

☞ VSCode의 src 폴더에 있는 App.js의 내용을 수정

```
import React from "react"

function App() {
  return (
    <div>
      <h2> Gachon, Helloworld </h2>
    </div>
  );
}

export default App;
```

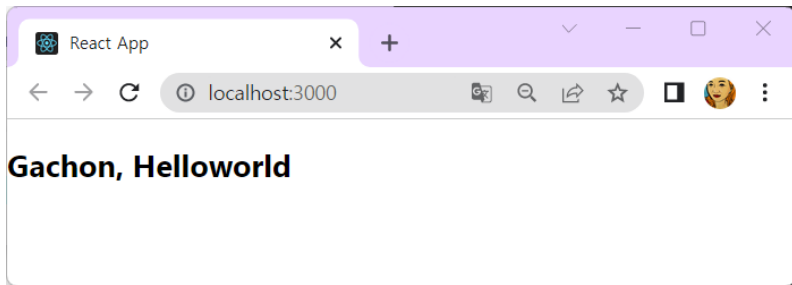
Practice1

hello-app

Execution

🖱️ Http://localhost:3000/

- 터미널 창에서 명령어를 실행
cd hello-app
npm start



- App.js에서 return 메소드 내의 div 영역에 쓰여진 "Gachon, Helloworld" 문구가 h2 태그가 반영되고, 화면에 렌더링 되어 보여집니다.
- html 요소(element), 또는 React 요소 등의 코드가 눈으로 볼 수 있도록 그려지는 것을 렌더링(rendering) 이라고 말합니다.

[REF.] Git에 배포하기

Git 설치

<https://git-scm.com/download/win>

Download for Windows

Click here to download the latest (2.35.3) 64-bit version of Git for Windows. This is the most recent maintained build. It was released 4 days ago, on 2022-04-15.

Other Git for Windows downloads

Standalone Installer

32-bit Git for Windows Setup.

64-bit Git for Windows Setup.



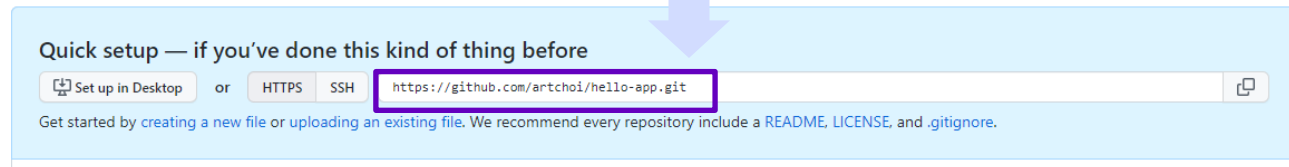
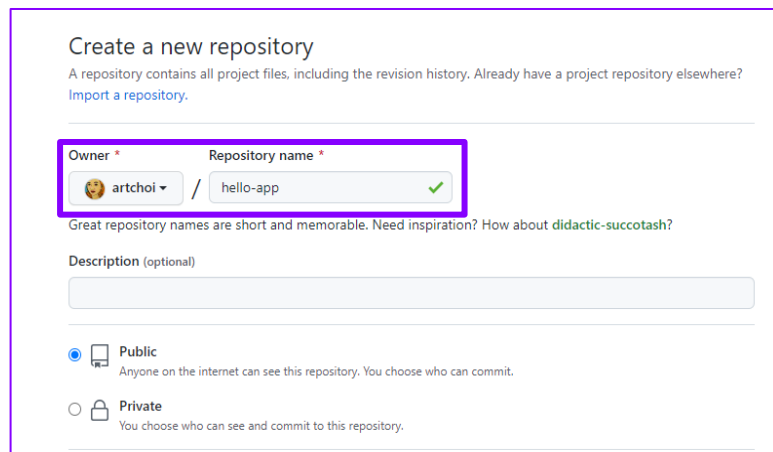
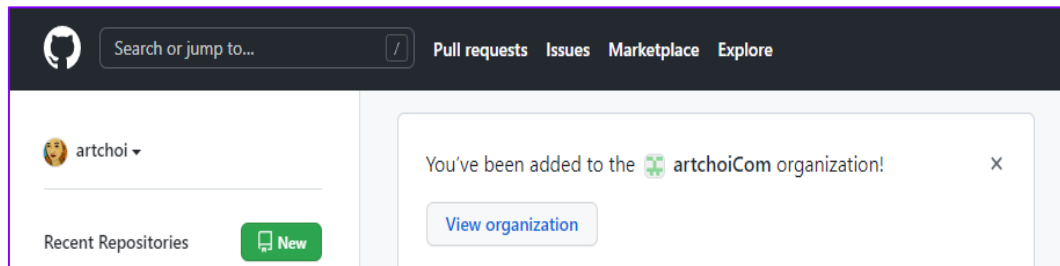
github 회원가입

<https://github.com/login>

[REF.] Git에 배포하기

Git Repository 생성

- Github.com 에 **내 계정으로 접속** 후 New Repositories를 생성



[REF.] Git에 배포하기

Git Repository 생성

- GitHub에 프로젝트에 대한 Repository가 만들어지면 수행할 명령라인 제공되므로 참조

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

 Set up in Desktop or **HTTPS** **SSH** `https://github.com/artchoi/gittest.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 "# gittest" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/artchoi/gittest.git
git push -u origin main
```



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

```
git remote add origin https://github.com/artchoi/gittest.git
git branch -M main
git push -u origin main
```

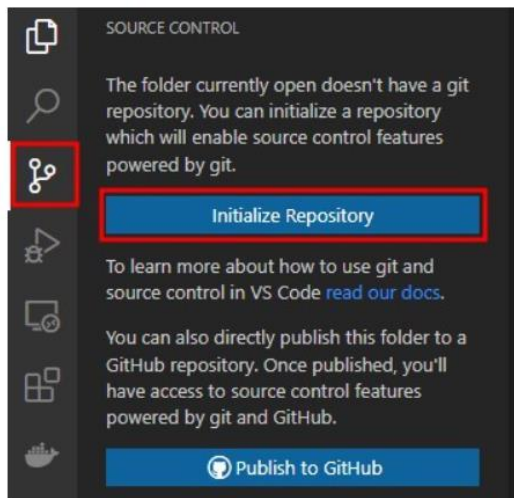


[REF.] Git에 배포하기

초기작업 이후에는
Ctrl+Alt+'p'
작업표시줄에서 Git:
Initialize Repository
활용

VSCode Git 연동

- 좌측 사이드 메뉴의 Initialize Repository
부가적인 Extension 설치 (GitLens, Git History)



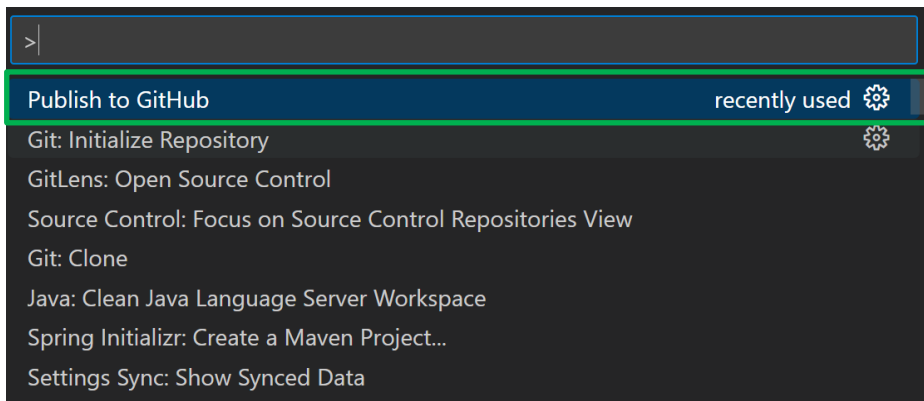
* Git init 효과

[REF.] Git에 배포하기

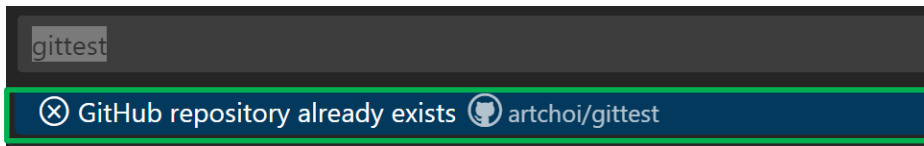
VSCode github 연동

👉 등록된 Git 레포지터리로 소스파일을 배포합니다.

1. 레포지터리가 등록이 되었다면 Ctrl+Shift='p' 단축키로 작업표시줄을 엽니다.

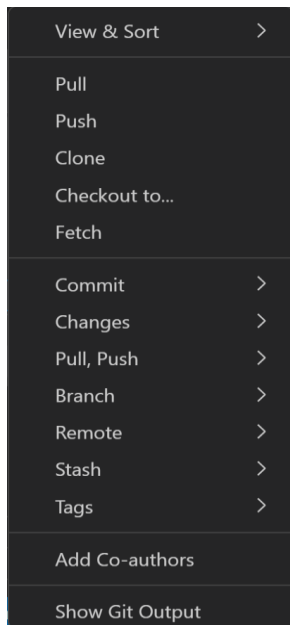


2. Publish to GitHub 를 선택

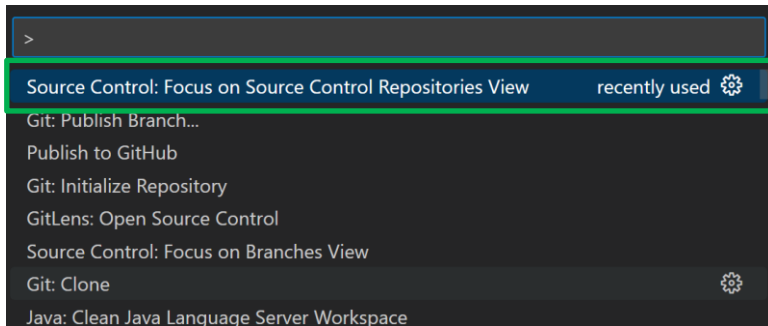


[REF.] Git에 배포하기

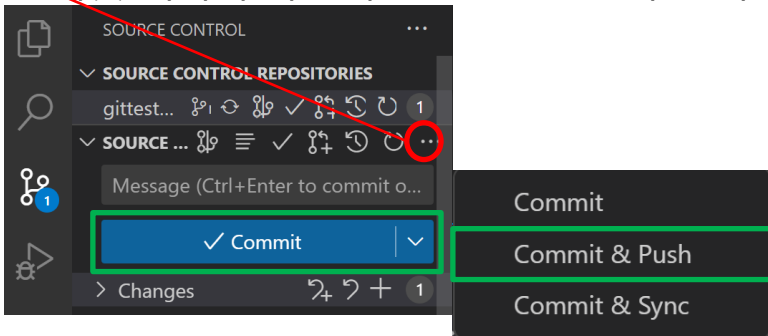
VSCode github 연동



3. Ctrl+Shift='p' 단축키로 작업표시줄을 열고, Source Control: Focus on ... 선택



4. 커밋 메시지 작성 후 Ctrl+Enter를 하면 메시지가 적용되고 Commit & Push



[REF.] Git에 배포하기

VSCode github 연동

☞ Git Command 배포과정

git init	❶ git 초기화
git add .	❷ Stage를 만듦
git commit -m "first commit"	❸ first commit 메시지를 가진 버전을 생성
git branch -M main	❹ main 브랜치에 있음
git remote add origin https://github.com/000/000.git	❺ 내계정의 github 저장소 추가
git push -u origin main	❻ main 브랜치에 소스 업로드(배포)

- 스테이지란? 깃에게 버전 만들 준비를 하라고 알려주는 것
- 커밋이란? 버전생성
- 브랜치란? 여러개발자들이 다양한 작업을 할 수 있도록 제공하는 독립적인 작업공간

“Git으로 협업하고 프로젝트의 버전관리를 원활히 할 수 있도록
팀원들과 충분히 훈련하시기 바랍니다.”

Practice1

hello-app

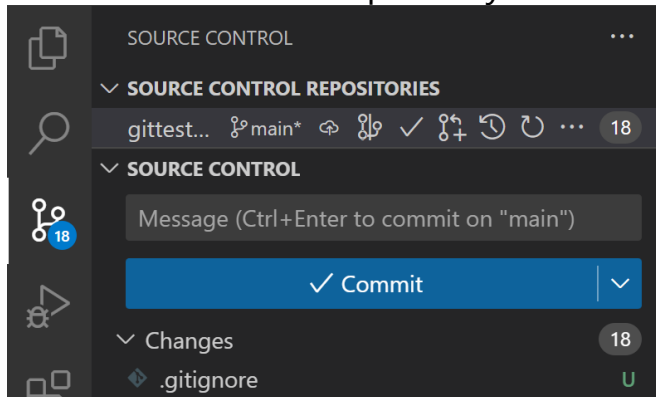
hello-app의 git 배포

👉 소스 배포 및 형상관리를 위해서 git을 이용하고 github 레포지터리(저장소)에 배포하는 과정

1. 레포지터리가 등록이 되었다면 Ctrl+Shift='p' 단축키로 작업표시줄을 엽니다.
2. Publish to GitHub 를 선택

이미 작업이 진행되었다면 별다른 액션이 없음

3. Source Control Repository를 불러와서 **Commit & Push** 실행 (p37 참조)

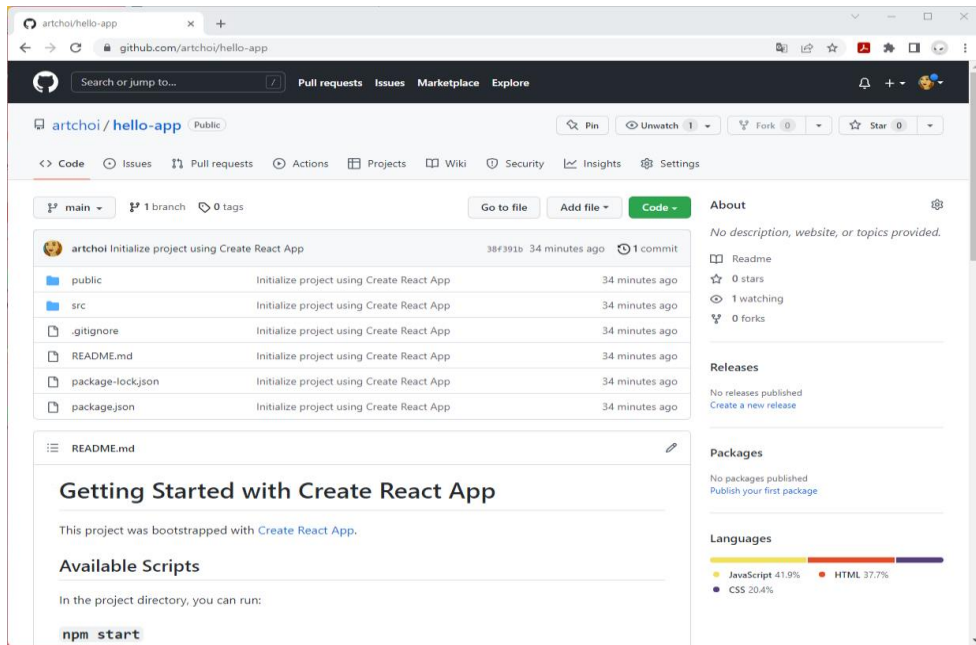


Practice1

hello-app

hello-app의 git 배포

👉 Github의 등록된 레포지터리에서 배포된 결과를 확인합니다



[REF.]

WSL2

Installation

👉 <https://learn.microsoft.com/ko-kr/windows/wsl/install>

Linux용 Windows 하위 시스템을 사용하면 개발자가 기존 가상 머신의 오버헤드 또는 듀얼 부팅 설정 없이 대부분의 명령줄 도구, 유틸리티 및 애플리케이션을 비롯한 GNU/Linux 환경을 수정하지 않고 Windows에서 직접 실행

- ❶ 관리자 권한으로 Windows PowerShell 실행
- ❷ `dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart`
- ❸ `dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart`
- ❹ Reboot
- ❺ `wsl --set-default-version 2`
- ❻ 작업완료 메시지 확인
- ❼ Microsoft store에서 ubuntu 검색 후 Ubuntu 22.04 LTS 다운로드 후 설치

[REF.]

WSL2

Installation

☞ Linux용 Windows 하위 시스템

개발자가 기존 가상 머신의 오버헤드 또는 듀얼 부팅 설정 없이 대부분의 명령줄 도구, 유틸리티 및 애플리케이션을 비롯한 **GNU/Linux** 환경을 수정하지 않고 Windows에서 직접 실행

- ❶ 관리자 권한으로 Windows PowerShell 실행
- ❷ `dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart`
- ❸ `dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart`
- ❹ Reboot
- ❺ `wsl --set-default-version 2`
- ❻ 작업완료 메시지 확인
- ❼ Microsoft store에서 ubuntu 검색 후 Ubuntu 22.04 LTS 다운로드 후 실행
- ❽ 사용자 계정 및 암호 생성
- ❾ `sudo apt update && sudo apt upgrade`
- ❿ `wsl -i -v` 로 확인

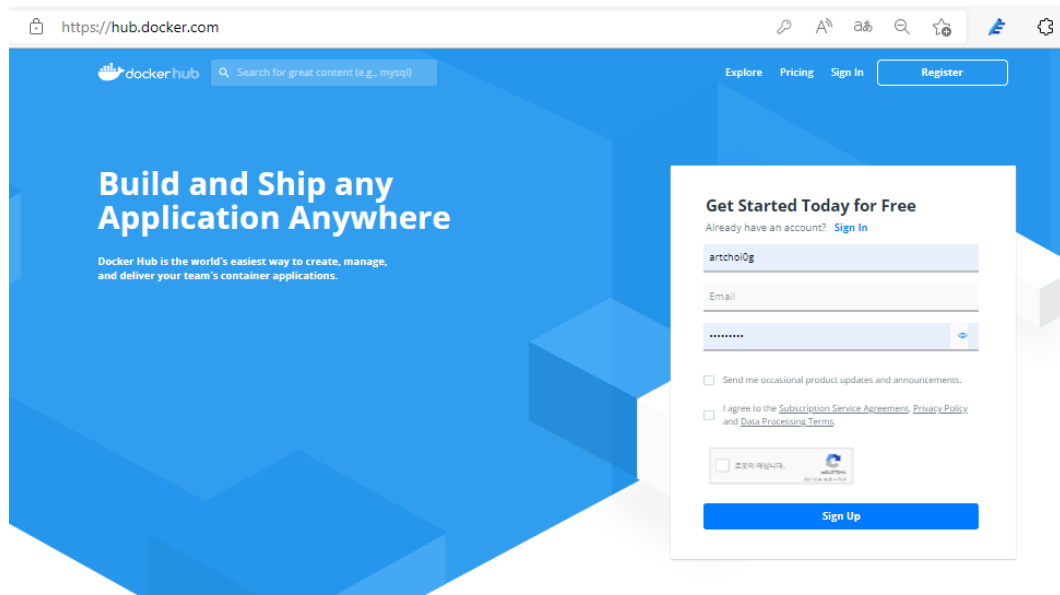
```
PS C:\WINDOWS\system32> wsl -i -v
NAME                STATE              VERSION
* Debian             Running            2
CentOS7             Stopped            2
docker-desktop      Running            2
docker-desktop-data Running            2
```

*Debian과 CentOS등이 윈도우에 설치됨

[REF.] Docker

Installation

👉 <https://hub.docker.com>



*Free 계정 생성 및 이메일 검증을 진행해서 회원가입을 진행합니다.
도커는 실행결과를 이미지로 생성해 주는 프로젝트 배포를 위한 최신
기술입니다.

[REF.] Docker



Installation

👉 Docker Desktop

Docker Desktop

Install Docker Desktop – the fastest way to containerize applications.

Mac with Intel Chip

Mac with Apple Chip

MOST COMMON

Also available for **Windows** and Linux

Mac OS 환경의 도커
테스트탑 설치 참조

<https://docs.docker.com/desktop/install/mac-install/>

- 다운받은 'Docker Desktop Installer' 실행
- Configuration 항목 모두 체크
- Installation succussed 메시지 확인

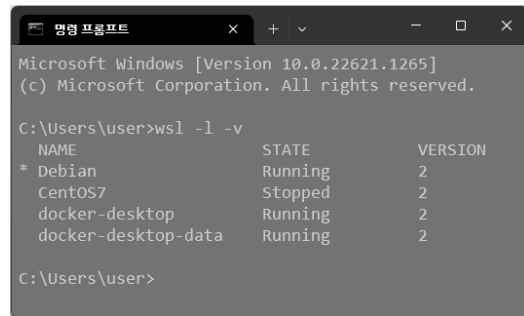
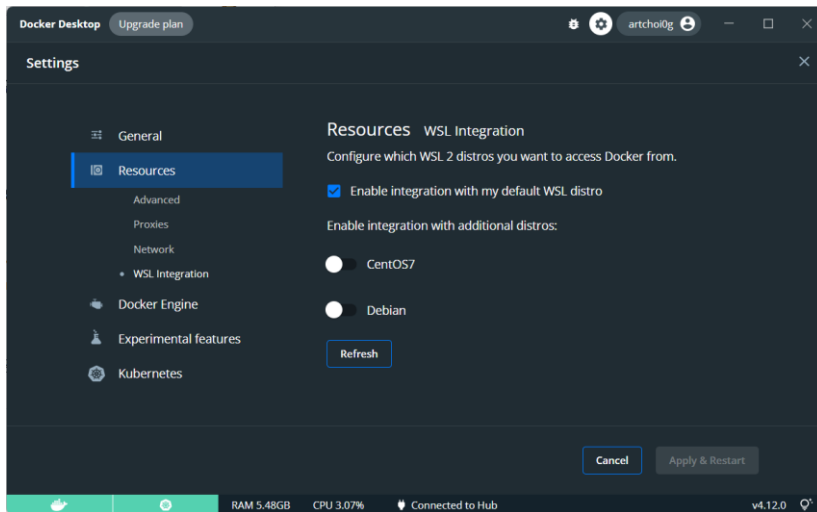
- ✓ WSL2 요구하는 윈도우 컴포넌트 설치
- ✓ 바탕화면 아이콘 추가

[REF.] Docker

WSL2 설정 확인

☞ 도커 설치를 위해서 윈도우용 리눅스 가상환경인 WSL2 설정확인

- Docker Desktop을 실행
- Settings - General - ☒ Use the WSL 2 based engine 항목 체크
- Settings - Resources - WSL Integration 설정확인



[참고] WSL2 관련

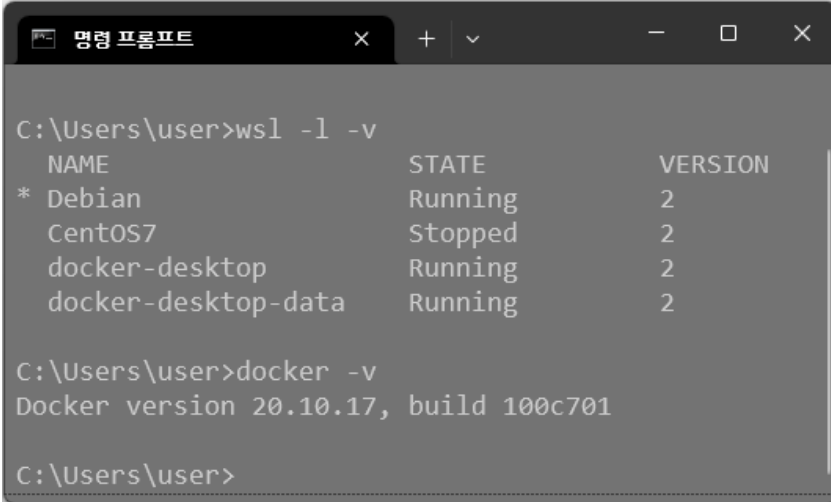
<https://www.lesstif.com/software-architect/wsl-2-windows-subsystem-for-linux-2-89555812.html>

[REF.] Docker

설치확인

☞ 관리자권한으로 콘솔창이나 파워 쉘을 실행하여 확인

```
wsl -l -v  
docker -v
```

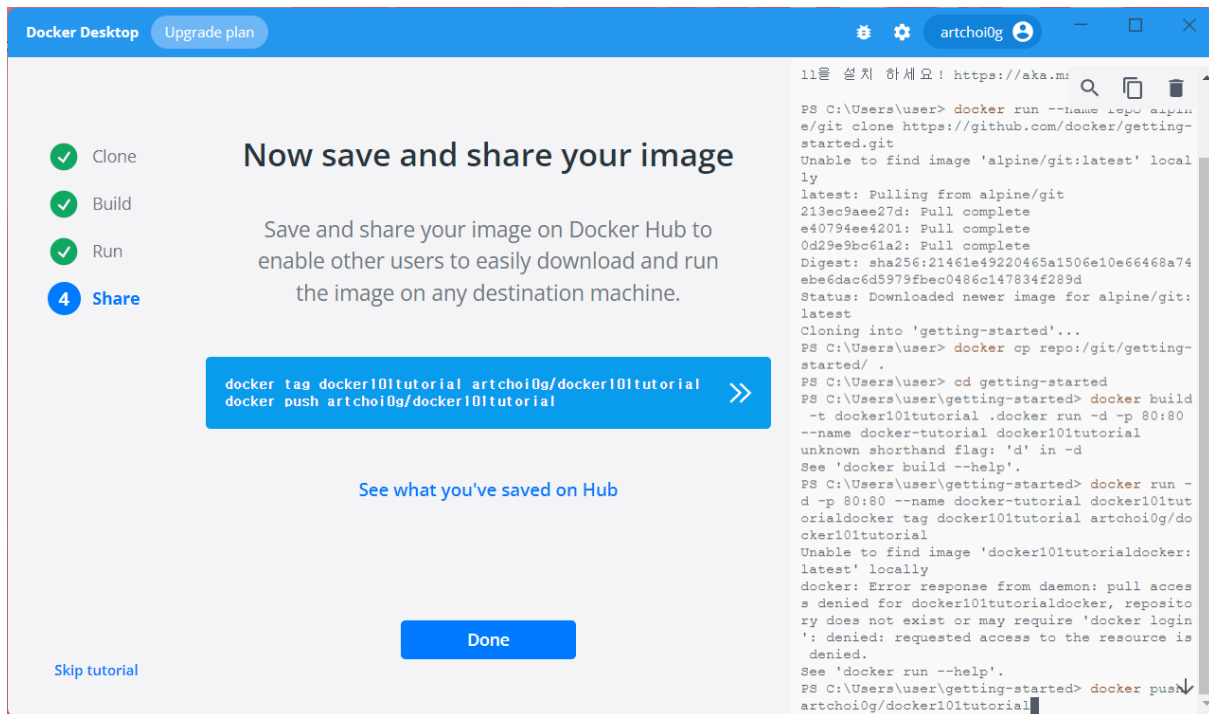


```
C:\Users\user>wsl -l -v  
NAME                STATE      VERSION  
* Debian             Running    2  
CentOS7              Stopped    2  
docker-desktop       Running    2  
docker-desktop-data  Running    2  
  
C:\Users\user>docker -v  
Docker version 20.10.17, build 100c701  
  
C:\Users\user>
```

[REF.] Docker

Docker 실행확인

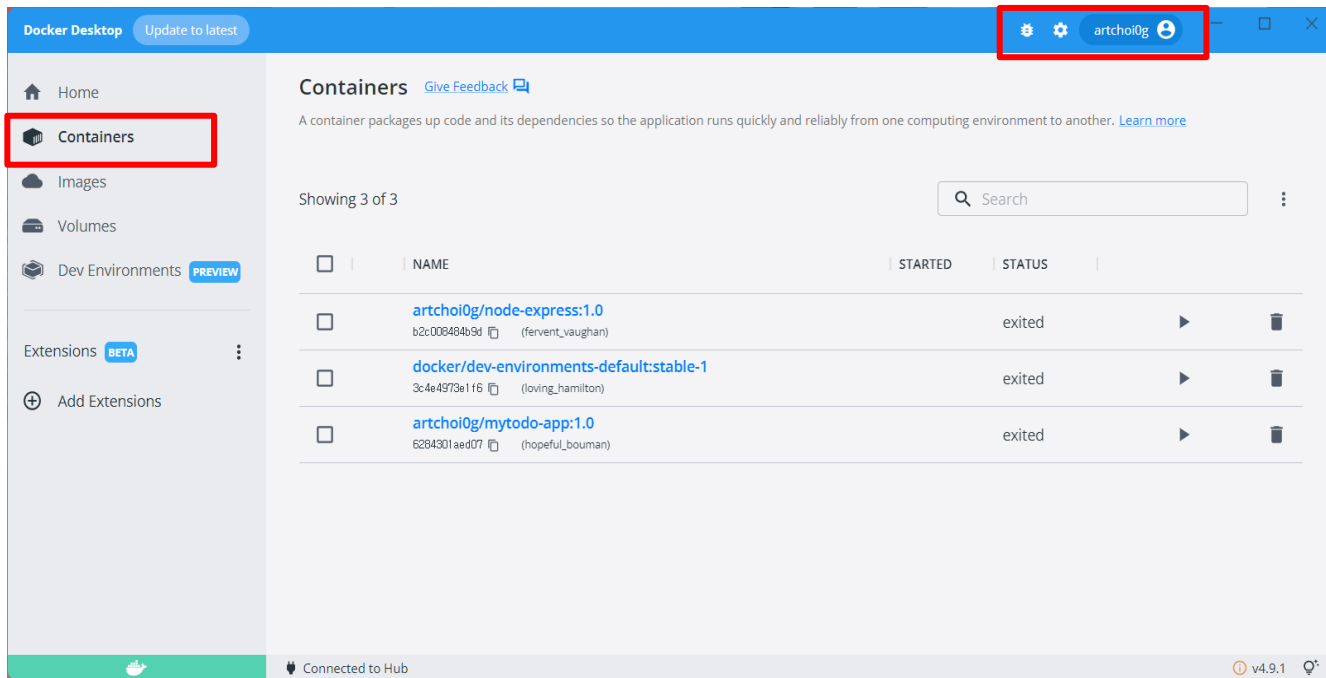
☞ 처음 실행할 때 sample container 제공



[REF.] Docker

Docker 실행확인

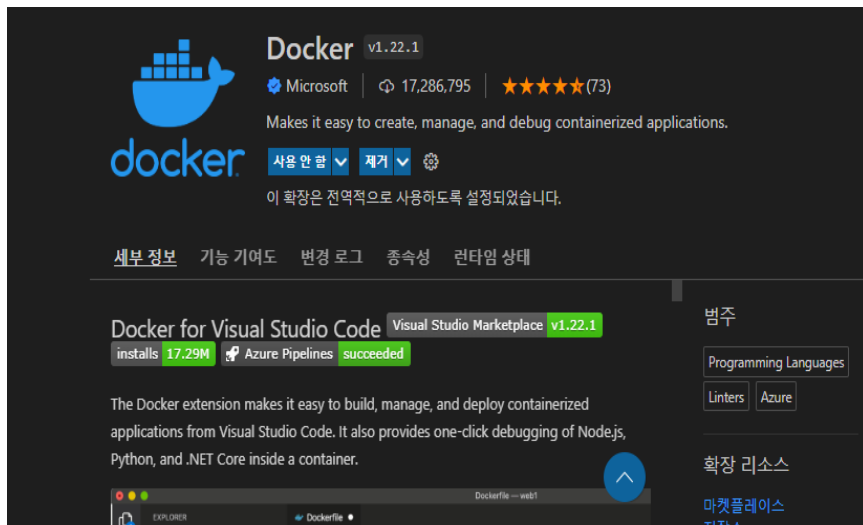
👉 Docker-hub에 로그인 되어 있어야 함



[REF.] Docker

VSCode Docker Extension 설치

☞ Extension을 클릭하고 Docker 설치파일 찾아서 설치

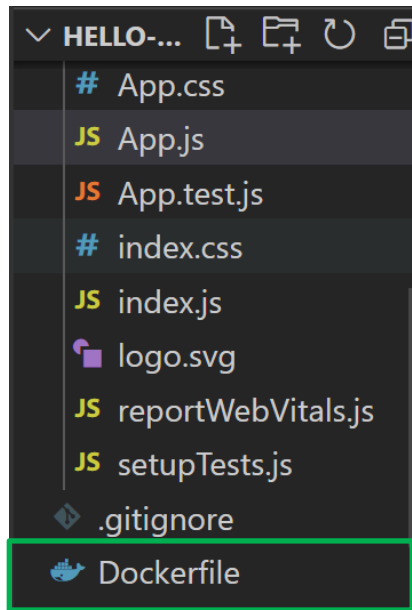


Practice1

hello-app

Docker 배포하기

📁 최상위 폴더에 Dockerfile 생성



```
Dockerfile > ...
1 FROM node:16-alpine
2 WORKDIR /usr/src/app
3 COPY package.json ./
4 RUN npm install
5 COPY ./ ./
6 EXPOSE 3000
7 CMD ["npm", "run", "start"]
8
```

Layering

- ① 노드이미지 v16의 최소단위의 리눅스 버전의 베이스이미지 지정
- ② 도커이미지안에 어떤 경로를 복사할지 지정
- ③ package.json을 현재 폴더에 복사
- ④ npm 설치
- ⑤ 호스트 현재위치를 도커이미지 현재위치에 복사
- ⑥ 포트 3000번 지정 (리액트포트)
- ⑦ npm run start 커맨드 실행

Practice1

hello-app

Docker 빌드 및 이미지 생성

☞ 도커 빌드 명령어를 실행하면 도커 데스크 탑에서 확인가능
docker build -t hello-app .

한 칸 띄우고, 현재위치를 나타냄

```
PS C:\Users\user\frontend-workspace\hello> docker build -t hello-app .  
[+] Building 28.2s (6/10)  
=> => sha256:63b65145d645c1250c391b2d16ebe53b3747c295ca8ba2fc 3.37MB / 3.37MB 1.9s  
=> => sha256:a67f65df360be748c09d983c17f1fd884f32990a71042c 36.73MB / 36.73MB 4.2s
```

Containers

Images

Volumes

Dev Environments BETA

*도커 데스크 탑 왼쪽메뉴의 Images를 클릭하면 리스트에서 해당 이미지 확인

hello-app	latest	be2f06ffca79	less than a minute ago	746.45 MB
-----------	--------	--------------	------------------------	-----------

```
PS C:\Users\user\frontend-workspace\hello-app> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hello-app	latest	33dc65b908bb	2 minutes ago	747MB

Practice1 hello-app

Docker 런 및 컨테이너 배포

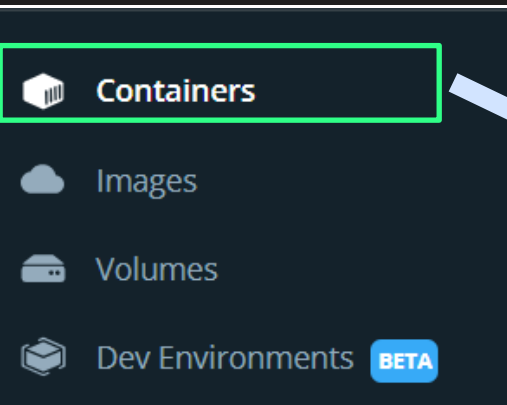
☞ 도커 런 명령어를 실행하면 도커 데스크 탑에서 확인가능
`docker run -it -p 3000:3000 --rm hello-app`

```
PS C:\Users\user\frontend-workspace\hello-app> docker run -it -p 3000:3000 --rm hello-app
```

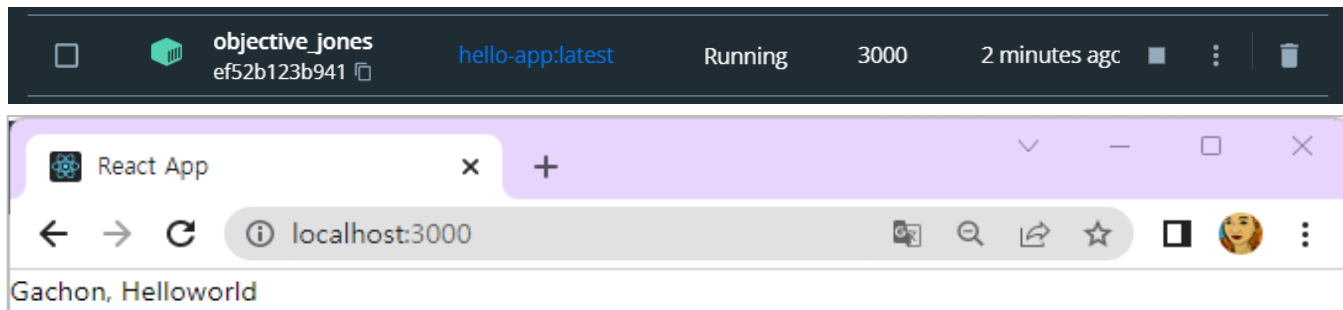
호스트 포트

컨테이너 포트

```
> hello-app@0.1.0 start  
> react-scripts start
```



*도커 데스크 탑 왼쪽메뉴의 Containers를 클릭하면 리스트에서 hello-app 이미지가 컨테이너로 로드 된 것을 확인가능하고, 포트는 3000번



Practice1 hello-app

Docker-hub Repository 생성

👉 도커 허브에 접속해서 Create repository에 앱 이름(hello-app)으로 저장소 생성

React App x Docker Hub x +

hub.docker.com/repository/create?namespace=artchoi0g

Wasm is a fast, light alternative to Linux containers – try it out today with the [Docker+Wasm Beta](#).

docker hub Search Docker Hub Explore Repositories Organizations Help Upgrade artchoi0g

Repositories Create Using 0 of 1 private repositories. [Get more](#)

Create repository

artchoi0g hello-app

Description

Pro tip
You can push a new image to this repository using the CLI

```
docker tag local-image:tagname new-repo:tagname  
docker push new-repo:tagname
```

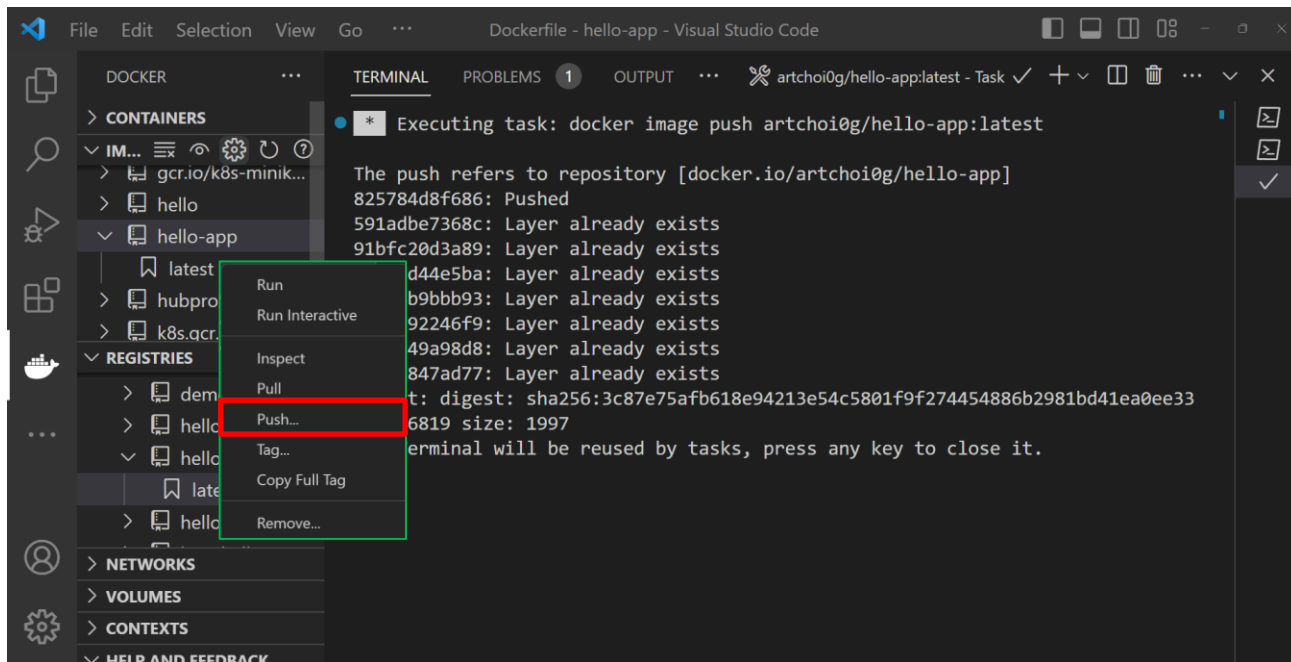
Make sure to change *tagname* with your desired image repository tag.

Practice1

hello-app

Docker 런 및 컨테이너 배포

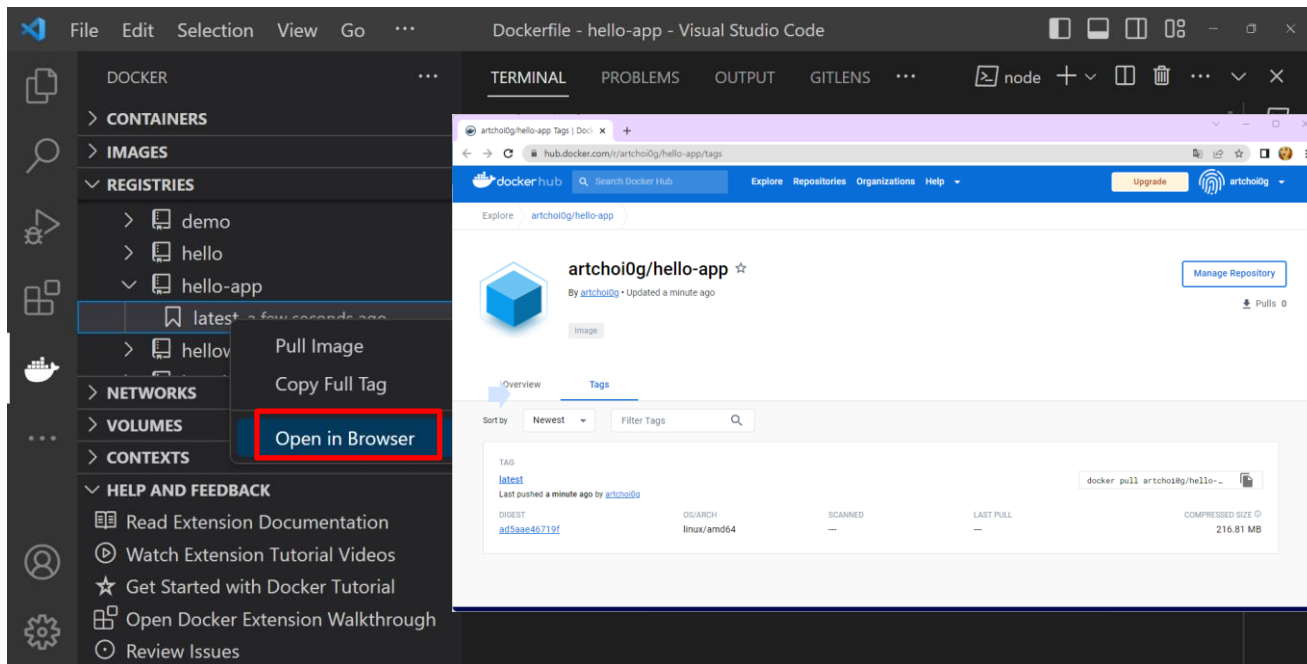
☞ VSCode Docker 메뉴를 활용해서 RUN & PUSH



Practice1 hello-app

Docker-hub 실행

☞ VSCode Docker 메뉴의 Open in Browser를 활용해서 Docker-hub 접근하고 hello-app 이미지 배포확인



Practice2

백엔드의 스프링 부트 개발환경을 구축하고
빌드 도구를 활용해서 웹서비스를 제공합니다.

Practice2 SpringBoot



start.spring.io

☞ start.spring.io에 접근해서 스프링 부트 프로젝트를 쉽게 작성

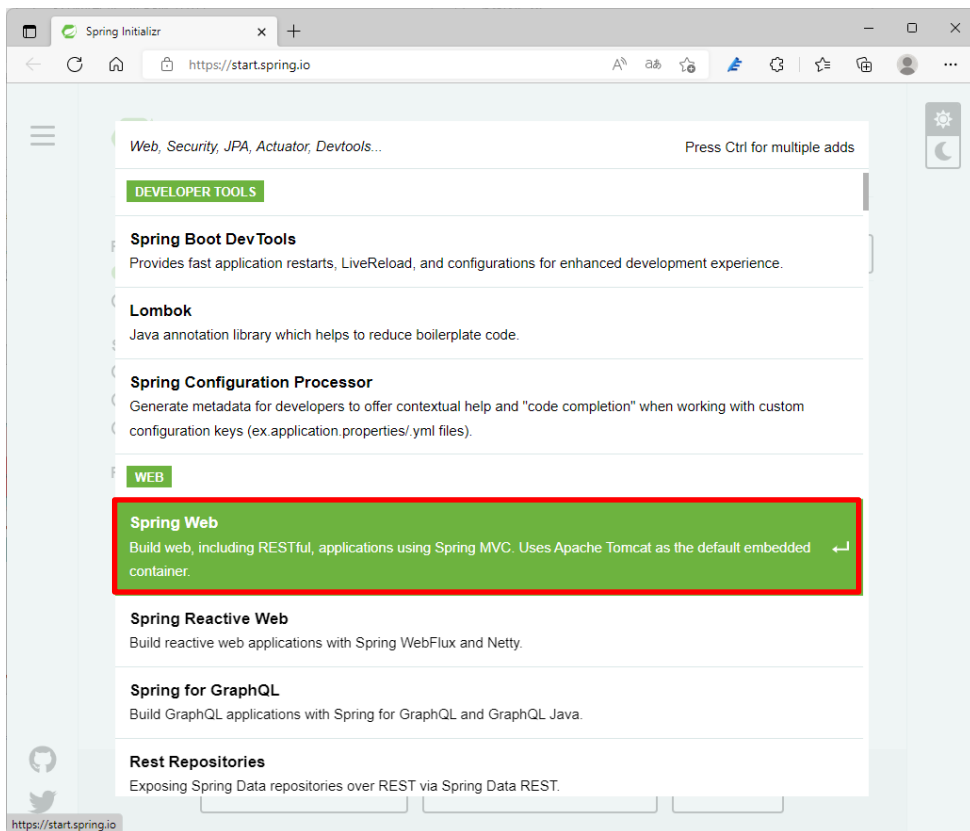
The screenshot shows the Spring Initializr web application interface. The browser address bar shows 'start.spring.io'. The page has a sidebar with a hamburger menu and a settings icon. The main content area is divided into several sections:

- Project**: Radio buttons for 'Gradle - Groovy', 'Gradle - Kotlin', and 'Maven'. 'Maven' is selected and highlighted with a red box.
- Language**: Radio buttons for 'Java', 'Kotlin', and 'Groovy'. 'Java' is selected.
- Spring Boot**: Radio buttons for versions '3.1.0 (SNAPSHOT)', '3.1.0 (M1)', '3.0.4 (SNAPSHOT)', '3.0.3', and '2.7.10 (SNAPSHOT)'. '3.0.3' is selected.
- Project Metadata**: Text input fields for 'Group' (com.example), 'Artifact' (helloworld), 'Name' (helloworld), 'Description' (Helloworld project for Spring Boot), and 'Package name' (com.example.helloworld).
- Dependencies**: A section with a button 'ADD DEPENDENCIES... CTRL + B' highlighted with a red box. Below it, 'Spring Web' is selected with a 'WEB' tag, and a description is provided: 'Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.'

At the bottom, there are three buttons: 'GENERATE CTRL + G', 'EXPLORE CTRL + SPACE', and 'SHARE...'.

Practice2 SpringBoot

Dependency



Practice2 SpringBoot

Project option 정의

- ☞ 개발 언어, 빌드 도구, 스프링 부트 버전과 기타사항을 선택
GENERATE 버튼을 클릭하면, 프로젝트 압축파일이 다운로드 됨.

The screenshot shows the Spring Initializr web application interface. The browser address bar displays 'start.spring.io'. The page features a sidebar with a hamburger menu and a settings icon. The main content area is divided into several sections:

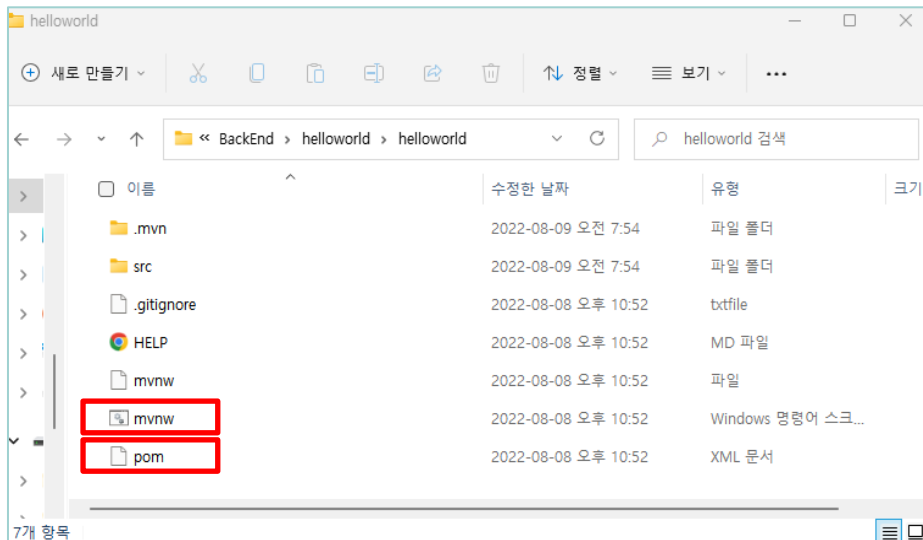
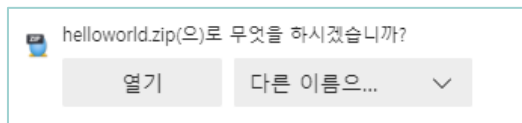
- Project:** Radio buttons for 'Gradle - Groovy', 'Gradle - Kotlin', and 'Maven' (selected).
- Language:** Radio buttons for 'Java' (selected), 'Kotlin', and 'Groovy'.
- Spring Boot:** Radio buttons for '3.1.0 (SNAPSHOT)', '3.1.0 (M1)', '3.0.4 (SNAPSHOT)', '3.0.3' (selected), and '2.7.10 (SNAPSHOT)'. There is also an option for '2.7.9'.
- Project Metadata:** Text input fields for 'Group' (com.example), 'Artifact' (helloworld), 'Name' (helloworld), 'Description' (Helloworld project for Spring Boot), and 'Package name' (com.example.helloworld, which is highlighted with a red rectangle).
- Dependencies:** A section with a 'Spring Web' dependency selected under the 'WEB' tab. A button 'ADD DEPENDENCIES... CTRL + B' is present.

At the bottom, there are three buttons: 'GENERATE CTRL + G', 'EXPLORE CTRL + SPACE', and 'SHARE...'.

Practice2 SpringBoot

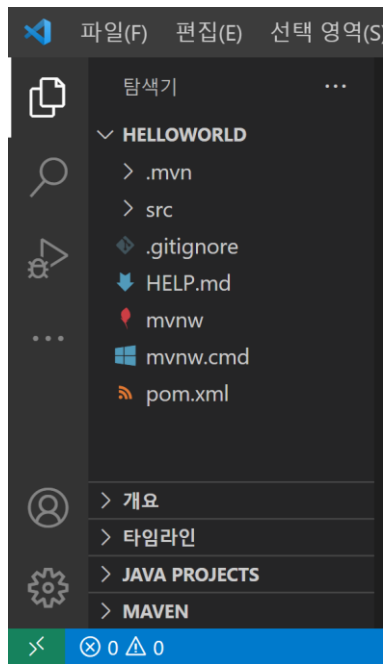
Project 다운로드 및 압축해제

👉 프로젝트를 다운로드하고 압축을 해제하여 프로젝트 구성 확인



- mvnw 실행파일은 메이븐 빌드도구
- pom파일은 XML 유형의 의존성 라이브러리와 빌드 정보에 대한 관리파일

Practice2 SpringBoot



VSCode에서 SpringBoot 프로젝트 열기

☞ VSCode IDE를 사용해서 해당 폴더를 열고 코드를 수정

```
package com.example.helloworld;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;
@SpringBootApplication
@RestController
public class HelloworldApplication {

    public static void main(String[] args) {
        SpringApplication.run(HelloworldApplication.class, args);
    }

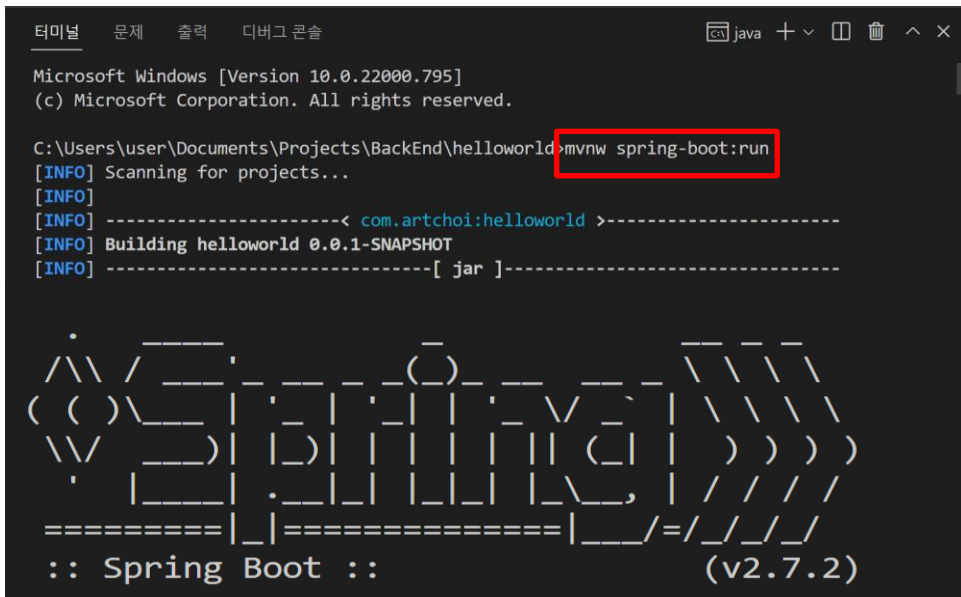
    @RequestMapping("/hello")
    public String hello(@RequestParam(value = "name", defaultValue = "World") String name) {
        return String.format("Hello %s!", name);
    }
}
```

Practice2 SpringBoot

- PowerShell
./mvnw spring-boot:run

SpringBoot 프로젝트 빌드 및 실행 (VSCode)

☞ VSCode IDE에서 스프링 부트 프로젝트를 빌드하고 실행
mvnw spring-boot:run



```
터미널 문제 출력 디버그 콘솔
Microsoft Windows [Version 10.0.22000.795]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user\Documents\Projects\BackEnd\helloworld>mvnw spring-boot:run
[INFO] Scanning for projects...
[INFO] -----< com.artchoi:helloworld >-----
[INFO] Building helloworld 0.0.1-SNAPSHOT
[INFO] -----[ jar ]-----

  ____ _
 / ___ \| | | |
/ /___ \| |_| |
\___)___|___|_|
:: Spring Boot ::
                    (v2.7.2)
```

*Extension Pack for Java 설치 권장

Practice2

SpringBoot

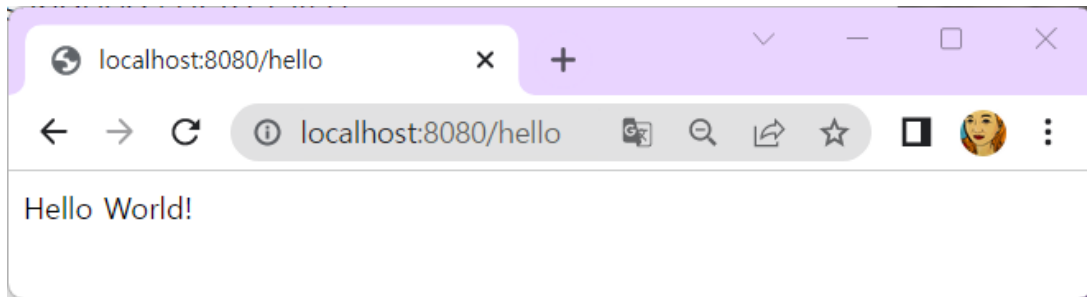
- 이미 8080포트가 실행되고 있다면 아래 명령어를 터미널에서 사용

```
netstat -a -o  
taskkill -f /pid 프로세스번호
```

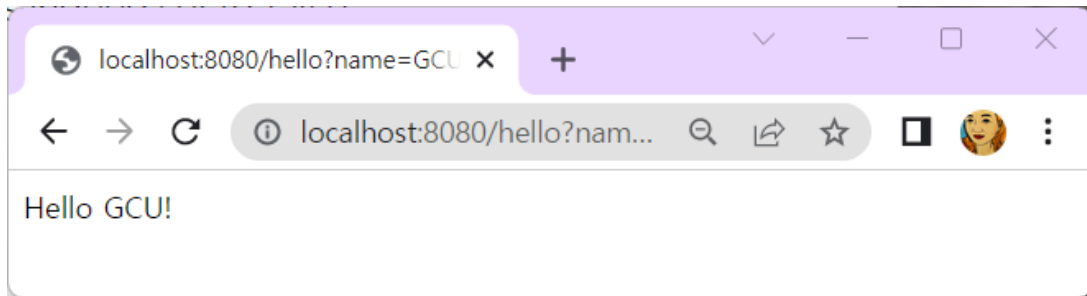
- 프로세스 제거

SpringBoot 프로젝트 빌드 (VSCode)

☞ 브라우저의 주소창에 아래 명령어를 실행
localhost:8080/hello



localhost:8080/hello?name=GCU



SW Engineering(1)

설문관리(챗봇) 서비스 기획 및 자료조사
기존 제품을 비교 분석하고 주요 기능을 도출합니다.

기존 제품 분석

자료조사



[상용 솔루션]

구글 Form; https://www.google.com/intl/ko_kr/forms/about/
카카오워크 설문;

[비교 분석표 작성]

접근성, 기능성, 특이한 점 등 분석한 내용을 정리해봅니다.

기존 제품 분석

Features Sample



[설문 관리자] 설문 관리

- 설문 생성
- 설문 제작; 다양한 유형(찬부식, 선다식, 서열식 등)으로 구성된 설문 작성
- 설문 수정/복사 등 관리 기능
- 설문 발송 대상자 관리
- 설문 발송(Email, SMS 등)

[설문 관리자] 설문 결과 분석

- 설문 결과

[피설문자]

- 설문 응답

요구분석

도출 문서

- ☞ ● 요구사항 명세서
- 유즈케이스 다이어그램
- 스토리 보드
- 와이어프레임 혹은 UI/UX, Figma
- 요구사항 변경 추적
- 테스트 시나리오

CONNECT.

SOLVE.

CREATE. 