

MS 클라우드 개발환경 깃허브 코드스페이스 활용



강환수 교수

AI Experts
Who Lead
The Future

01

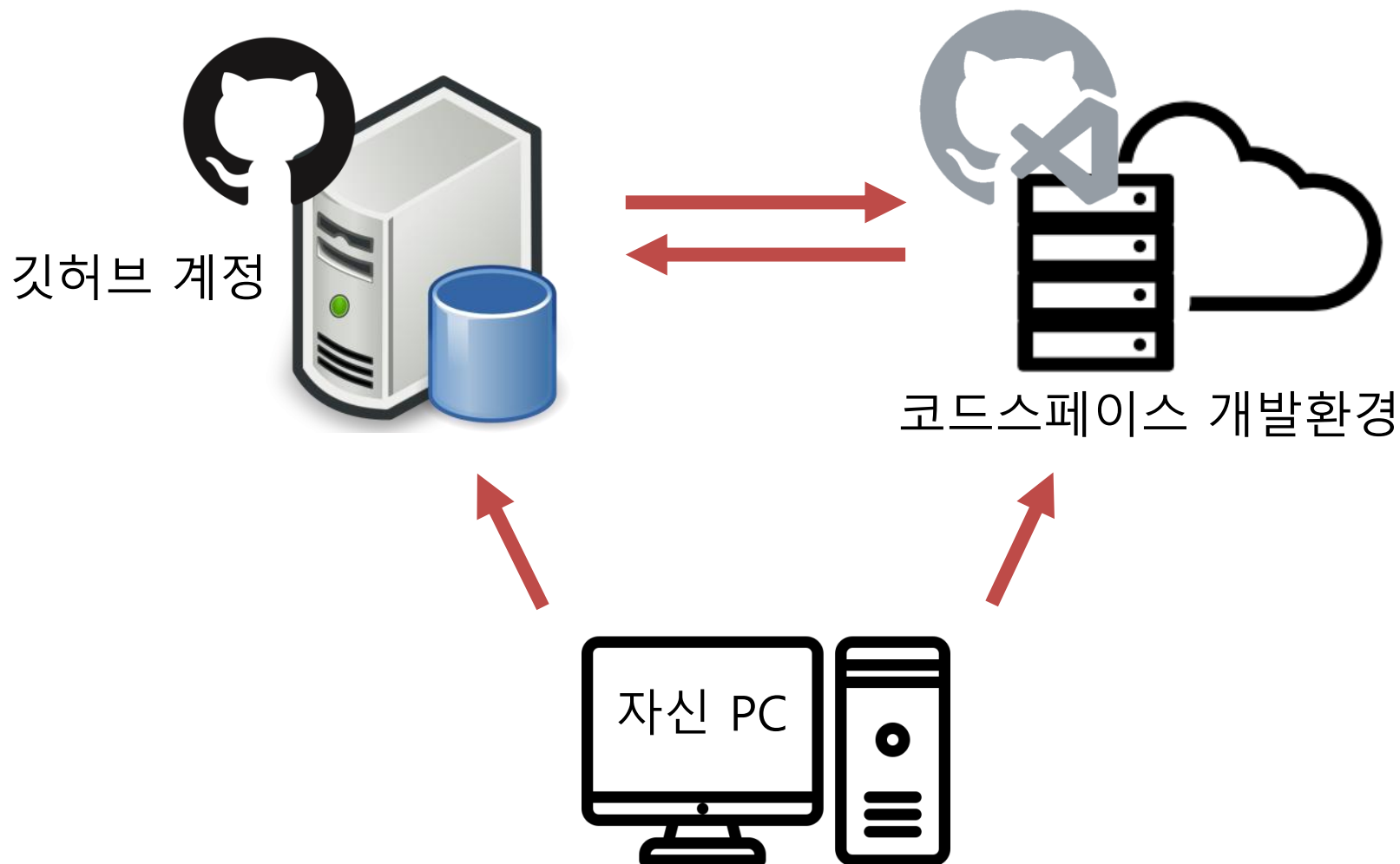
깃허브 코드스페이스 개요와 연결 실행

- **Github Codespaces**

- 깃허브 저장소 기반 클라우드 개발환경
 - 간단히 코드스페이스라고 명명
- 클라우드 기반 통합개발환경(cloud integrated development environments)
 - 여러 개발자가 협업하며 개발 가능
 - 기본으로 브라우저에서 개발환경 실행
- 장점
 - 다양한 분야의 개발환경을 지원하거나 설정 가능
 - 서버에서 웹개발이나 응용프로그램과 머신러닝, 딥러닝 개발 등
 - 자신의 PC 등의 지역 시스템에 소프트웨어를 설치하거나 유지 관리할 필요가 없음
 - Linux에서 컨테이너 개발환경 구성 및 실행
 - 개인적으로 별도의 개발환경을 설치할 필요가 없이 클라우드에서 제공하는 개발환경을 사용
 - » 필요하면 추가의 환경 설치도 가능
- 2022년 11월부터 무료 지원 확대

- 자신의 깃허브에서 서버에 연결, 수정 내용은 그대로 깃허브 저장소 저장 가능
 - 자신의 깃허브 저장소에서 마우스 클릭으로
 - 손쉽게 브라우저에서 원하는 코드를 작성해 컴파일과 디버깅, 실행 등의 작업을 수행
 - 다양한 방법으로 코드 공간에 연결 가능
 - 브라우저, Visual Studio Code, JetBrains Gateway 응용 프로그램 또는 GitHub CLI를 사용
 - 클라우드에 저장된 데이터와 코드는 버전관리 깃을 활용
 - 소스와 구성 파일을 저장소에 커밋하여 GitHub 코드 공간에 대한 프로젝트를 저장 가능

- 깃허브에서 실행하는 클라우드 개발환경



코드를 저장할 저장소 생성

오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

- **py-space**
 - README.md 체크해 생성

Create a New Repository

github.com/new

Search or jump to...

Pull requests Issues Codespaces Marketplace Explore

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 * chatkang / **Repository name *** py-space ✓

Great repository names are short and memorable. Need inspiration? How about [bug-free-rotary-phone?](#)

Description (optional)
기초 파이썬 코딩 활용

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

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

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: None

This will set `main` as the default branch. Change the default name in your [settings](#).

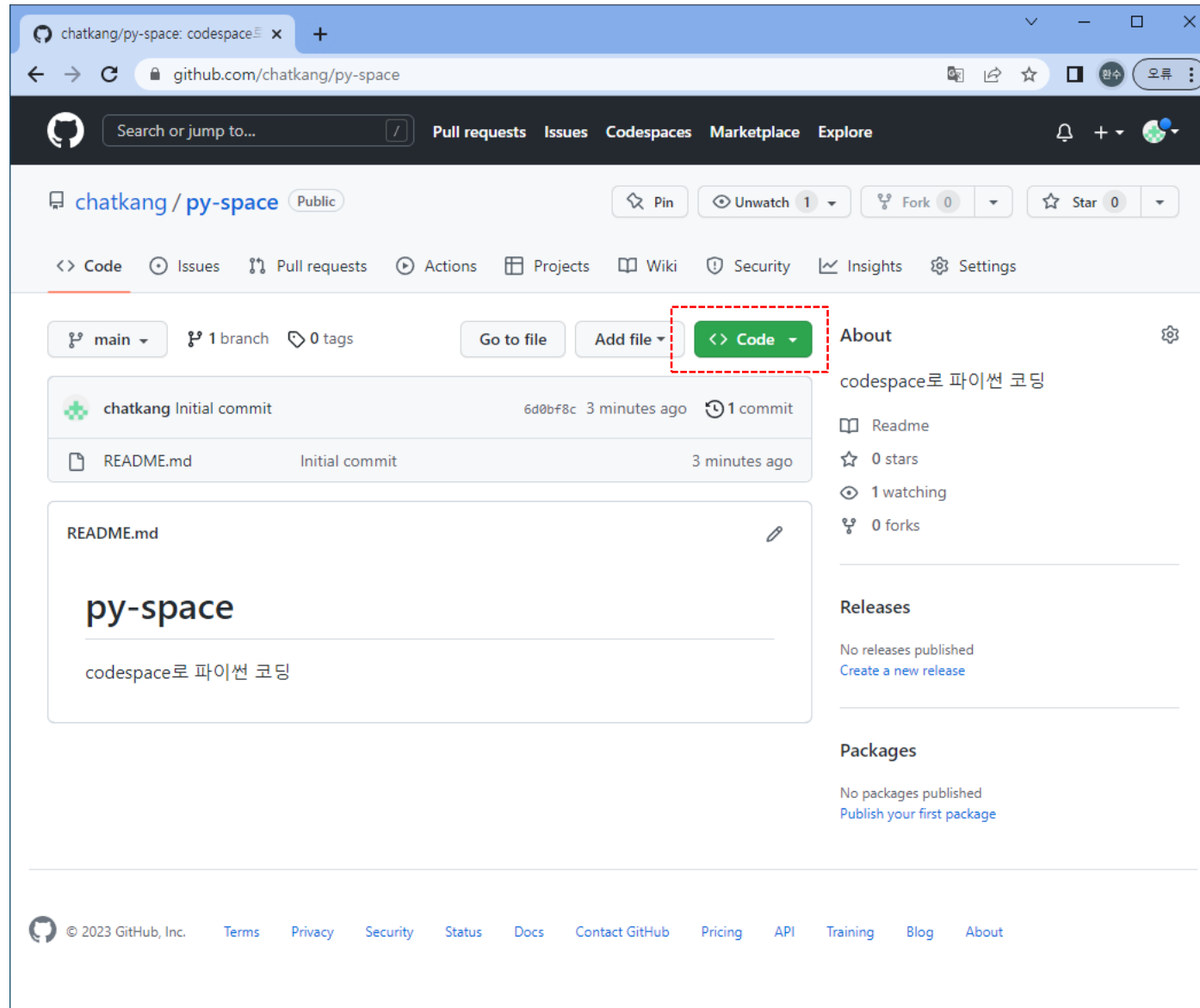
ⓘ You are creating a public repository in your personal account.

Create repository

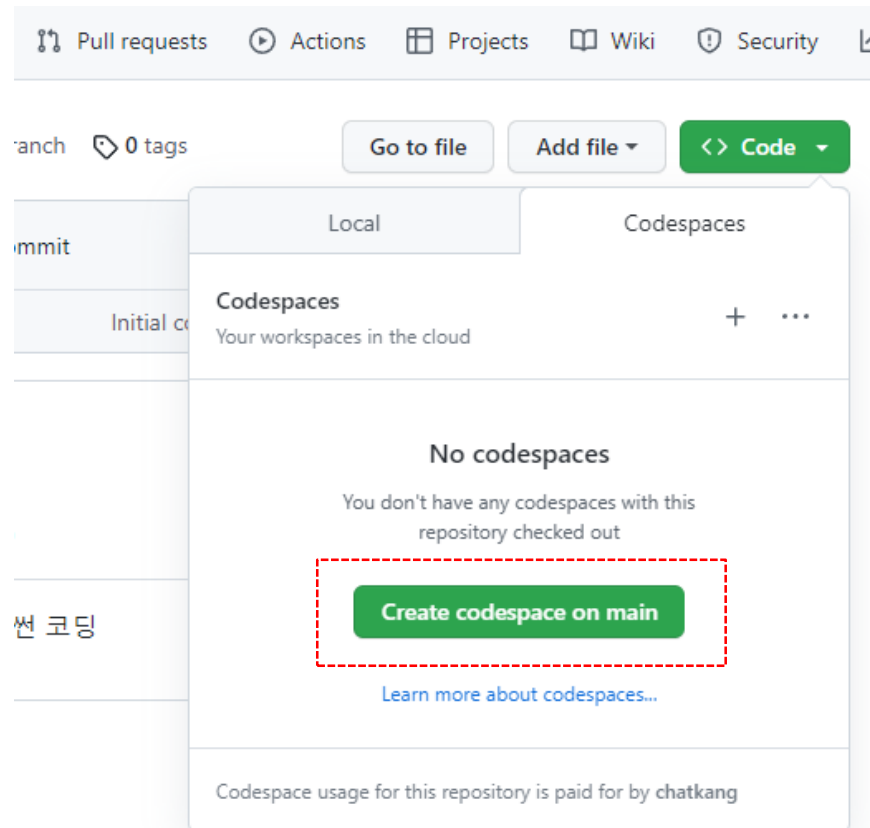
코드를 저장할 저장소 py-space

오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

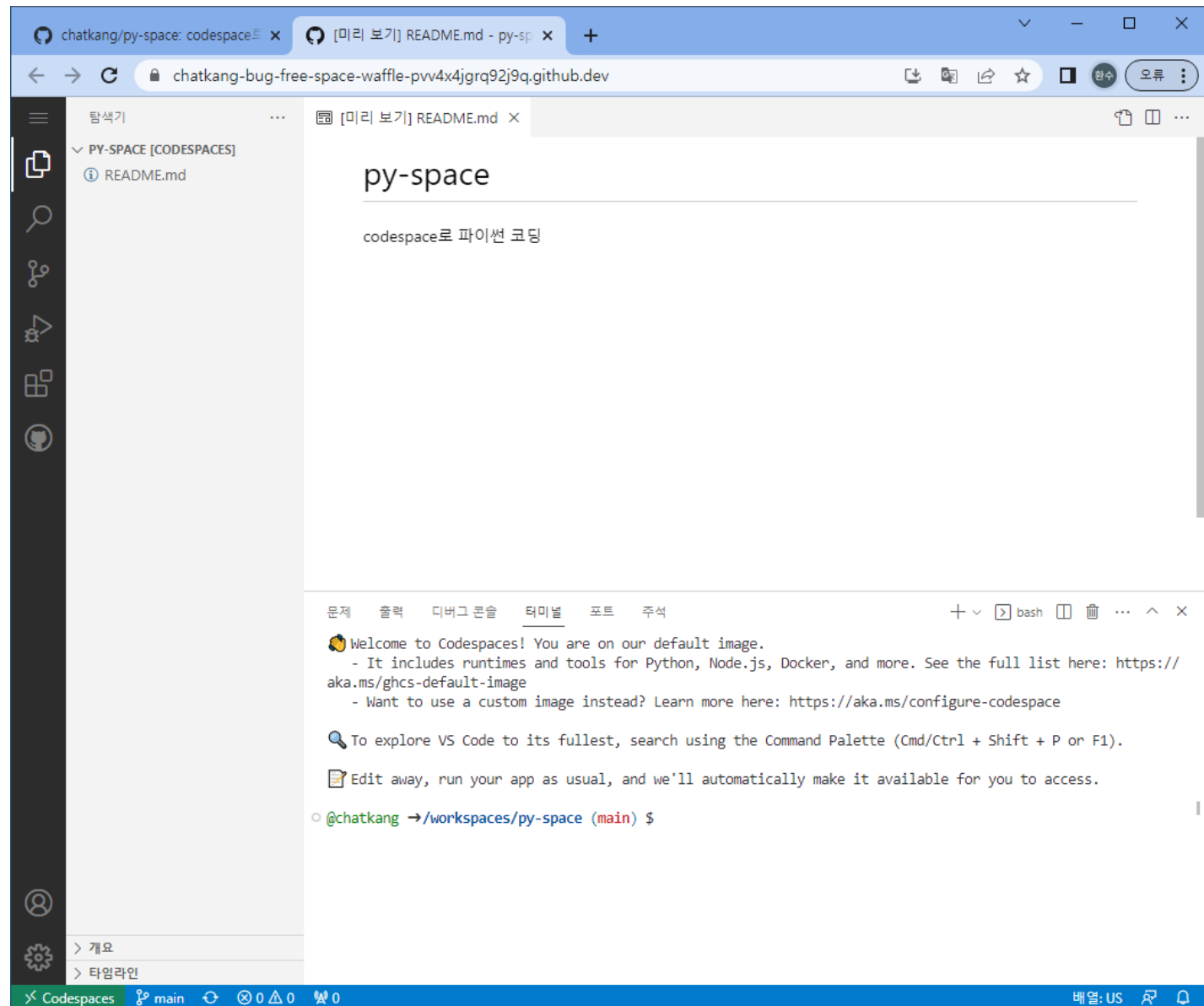
- py-space



- 버튼 Code 하부
 - Create codespace on main
 - 브랜치 메인에 코드스페이스 생성
 - 기본 설정으로 코드스페이스 생성해 연결
 - » 간단한 설정으로 파이썬 활용 가능



- 브라우저에 생성



파이썬 설치 모듈 확인

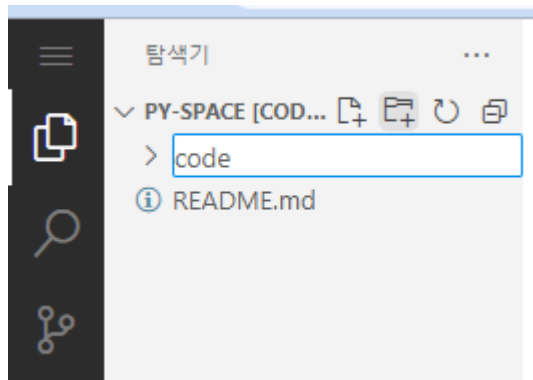
- \$ python --version
- \$ pip list
- \$ pip show numpy
- \$ python
- >>> import site, sys
- >>> import inspect
- >>> sys.version
- >>> site.getsitepackages()
- >>> import random
- >>> inspect.getfile(random)
- ...
- >>> exit()

```
문제  출력  디버그 콘솔  터미널  포트
setuptools      67.8.0
six             1.16.0
smmap           5.0.0
sniffio         1.3.0
soupsieve       2.4.1
stack-data      0.6.2
sympy           1.12
tenacity        8.2.2
terminado       0.17.1
threadpoolctl   3.1.0
tinyccs2        1.2.1
tomli           2.0.1
torch           2.0.1
tornado         6.3.2
traitlets       5.9.0
triton          2.0.0
typing_extensions 4.6.3
tzdata          2023.3
uri-template    1.2.0
urllib3         2.0.3
wcwidth         0.2.6
webcolors       1.13
webencodings    0.5.1
websocket-client 1.5.3
wheel           0.40.0

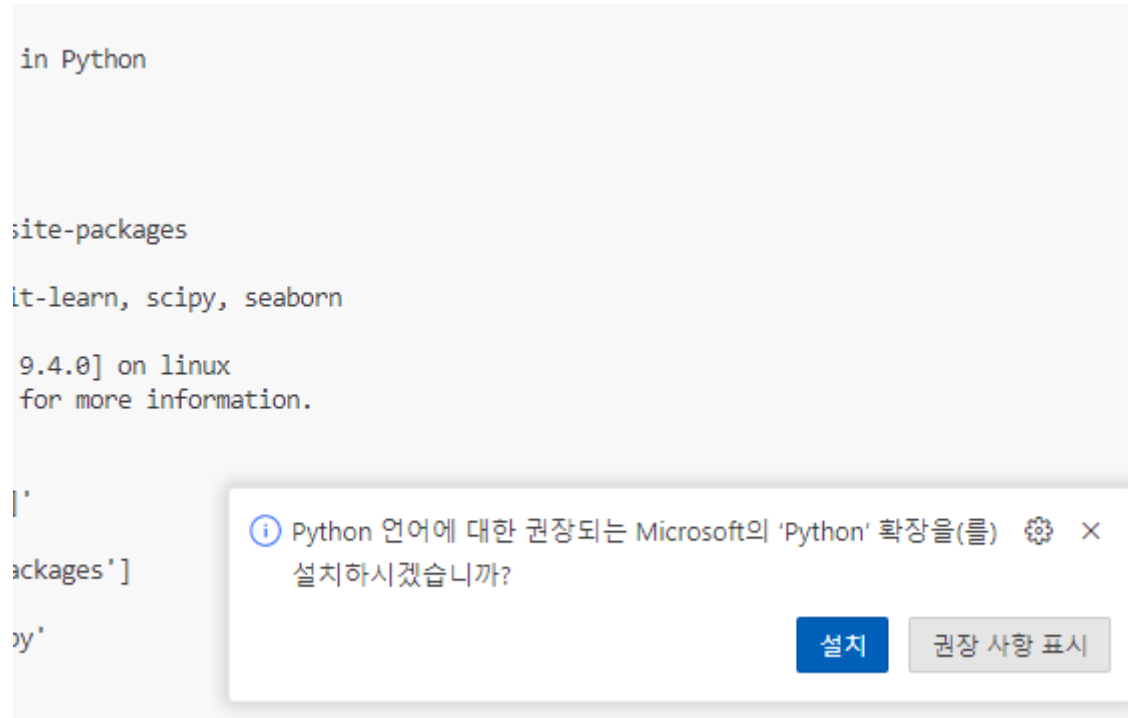
[notice] A new release of pip is available: 23.1.2 -> 23.2.1
[notice] To update, run: python -m pip install --upgrade pip
@chatkang →/workspaces/py-space (main) $ pip shpw numpy
ERROR: unknown command "shpw" - maybe you meant "show"
@chatkang →/workspaces/py-space (main) $ pip show numpy
Name: numpy
Version: 1.24.3
Summary: Fundamental package for array computing in Python
Home-page: https://www.numpy.org
Author: Travis E. Oliphant et al.
Author-email:
License: BSD-3-Clause
Location: /home/codespace/.local/lib/python3.10/site-packages
Requires:
Required-by: contourpy, matplotlib, pandas, scikit-learn, scipy, seaborn
@chatkang →/workspaces/py-space (main) $ python
Python 3.10.8 (main, Jun 15 2023, 01:39:58) [GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import site, sys, inspect, random
>>> sys.version
'3.10.8 (main, Jun 15 2023, 01:39:58) [GCC 9.4.0]'
>>> site.getsitepackages()
['/usr/local/python/3.10.8/lib/python3.10/site-packages']
>>> inspect.getfile(random)
'/usr/local/python/3.10.8/lib/python3.10/random.py'
>>> exit()
@chatkang →/workspaces/py-space (main) $
```

- 파일 생성

- 폴더 code 생성 후 하부에 파일 hello.py 생성



- 작성하면 우측 하단에 확장 설치 권장 화면 표시



```
문제 출력 디버그 콘솔 터미널 포트 주석
Welcome to Codespaces! You are on our default image.
- It includes runtimes and tools for Python, Node.js, Docker, and more. See the full list here: https://aka.ms/ghcs-default-image
- Want to use a custom image instead? Learn more here: https://aka.ms/configure-codespace

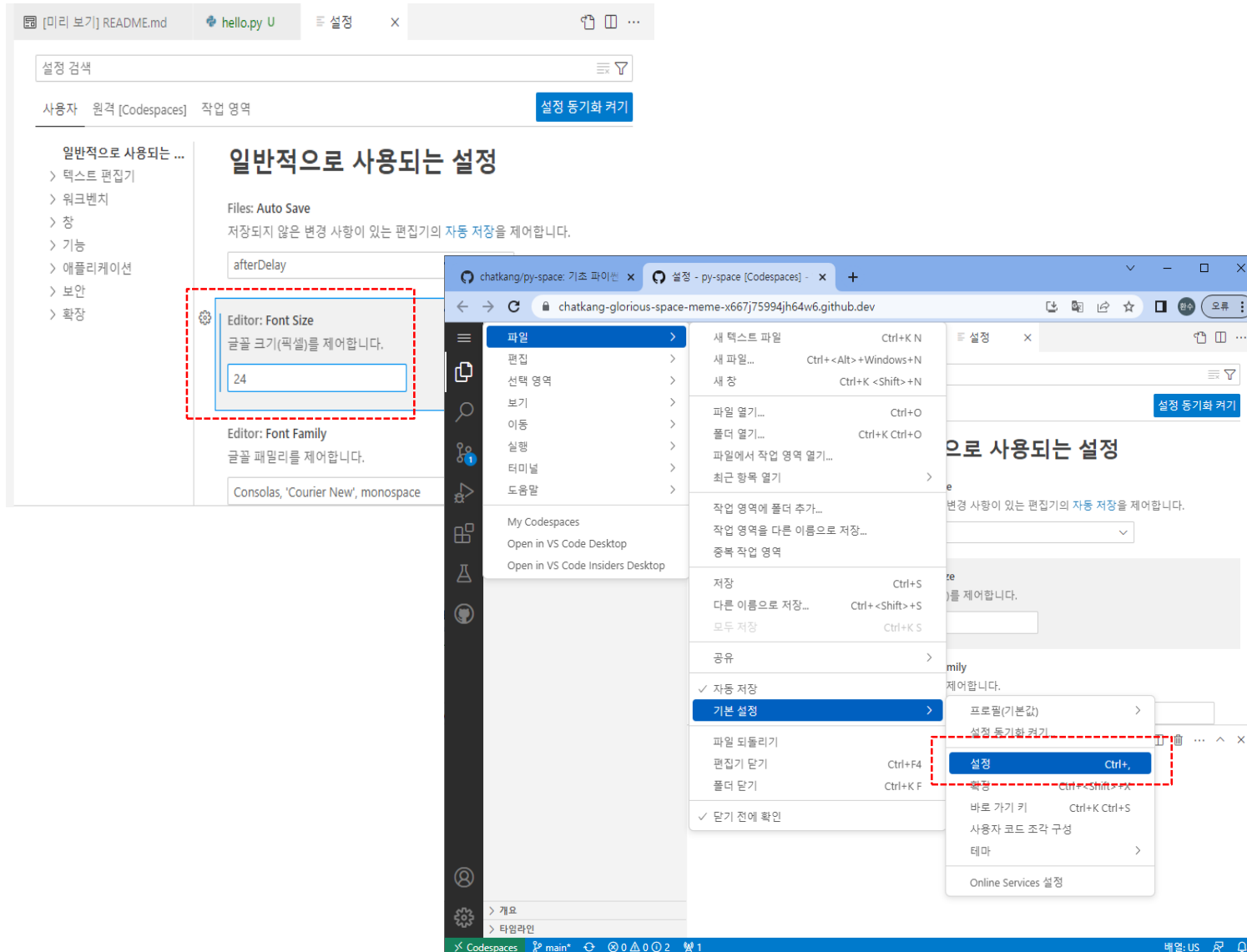
To explore VS Code to its fullest, search using the Command Palette (Cmd/Ctrl + Shift + P or F1).

Edit away, run your app as usual, and we'll automatically make it available for you to access.

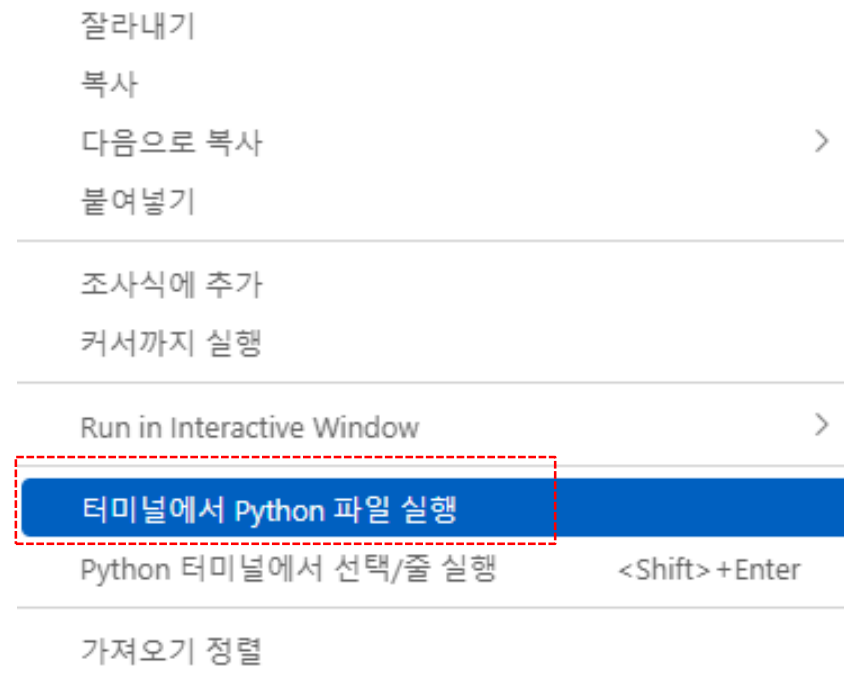
@chatkang →/workspaces/py-space (main) $
```



- 단축키 **ctrl + ,**



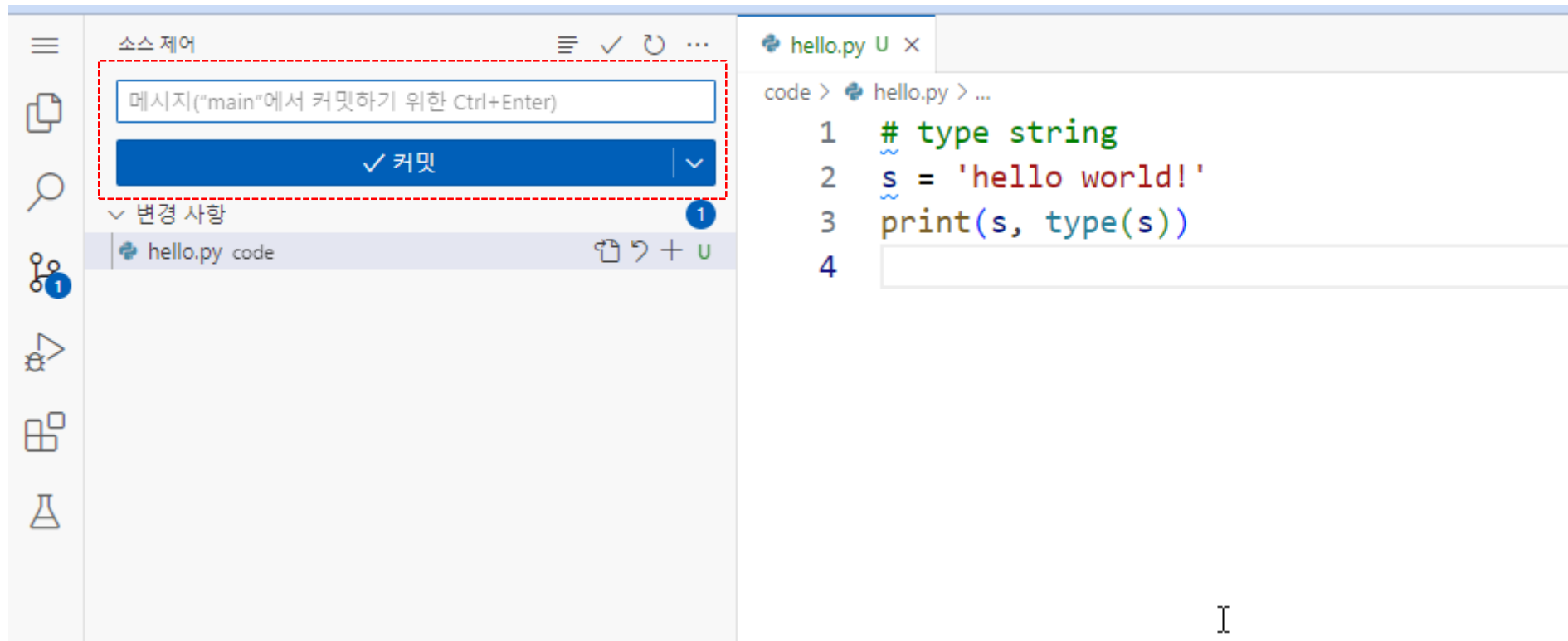
- 실행
 - 소스에서 팝업 메뉴



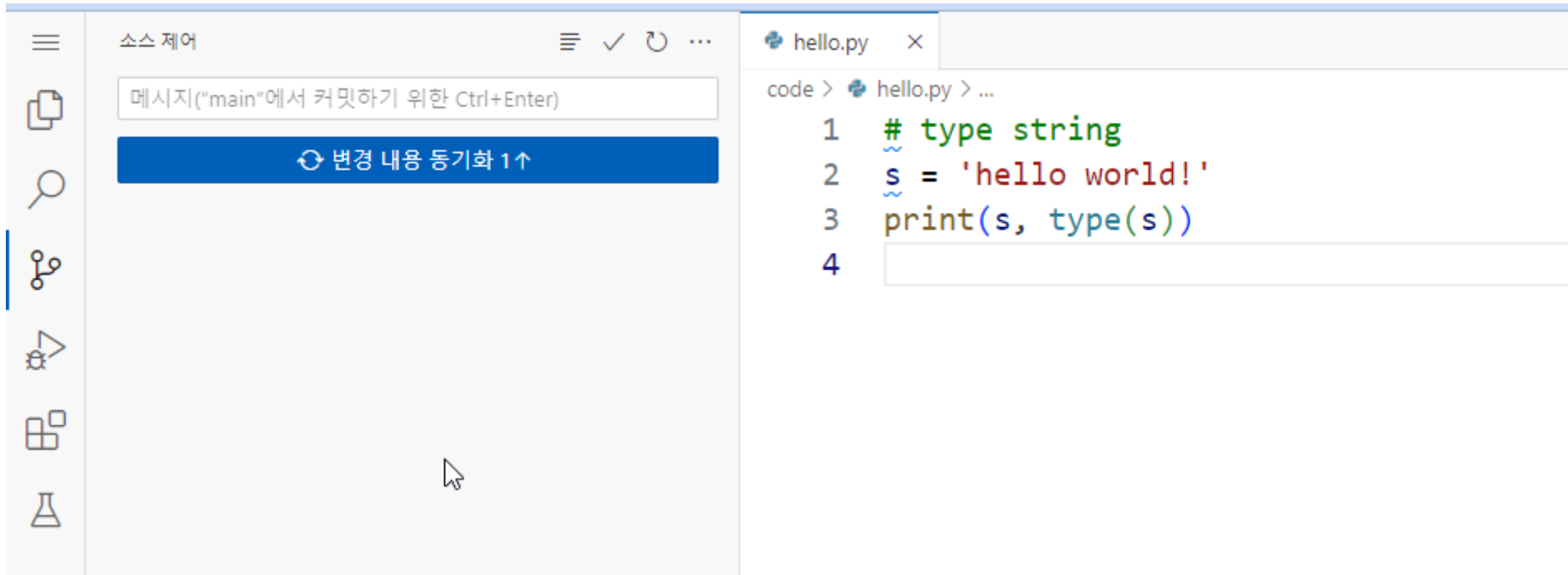
- 실행 결과

```

문제 2  출력  디버그 콘솔  터미널  포트 1  주석
/home/codespace/.python/current/bin/python3 /workspaces/py-space/code/hello.py
• @chatkang →/workspaces/py-space (main) $ /home/codespace/.python/current/bin/python3 /wor
kspaces/py-space/code/hello.py
Hello, Codespaces!
['p', 'y', 't', 'h', 'o', 'n']
○ @chatkang →/workspaces/py-space (main) $
    
```



- 코드스페이스 저장소 -> 깃허브 저장소: 저장



깃허브 저장소 확인

오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

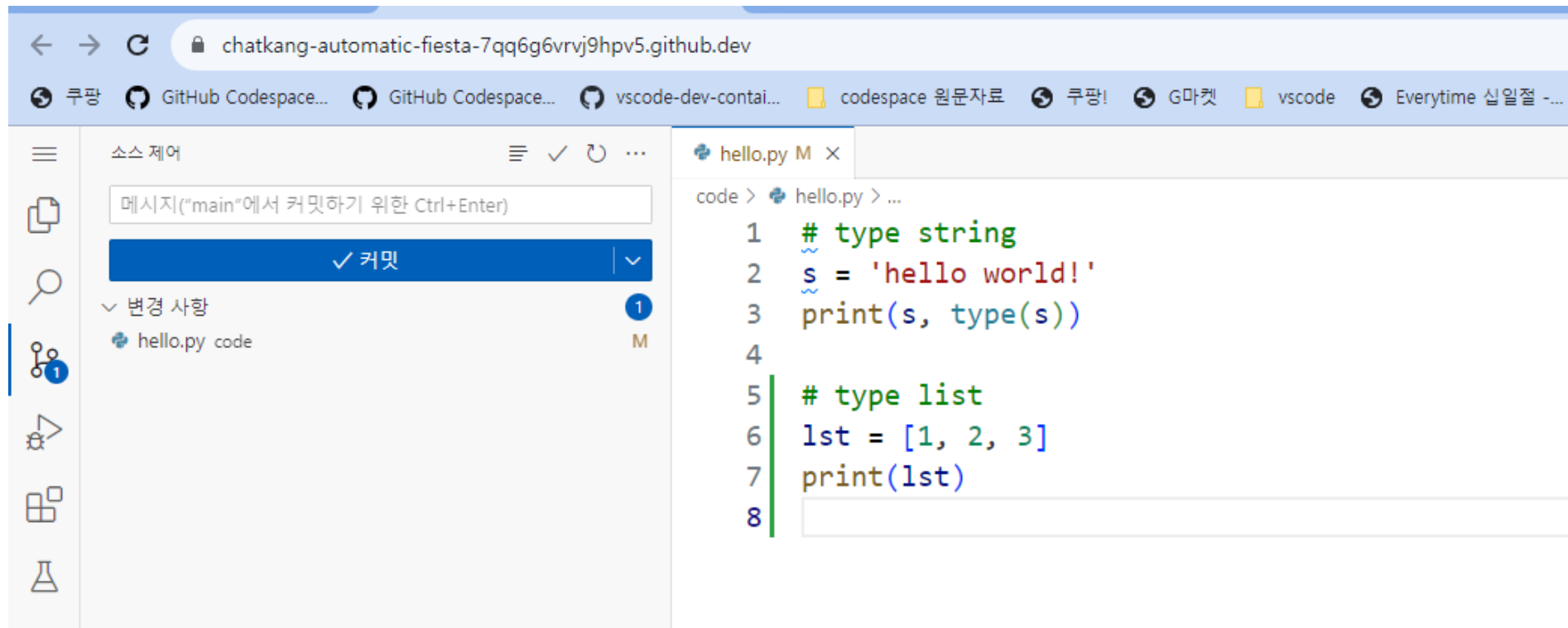
The screenshot shows the GitHub interface for the repository 'chatkang / py-space'. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The repository path is 'py-space / code /'. A search bar is present with the text 'Type / to search'. Below the navigation bar, there is a commit history section showing a commit by 'chatkang' with the message '문자열 학습' (String Learning) and a commit hash '971e7aa' from '1 minute ago'. A table below the commit history lists the files in the repository.

Name	Last commit message	Last commit date
..		
hello.py	문자열 학습	1 minute ago

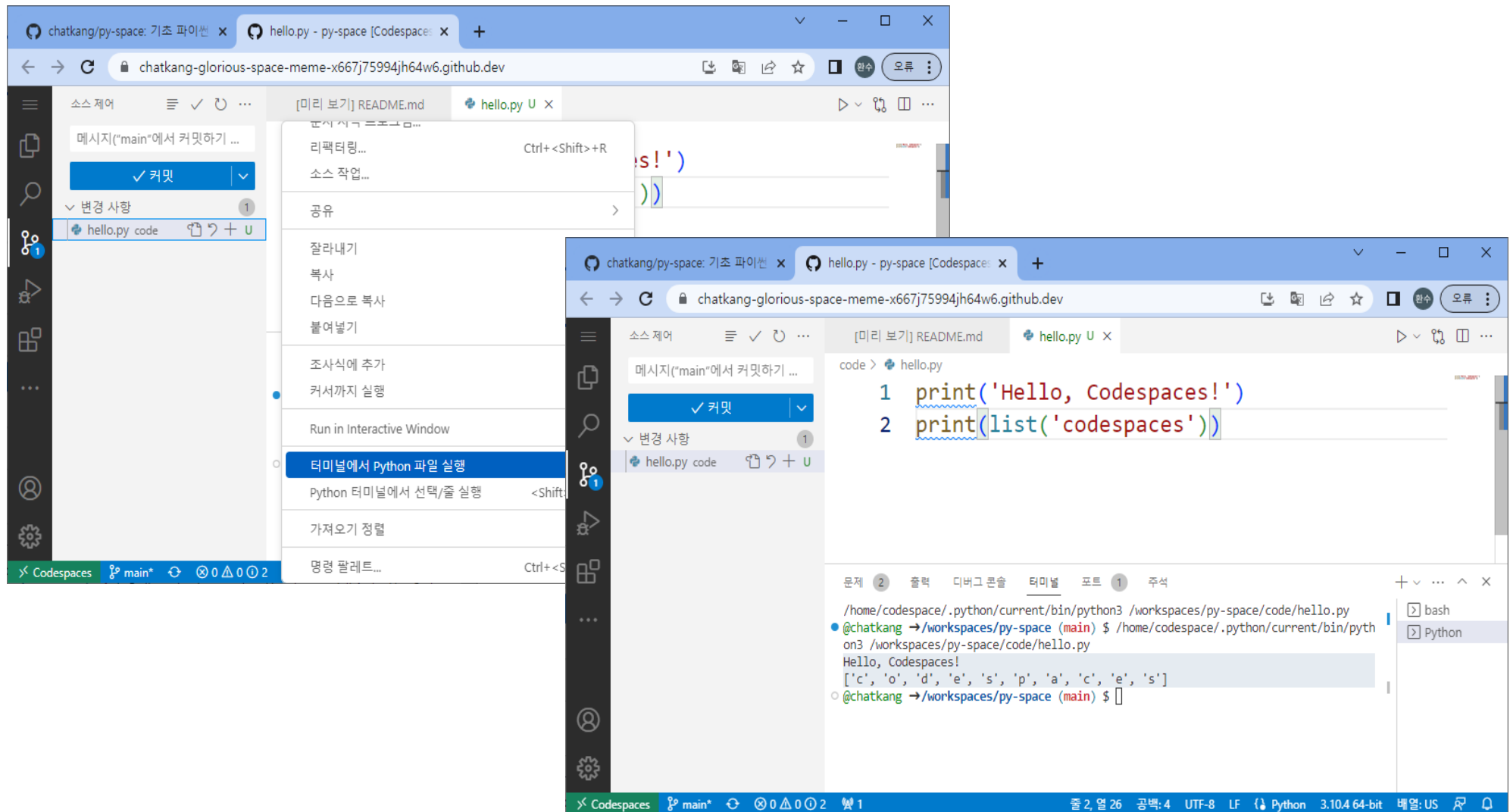
AI Experts
Who Lead
The Future

02

다양한 파이썬 실행



- 파이썬 파일 전체를 실행
 - 메뉴 [터미널에서 Python 파일 실행]

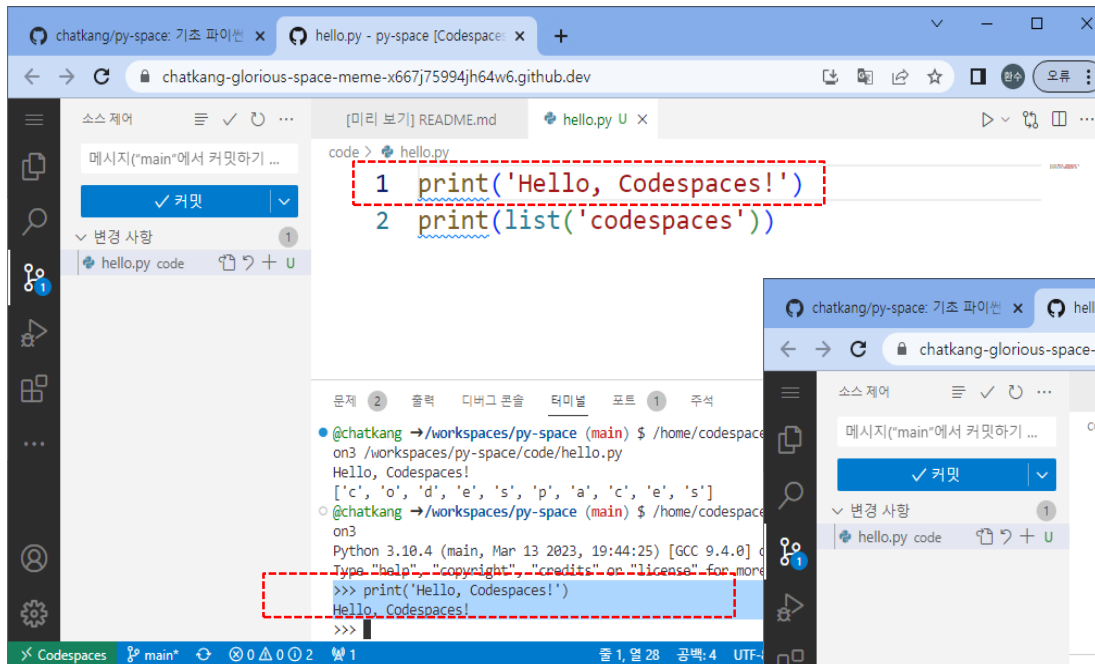


- `ctrl + F5`
- 우측 상단 삼각형

파이썬 터미널에서 선택/줄 실행

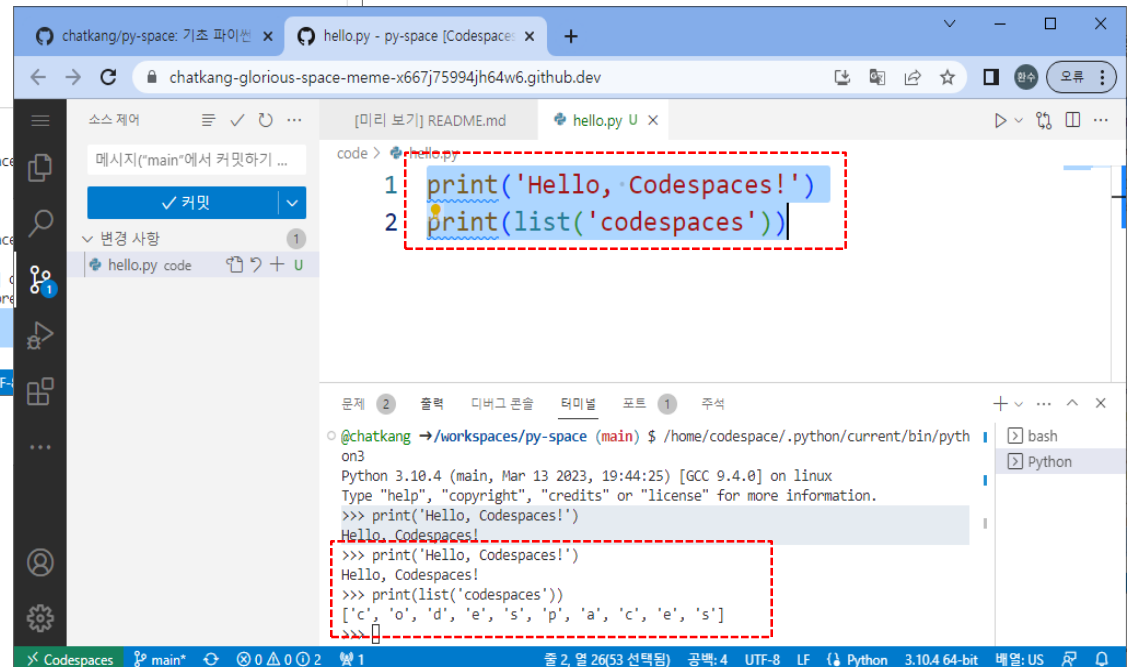
오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

- 선택한 줄/블록이 실행
 - 실행을 원하는 줄에 커서를 놓고 **shift + enter**
 - Python 쉘로 실행



```
1 print('Hello, Codespaces!')
2 print(list('codespaces'))
```

```
@chatkang →/workspaces/py-space (main) $ /home/codespace/on3 /workspaces/py-space/code/hello.py
Hello, Codespaces!
['c', 'o', 'd', 'e', 's', 'p', 'a', 'c', 'e', 's']
@chatkang →/workspaces/py-space (main) $ /home/codespace/on3 Python 3.10.4 (main, Mar 13 2023, 19:44:25) [GCC 9.4.0]
Type "help", "copyright", "credits" or "license()" for more
>>> print('Hello, Codespaces!')
Hello, Codespaces!
>>>
```



```
1 print('Hello, Codespaces!')
2 print(list('codespaces'))
```

```
@chatkang →/workspaces/py-space (main) $ /home/codespace/.python/current/bin/python3
Python 3.10.4 (main, Mar 13 2023, 19:44:25) [GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello, Codespaces!')
Hello, Codespaces!
>>> print('Hello, Codespaces!')
Hello, Codespaces!
>>> print(list('codespaces'))
['c', 'o', 'd', 'e', 's', 'p', 'a', 'c', 'e', 's']
>>>
```

AI Experts
Who Lead
The Future

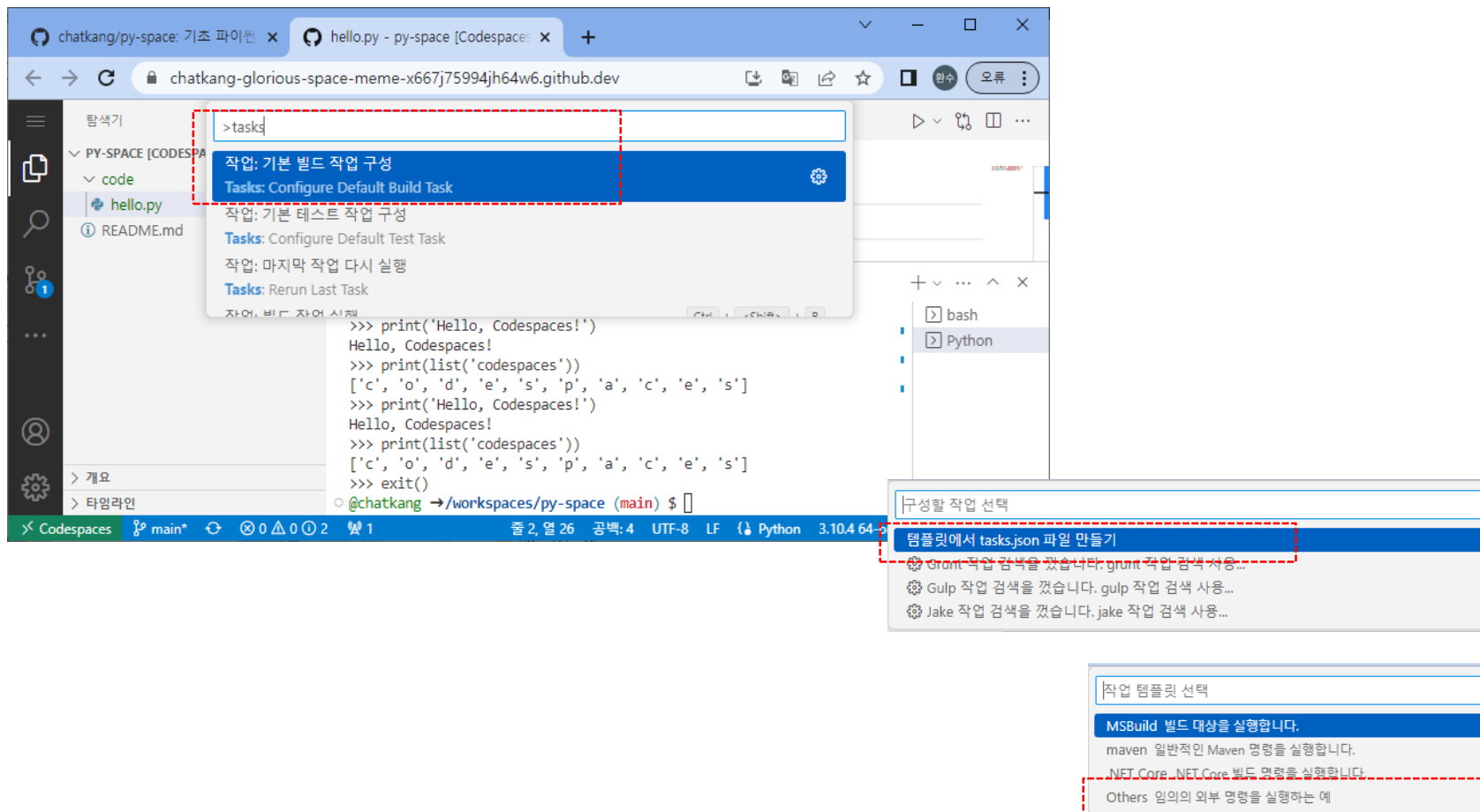
03

빌드 작업 실행

실행할 빌드 작업으로 파이썬 실행 설정

오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

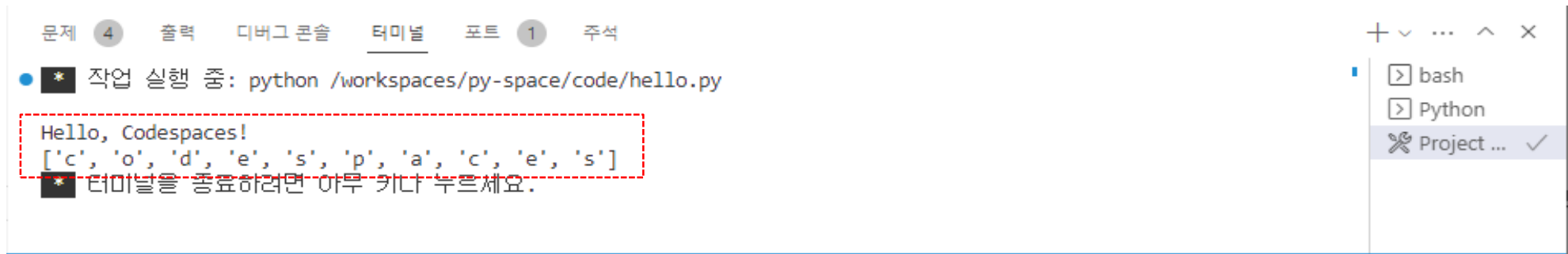
- Vs code에서 실행할 빌드를 설정하기 위해 먼저 ctrl + shift + P로 팔레트를 실행
 - Tasks 입력
 - [템플릿에서 tasks.json 파일 만들기]
 - 다음으로 마지막 항목 [Others 임의의 외부 명령을 실행하는 예]



- 파일 `.vscode/tasks.json` 수정 후 저장(ctrl + S)

```
{
  // See https://go.microsoft.com/fwlink/?LinkId=733558
  // for the documentation about the tasks.json format
  "version": "2.0.0",
  "tasks": [
    {
      "label": "Project Label",
      "type": "shell",
      "command": "python",
      "args": [
        "${file}"
      ],
      "presentation": {
        "reveal": "always",
        "panel": "new"
      },
      "options": {
        "env": {
          "PYTHONIOENCODING": "UTF-8"
        }
      },
      "group": {
        "kind": "build",
        "isDefault": true
      }
    }
  ]
}
```

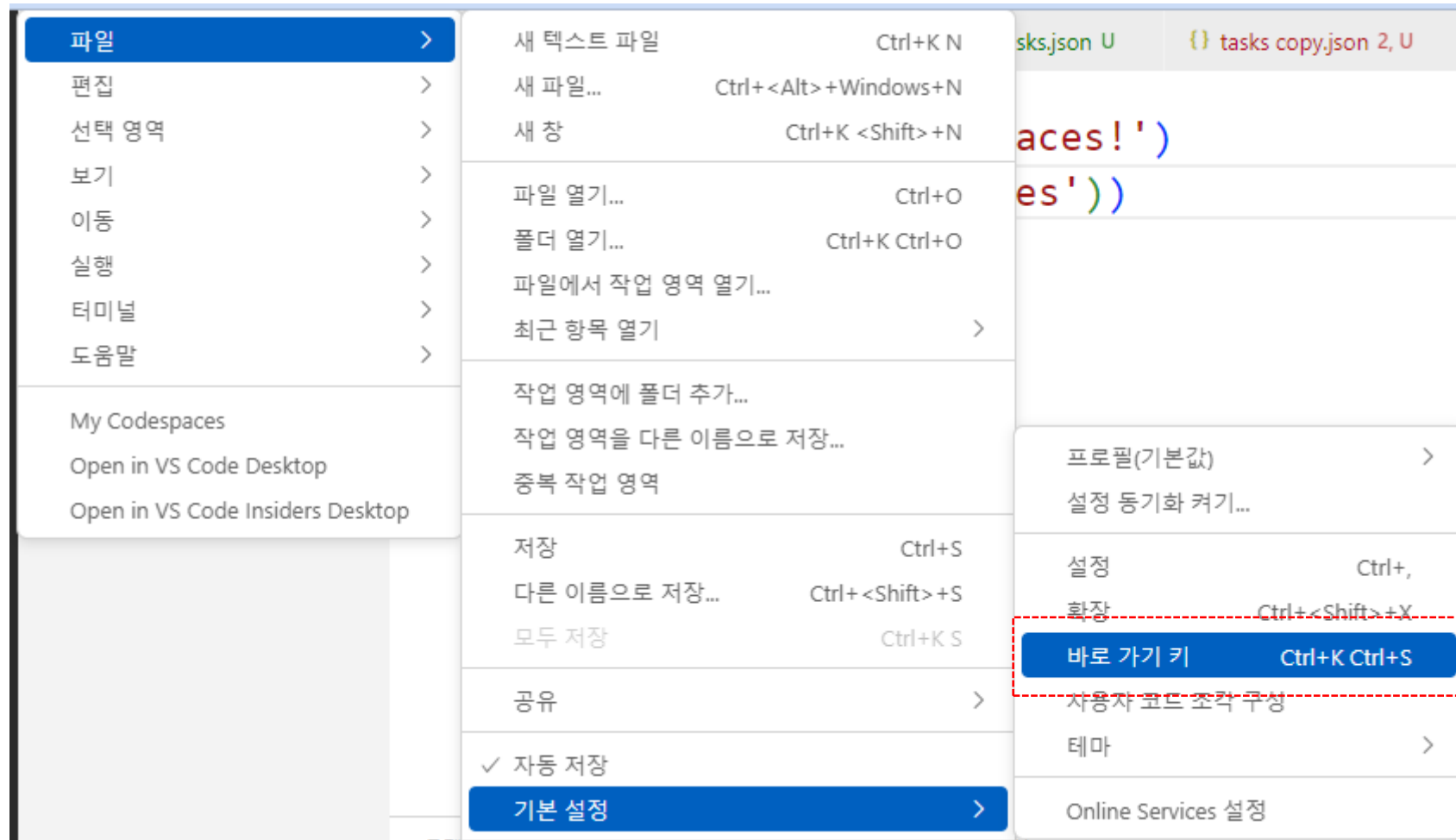
- 파이썬 소스 파일에서 설정한 태스크를 Ctrl + Shift + B로 실행



```
문제 4 출력 디버그 콘솔 터미널 포트 1 주석
* 작업 실행 중: python /workspaces/py-space/code/hello.py
Hello, Codespaces!
['c', 'o', 'd', 'e', 's', 'p', 'a', 'c', 'e', 's']
* 터미널을 종료하려면 아무 키나 누르세요.
```

- 단축 키를 설정

- 메뉴 [파일 | 기본 설정 | 바로 가기 키], 단축 키는 **ctrl + K ctrl + S**



현재 설정된 바로 가기 키가 표시

오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

[미리 보기] README.md hello.py U 바로 가기 키 X tasks.json U tasks copy.json 2, U			
키 바인딩에서 검색하려면 입력			
명령	키 바인딩	언제	소스
<Tab> 키로 포커스 이동 설정/해제	Ctrl + M	-	시스템
가리키기 또는 포커스 표시	Ctrl + K Ctrl + I	editorTextFocus	시스템
가져오기 구성	<Shift> + <Alt> + O	editorTextFocus && !editorReadOnly && suppo...	시스템
개발자: 창 다시 로드	Ctrl + R	isDevelopment	시스템
검색 편집기: 검색 편집기 입력 포커스	Escape	inSearchEditor	시스템
검색 편집기: 다시 검색	Ctrl + <Shift> + R	inSearchEditor	시스템
검색 편집기: 단어 단위로 토글	<Alt> + W	inSearchEditor && searchInputBoxFocus	시스템
검색 편집기: 대/소문자 구분 토글	<Alt> + C	inSearchEditor && searchInputBoxFocus	시스템
검색 편집기: 모든 일치 항목 선택	Ctrl + <Shift> + L	inSearchEditor	시스템
검색 편집기: 정규식 사용 토글	<Alt> + R	inSearchEditor && searchInputBoxFocus	시스템
검색 편집기: 컨텍스트 줄 늘이기	<Alt> + =	inSearchEditor	시스템
검색 편집기: 컨텍스트 줄 줄이기	<Alt> + -	inSearchEditor	시스템
검색 편집기: 컨텍스트 줄 토글	<Alt> + L	inSearchEditor	시스템
검색 편집기: 파일 결과 삭제	Ctrl + <Shift> + Backspace	inSearchEditor	시스템
검색 편집기: 편집기에서 결과 열기	<Alt> + Enter	hasSearchResult && searchViewletFocus	시스템
검색: 검색 취소	Escape	listFocus && searchViewletVisible && !input...	시스템
검색: 다음 검색 결과에 포커스	F4	hasSearchResult inSearchEditor	시스템
검색: 이전 검색 결과에 포커스	<Shift> + F4	hasSearchResult inSearchEditor	시스템
검색: 파일에서 바꾸기	Ctrl + <Shift> + H	-	시스템
검색: 파일에서 찾기	Ctrl + <Shift> + F	-	시스템
구현으로 이동	Ctrl + F12	editorHasImplementationProvider && editorTe...	시스템

- 검색 입력 창에 '터미널에서 ...' 입력
 - 단축키를 설정하고 싶은 마지막 항목에서 더블 클릭

터미널에서

명령	키 바인딩	언제	소스
Python: Python 터미널에서 선택/줄 실행	<Shift> + Enter	editorTextFocus && !findInputFocussed && !j...	Python
터미널: 활성 터미널에서 선택			
터미널: 활성 터미널에서 활			
Python: 터미널에서 Python			

터미널에서

명령	키 바인딩	언제	소스
Python: Python 터미널에서 선택/줄 실행	<Shift> + Enter	editorTextFocus && !findInputFocussed && !j...	Python
터미널: 활성 터미널에서 선택한 텍스트 실행		-	시스템
터미널: 활성 터미널에서 활성 파일 실행		-	시스템
+ Python: 터미널에서 Python 파일 실행		-	Python

원하는 키 조합을 누르고 <Enter> 키를 누르세요.

ctrl+shift

Ctrl + <Shift> +

- 직접 원하는 조합 키 Ctrl + Shift + F10를 누른 후 Enter키



명령	키 바인딩	언제	소스
Python: 터미널에서 Python 파일 실행	Ctrl + <Shift> + F10	-	사용자
Python: Python 터미널에서 선택/줄 실행	<Shift> + Enter	editorTextFocus && !findInputFocussed && !j...	Python
터미널: 활성 터미널에서 선택한 텍스트 실행		-	시스템
터미널: 활성 터미널에서 활성 파일 실행		-	시스템

설정된 단축 키 실행

오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

- **ctrl + shift + F10** 으로 실행

The screenshot displays a web-based IDE interface for a GitHub Codespace. The browser address bar shows the URL `chatkang-glorious-space-meme-x667j75994jh64w6.github.dev`. The left sidebar contains a file explorer with the following structure:

- PY-SPACE [CODESPACES]
 - .vscode
 - tasks copy.json
 - tasks.json
 - code
 - hello.py
 - README.md

The main editor area shows the `hello.py` file with the following code:

```
1 print('Hello, Codespaces!')
2 print(list('codespaces'))
```

Below the editor, the 'TERMINAL' tab is active, showing the execution output:

```
Hello, Codespaces!
>>> print(list('codespaces'))
['c', 'o', 'd', 'e', 's', 'p', 'a', 'c', 'e', 's']
>>> exit()
@chatkang → /workspaces/py-space (main) $ /home/codespace/.python/current/bin/python3 /workspaces/py-space/code/hello.py
Hello, Codespaces!
['c', 'o', 'd', 'e', 's', 'p', 'a', 'c', 'e', 's']
@chatkang → /workspaces/py-space (main) $
```

The bottom status bar indicates the environment is 'Python 3.10.4 64-bit' with a 'UTF-8' encoding and 'LF' line endings.

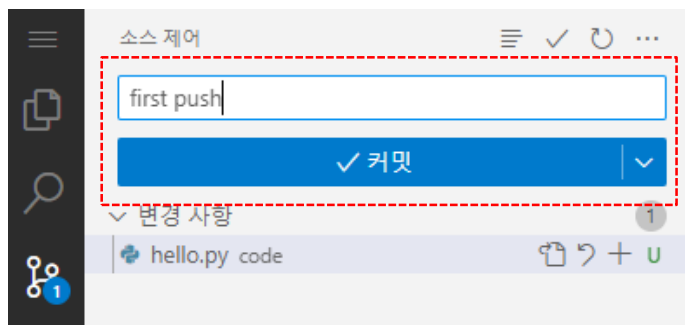
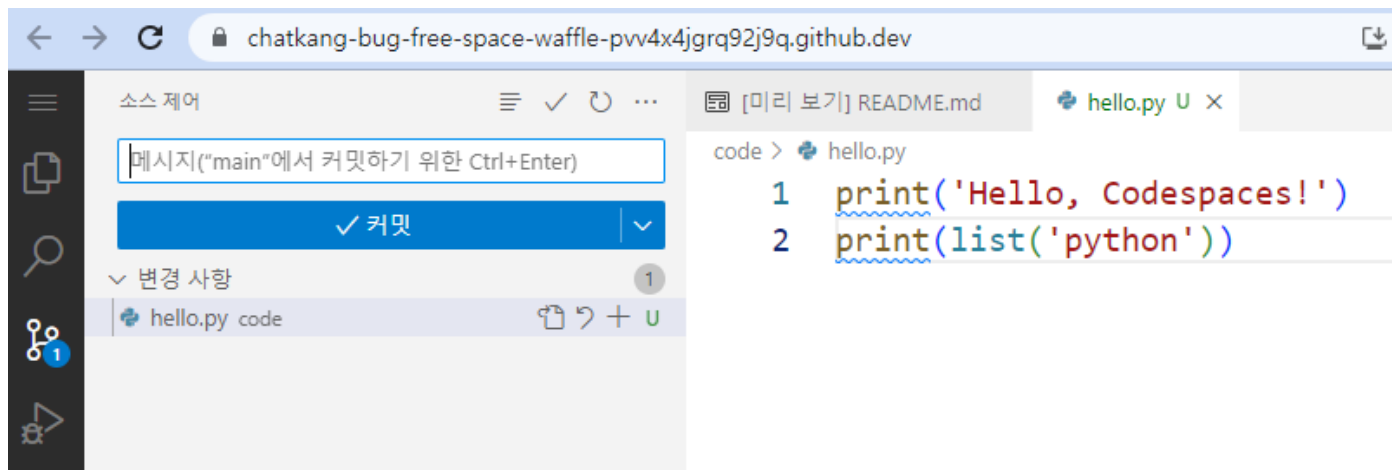
AI Experts
Who Lead
The Future

04

소스 파일 깃허브 저장

- 커밋

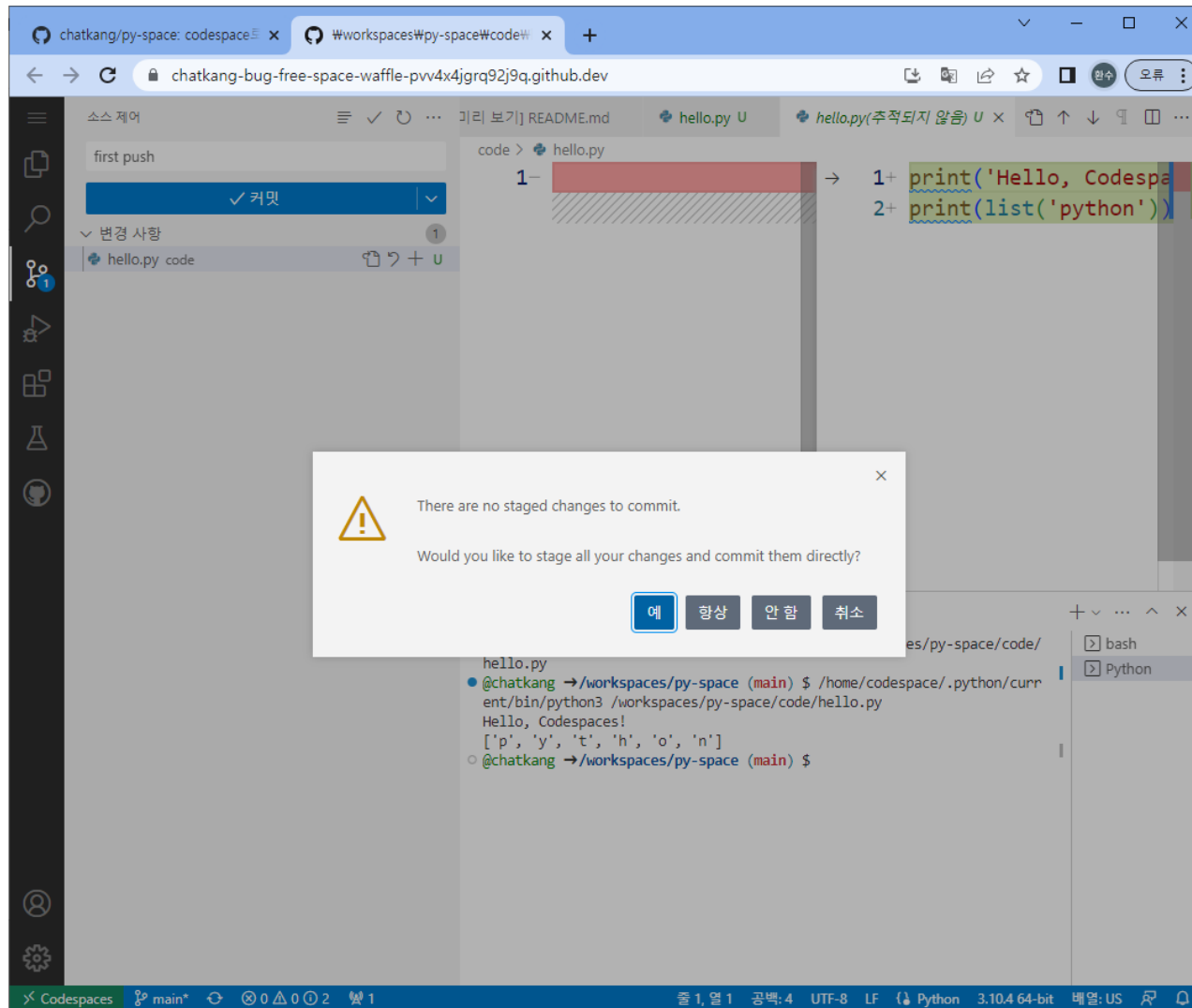
- 버전 제어 활동바 선택
- 커밋 메시지 입력 후
 - 커밋 버튼 클릭



Add와 커밋 함께

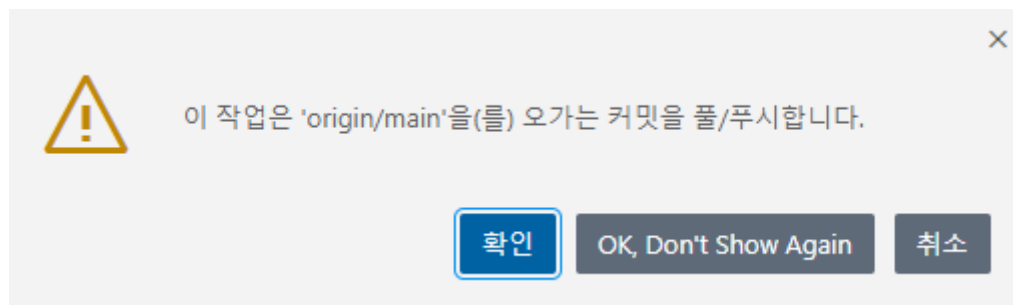
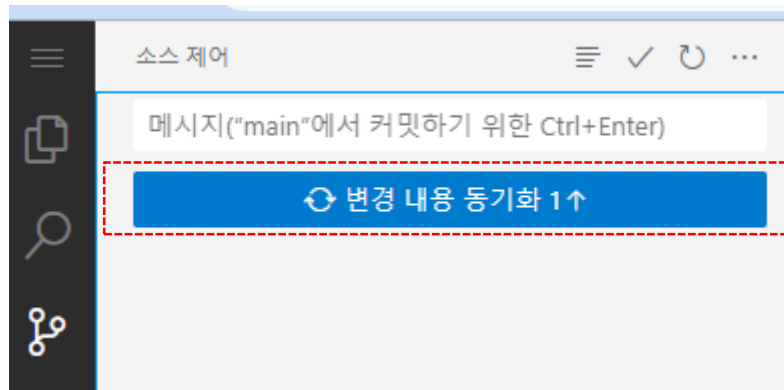
오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

- 예 선택

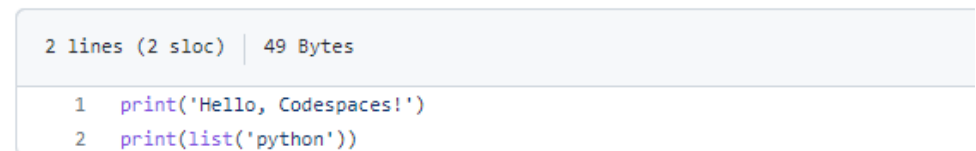
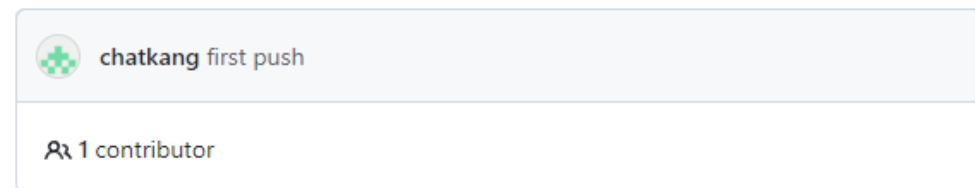
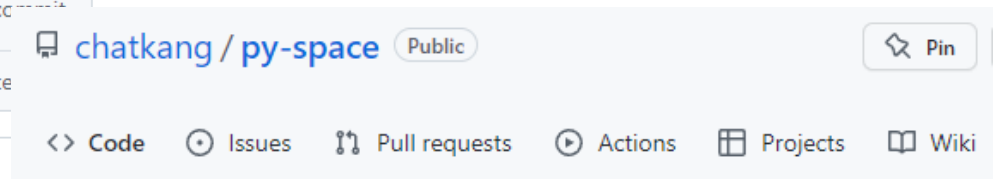
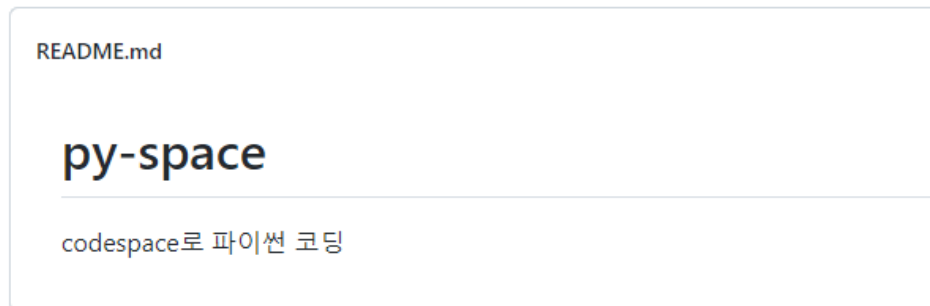
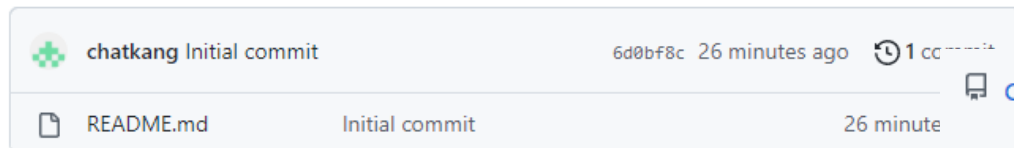
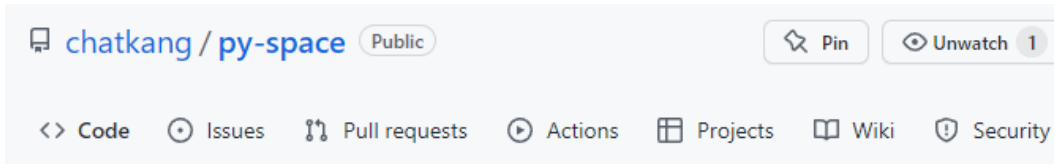


- **Push**

- 변경 내용 동기화 클릭



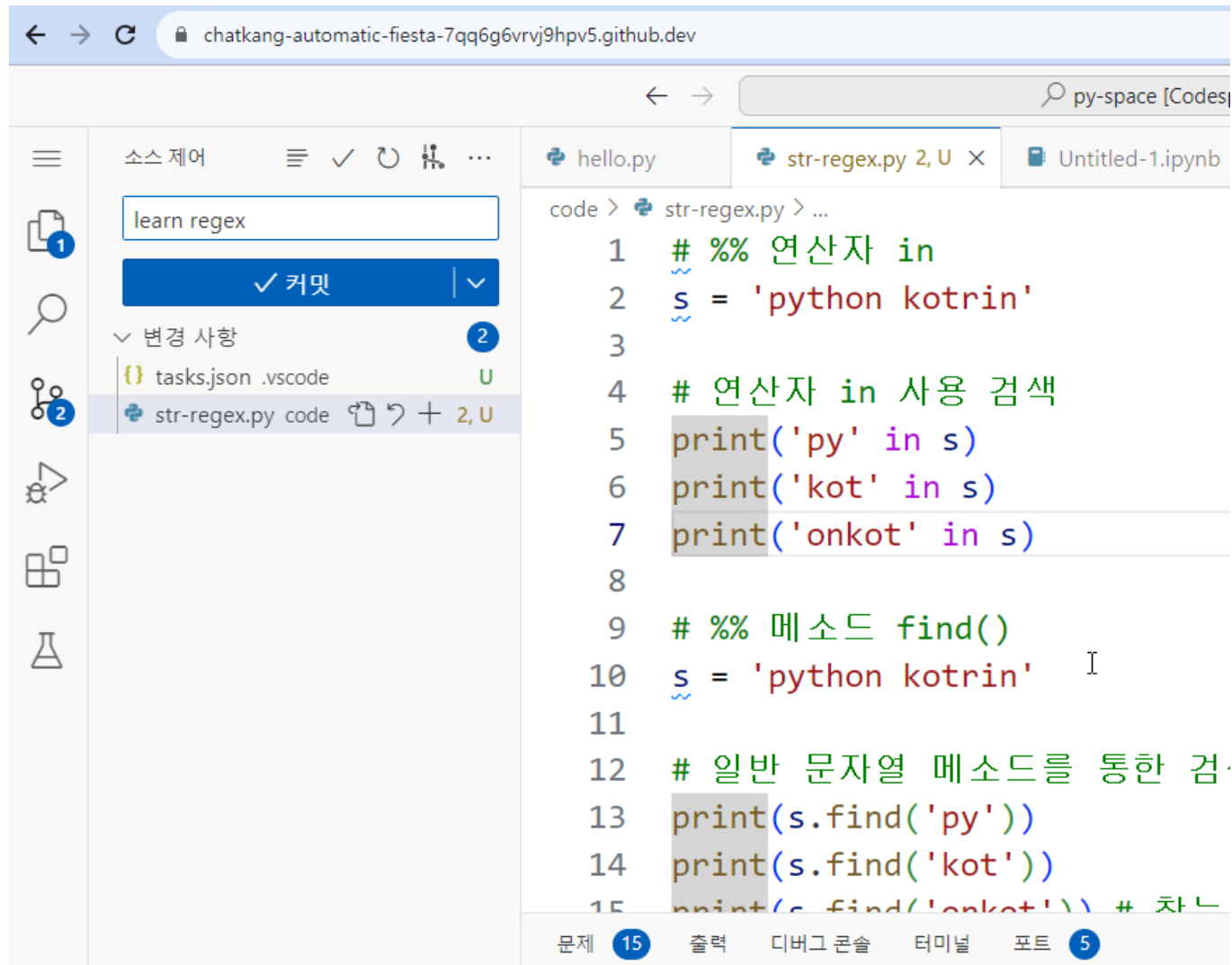
- 깃허브에서 refresh



[Give feedback](#)

다시 코드스페이스에서 코딩 후 push

오픈소스 소프트웨어를 위한 깃과 깃허브 Python language



```
# %% 연산자 in
s = 'python kotrin'

# 연산자 in 사용 검색
print('py' in s)
print('kot' in s)
print('onkot' in s)

# %% 메소드 find()
s = 'python kotrin'

# 일반 문자열 메소드를 통한 검색
print(s.find('py'))
print(s.find('kot'))
print(s.find('onkot')) # 찾는 문자열이 없으면 -1

# %% 메소드 index()
s = 'python kotrin'
print(s.index('rin'))
print(s.index('java')) # 오류 발생

# %% 모듈 re
import re # re.serarch() 사용
from re import search # serarch() 바로 사용
```

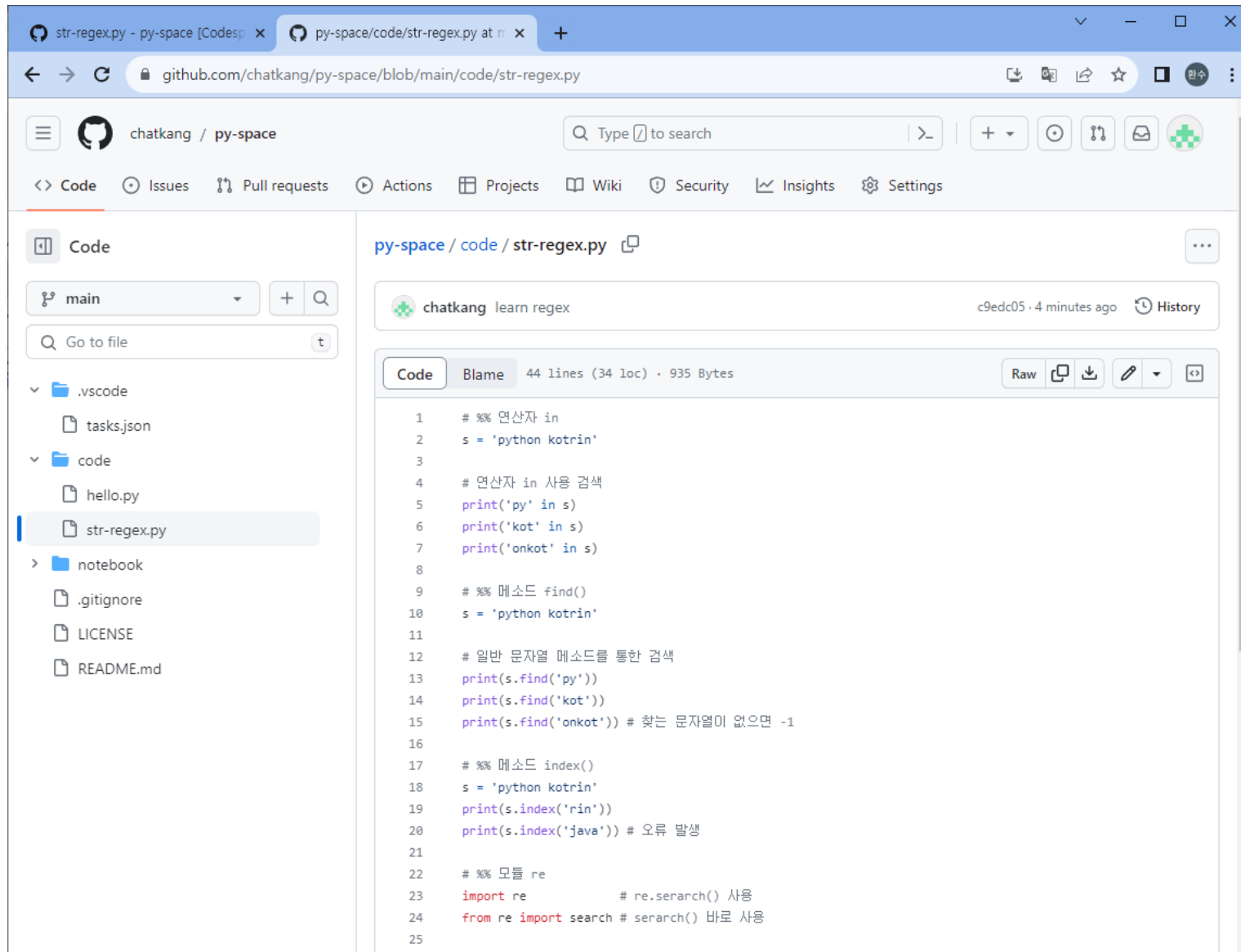
```
# %% function search
import re

m = re.search('thon', 'python kotlin')
print(type(m)) # re.Match 유형
print(m) # re.Match 객체 내용

# %% method group() span() start() end()
from re import search

s = 'python kotrin'
m = search('thon', s)
print(m)
print(m.group()) # 실제 일치하는 문자열

# 일치하는 문자열의 위치 첨자
x, y = m.start(), m.end()
print(x, y, m.span())
print(s[x:y])
```



str-regex.py - py-space [Codesp... x] py-space/code/str-regex.py at n x +

github.com/chatkang/py-space/blob/main/code/str-regex.py

chatkang / py-space

Type to search

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Code

main + Q

Go to file t

- .vscode
 - tasks.json
- code
 - hello.py
 - str-regex.py
- notebook
 - .gitignore
 - LICENSE
 - README.md

py-space / code / str-regex.py

chatkang learn regex c9edc05 · 4 minutes ago History

Code Blame 44 lines (34 loc) · 935 Bytes Raw Copy Download Edit

```
1  # %% 연산자 in
2  s = 'python kotrin'
3
4  # 연산자 in 사용 검색
5  print('py' in s)
6  print('kot' in s)
7  print('onkot' in s)
8
9  # %% 메소드 find()
10 s = 'python kotrin'
11
12 # 일반 문자열 메소드를 통한 검색
13 print(s.find('py'))
14 print(s.find('kot'))
15 print(s.find('onkot')) # 찾는 문자열이 없으면 -1
16
17 # %% 메소드 index()
18 s = 'python kotrin'
19 print(s.index('rin'))
20 print(s.index('java')) # 오류 발생
21
22 # %% 모듈 re
23 import re # re.serarch() 사용
24 from re import search # serarch() 바로 사용
25
```