

**Github**

# GitHub Action

강사 양석환



# Github Action 개요

## • 개념

- 코드 빌드, 테스트, 배포 등의 작업을 자동화할 수 있게 해 주는 Github에서 직접 제공하는 CI/CD (Continuous Integration/Continuous Delivery) 플랫폼
- Github 저장소를 기반으로 소프트웨어 개발 Workflow를 자동화 할 수 있는 도구

## • 활용

- GitHub Actions를 사용하여 Repository(저장소)에서 바로 소프트웨어 개발 워크플로를 자동화, 사용자 지정 및 실행
- CI/CD를 포함하여 원하는 작업을 수행하기 위한 작업을 검색, 생성 및 공유하고 완전히 사용자 정의된 워크플로에서 작업을 결합할 수 있음

## • 개요

- 하나 이상의 작업을 실행하는 구성 가능한 자동화된 프로세스
- 특정 순서에 따라 발생하는 반복적인 프로세스와 작업을 관리하는 시스템
  - 주로 제품 제조, 서비스 제공, 정보 처리 또는 기타 가치 창출 활동 등 업무를 수행하는 메커니즘을 가리킴
- 비즈니스 프로세스 관리에서의 워크플로우
  - 단순한 일련의 개별 작업으로 정의할 수도 있지만
  - 일반적으로 비즈니스 프로세스는 여러 워크플로우, 정보 시스템, 데이터, 인력, 인력의 활동 패턴으로 구성된 보다 복잡한 것으로 간주됨
- 단순성과 반복성이라는 특징을 가지며 일반적으로 다이어그램이나 체크리스트로 시각화 됨

- **GitHub Actions의 워크플로우(Workflow)**

- 소프트웨어 개발 프로세스를 자동화하는 데 사용되는 구성 가능한 자동화 프로세스
- 하나 이상의 작업(Job)으로 구성되며, 특정 이벤트가 발생할 때 실행됨
- YAML 파일로 정의됨
- Github 저장소에서 발생하는 build, test, package, release, deploy 등 다양한 이벤트를 기반으로 직접 원하는 Workflow를 만들 수 있음
- Runners라고 불리는 Github에서 호스팅 하는 Linux, macOS, Windows 환경에서 실행
- Runners를 사용자가 직접 호스팅하는 환경에서 직접 구동 시킬 수도 있음

## • 이벤트(Event)

- 워크플로우를 트리거하는 특정 활동
  - 예: 코드가 푸시되거나 풀 리퀘스트가 생성될 때 워크플로우가 실행될 수 있음.
- GitHub에서 발생하는 활동 뿐만 아니라, 일정에 따라 또는 REST API를 통해 수동으로 트리거할 수도 있음

## • 작업(Job)

- 동일한 러너(Runner)에서 실행되는 일련의 단계(Step)
- 각 단계는 스크립트를 실행하거나, 액션(Action)을 실행할 수 있음
- 작업은 순차적으로 또는 병렬로 실행될 수 있음

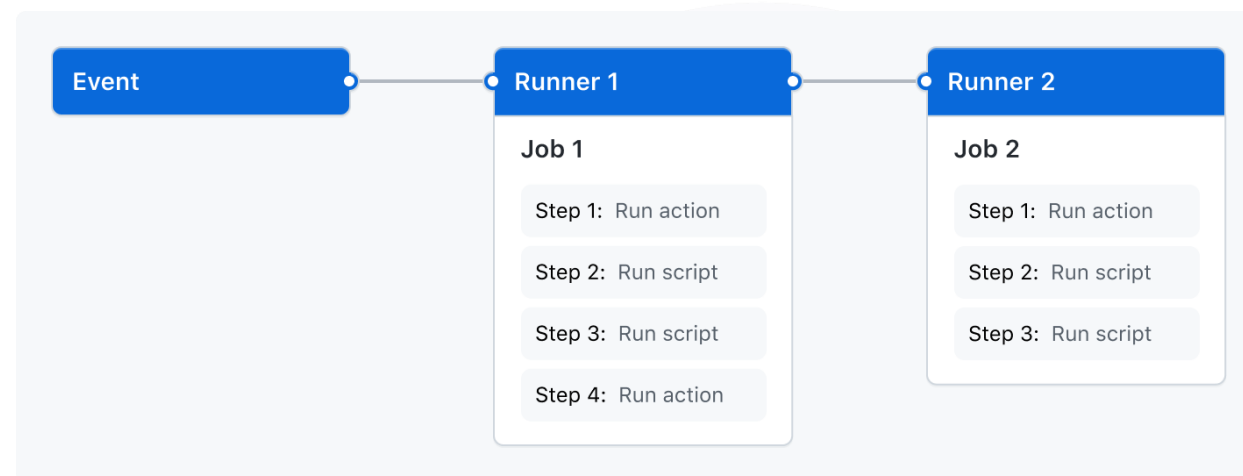


## • 단계(Step)

- 작업 내에서 실행되는 개별 명령어 또는 액션
- 각 단계는 스크립트나 커스텀 액션을 실행할 수 있음

## • 액션(Action)

- 워크플로우 내에서 재사용 가능한 확장 기능
- 커뮤니티에서 공유된 액션을 사용할 수도 있고, 직접 커스텀 액션을 만들어 사용할 수도 있음



## • Workflow 예시

- 이 예제에서는 코드가 푸시될 때마다 실행되는 워크플로우를 정의함
- build 작업은 Ubuntu 환경에서 실행되며, 코드 체크아웃, Node.js 설정, 의존성 설치, 테스트 실행의 단계를 포함

```
name: CI
on: [push]
jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v2

      - name: Set up Node.js
        uses: actions/setup-node@v2
        with:
          node-version: '14'

      - name: Install dependencies
        run: npm install

      - name: Run tests
        run: npm test
```

- **Test Code**

- 특정 함수의 return 값이 어떻게 나오는지 확인하는 테스트 코드
- df의 타입이 pd.DataFrame이 맞는가? value1에 특정 값이 들어가는가?
- 쿼리를 날리고 데이터가 맞는지 정합성 체크하는 것도 일종의 테스트

- **배포**

- 서버에 새로운 기능, 버전 등을 배포

- **기타 자동화하고 싶은 스크립트**

- 주기적으로 데이터를 수집해 처리

- **다양한 파이썬 버전에서 실행되는지 확인**



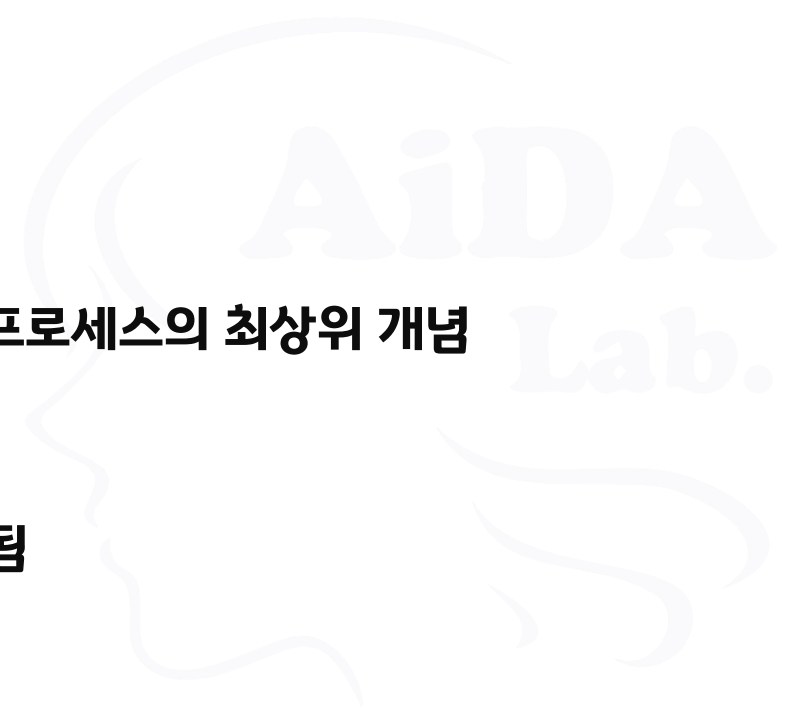
# Github Action Core 개요

- **Github Action Core**

- Github Action을 이해하기 위해서 알아야 하는 핵심 개념
- Workflow, Event, Job, Step, Action, Runner 등이 있음

- **Workflow**

- 여러 Job으로 구성되고, Event에 의해 트리거될 수 있는 자동화된 프로세스의 최상위 개념
- Workflow 파일은 YAML으로 작성됨
- Github Repository의 `.github/workflows` 폴더 아래에 저장됨



## • Event

- Workflow를 Trigger(실행)하는 특정 활동이나 규칙
- 예를 들어 다음과 같은 상황에서 사용할 수 있음
  - 특정 브랜치로 Push하거나
  - 특정 브랜치로 Pull Request하거나
  - 특정 시간대에 반복(Cron)
  - Webhook을 사용해 외부 이벤트를 통해 실행



- **Job**

- Job은 여러 Step으로 구성되고, 가상 환경의 인스턴스에서 실행됨
- 다른 Job에 의존 관계를 가질 수 있고, 독립적으로 병렬 실행도 가능함

- **Step**

- Task들의 집합으로
- 커맨드를 날리거나 action을 실행할 수 있음



## • Action

- Workflow의 가장 작은 블록(smallest portable building block)
- Job을 만들기 위해 Step들을 연결할 수 있음
- 재사용이 가능한 컴포넌트
- 개인적으로 만든 Action을 사용할 수도 있고, Marketplace에 있는 공용 Action을 사용할 수도 있음
- Github Marketplace와 Github Actions Repository에서 확인 가능

- **Runner**

- **Gitbub Action Runner 어플리케이션이 설치된 머신**
- **Workflow가 실행될 인스턴스**

- **구분**

- **Github에서 호스팅해주는 Github-hosted runner**
  - **Azure의 Standard\_DS2\_v2를 이용**
    - vCPU 2, 메모리 7GB, 임시 스토리지 14GB
- **직접 호스팅하는 Self-hosted runner**



# **Github Action 사용하기**

1. 코드 작성
2. 코드 작성 후, Workflow 정의
3. 정상적으로 작동하는지 Test



## • Workflow 정의

- 기본적인 방법 : `.github/workflows` 폴더 안에 `.yml` 파일을 생성 → 템플릿 활용 가능

## • yml 파일 예시

- Master 브랜치에 Push 또는 Pull Request가 올 경우 실행되는

### CI란 이름을 가지는 Workflow

#### name

- Workflow의 이름을 지정

#### On

- Event에 대해 작성하는 부분
- 어떤 조건에 Workflow를 Trigger 시킬지
- push(Branch or Tag), pull\_request, schedule을 사용할 수 있음
- 단, 다른 CI/CD 도구에 있는 즉시 실행 버튼은 없음(추후에 생길 수도?)
- 단일 Event를 사용할 수도 있고, array로 작성할 수도 있음

#### Jobs

- Workflow는 다양한 Job으로 구성됨. 여러 Job이 있을 경우, Default로 병렬 실행
- build라는 job을 생성하고, 그 아래에 2개의 step이 존재하는 구조
- runs-on은 어떤 OS에서 실행될지 지정
- strategy - matrix 인자를 사용하면 어떤 파이썬 버전에서 테스트할지 확인할 수 있음
- steps의 uses는 어떤 액션을 사용할지 지정함. 이미 만들어진 액션을 사용할 때 지정

```
name: CI

on:
  push:
    branches: [ master ]
  pull_request:
    branches: [ master ]

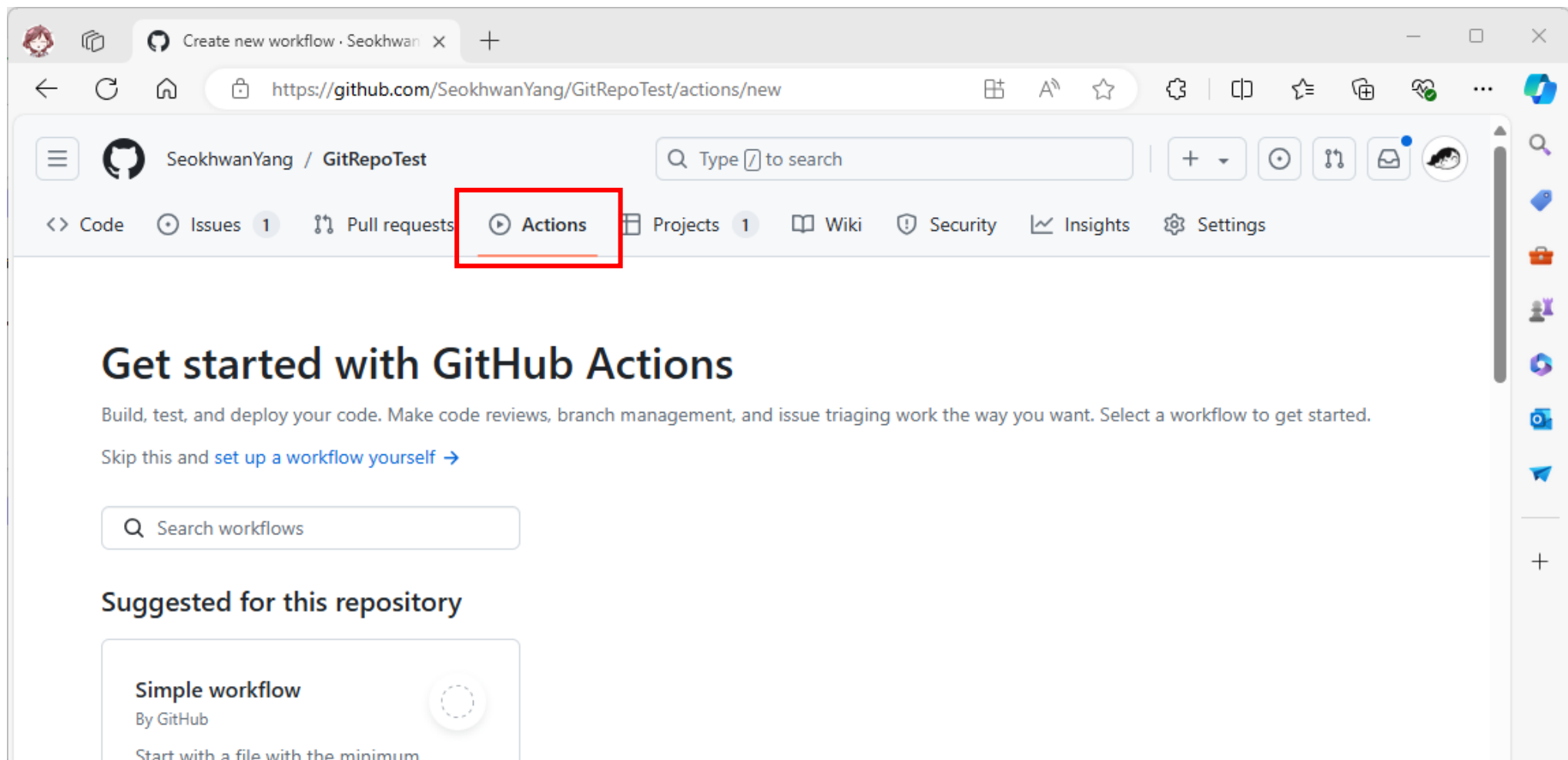
jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@v2

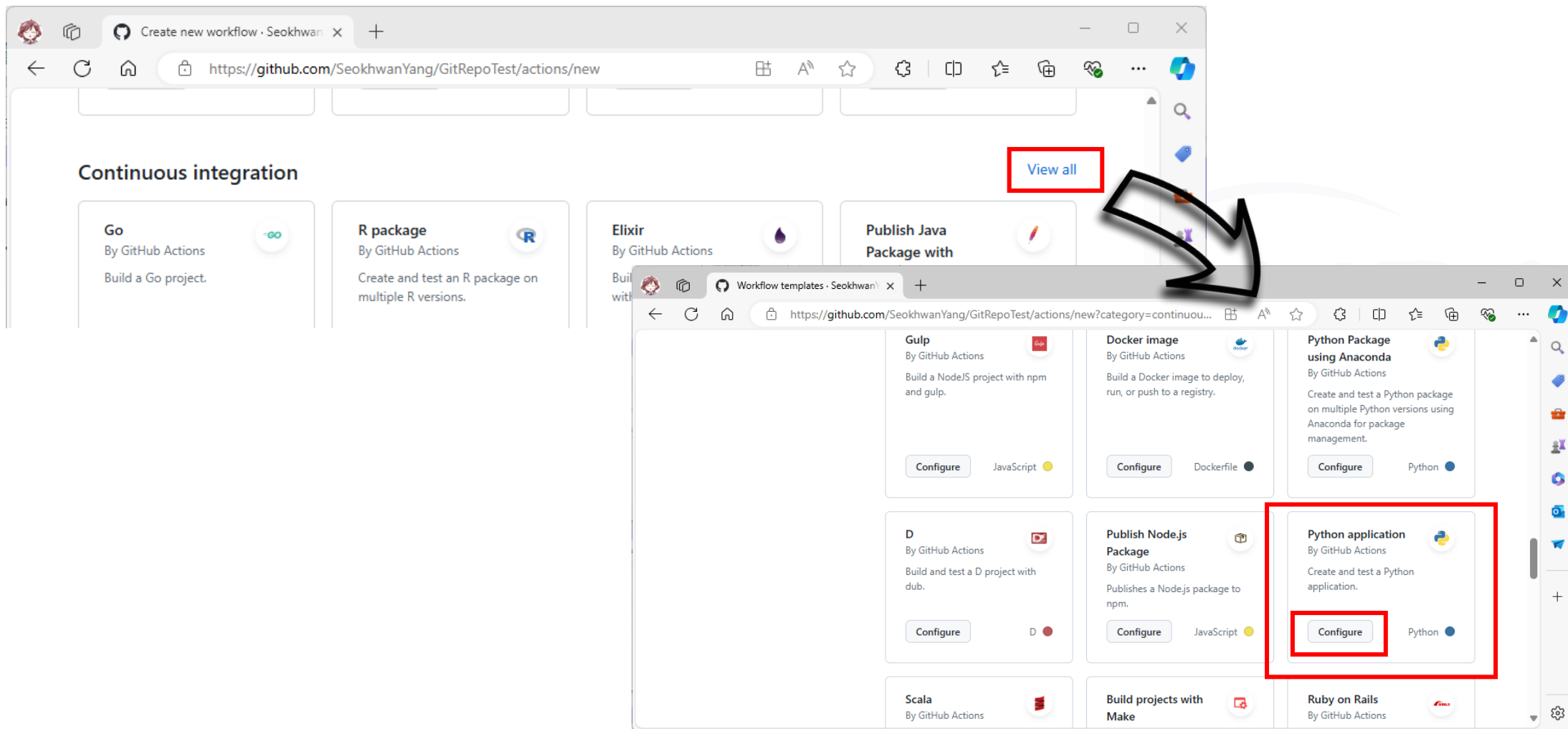
      - name: Run a one-line script
        run: echo Hello, world!

      - name: Run a multi-line script
        run: |
          echo Add other actions to build,
          echo test, and deploy your project.
```

- Github Repo에서 Actions 클릭



- 언어 등 개발환경 선택 후 **Configure** 클릭하면 간단한 workflow를 생성할 수 있음



The screenshot shows a GitHub repository page for 'SeokhwanYang / GitRepoTest'. The main content is a workflow definition file named 'python-app.yml' located in the 'main' branch. The code in the editor is as follows:

```
1 # This workflow will install Python dependencies, run tests and lint with a single version of Python
2 # For more information see: https://docs.github.com/en/actions/automating-builds-and-tests/building-and-testing-python
3
4 name: Python application
5
6 on:
7   push:
8     branches: [ "main" ]
9   pull_request:
10    branches: [ "main" ]
11
12 permissions:
13   contents: read
14
15 jobs:
16   build:
17
18     runs-on: ubuntu-latest
19
20     steps:
21     - uses: actions/checkout@v4
22     - name: Set up Python 3.10
23       uses: actions/setup-python@v3
24       with:
```

On the right side of the editor, there is a sidebar with a 'Marketplace' tab. It contains a search bar for 'Search Marketplace for Actions' and a list of 'Featured Actions':

- Upload a Build Artifact** (3k stars): Upload a build artifact that can be used by subsequent workflow steps.
- Setup Go environment** (1.4k stars): Setup a Go environment and add it to the PATH.
- Download a Build Artifact** (1.3k stars): Download a build artifact that was previously uploaded in the workflow by the upload-artifact action.
- Setup .NET Core SDK** (930 stars): Used to build and publish .NET source. Set up a

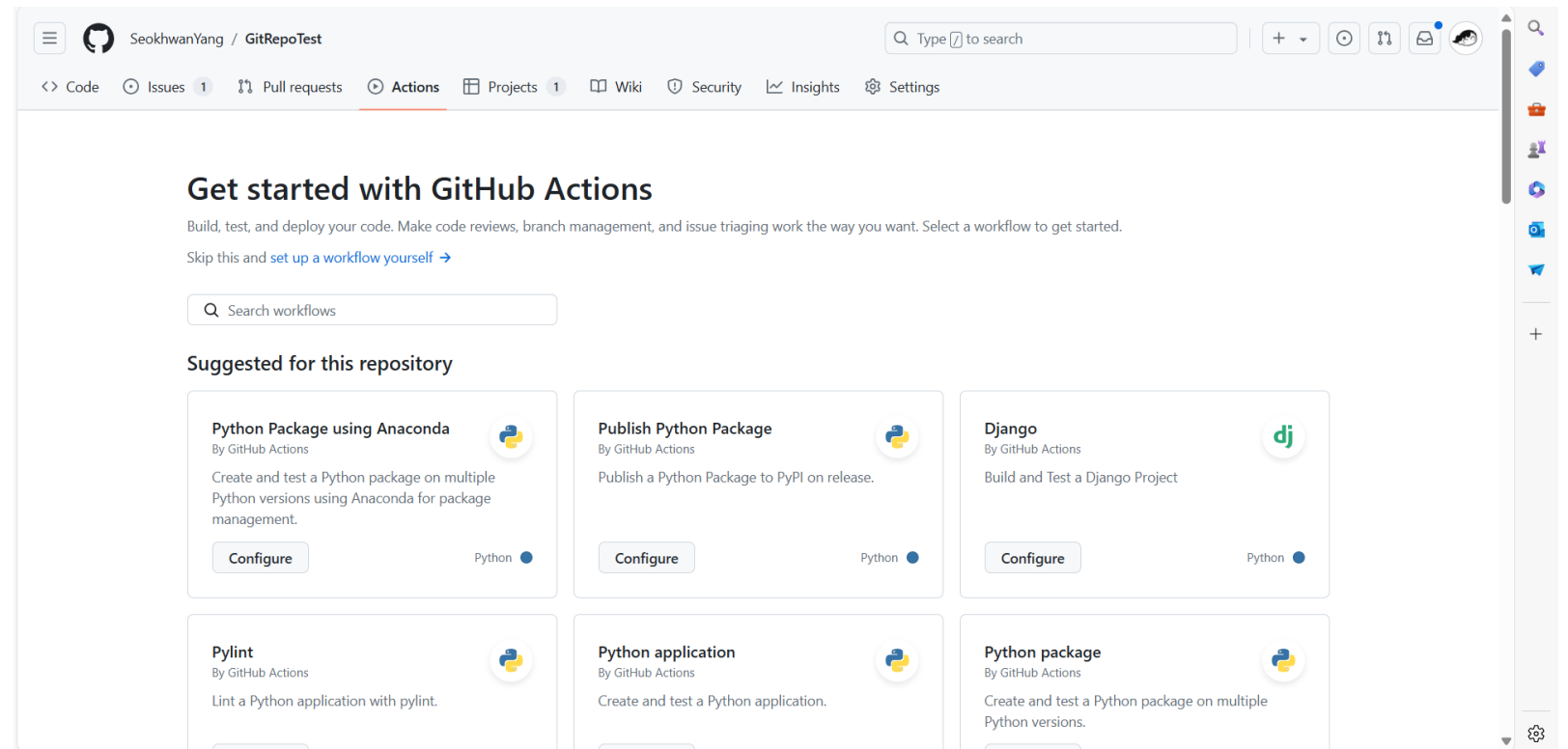
# **Github Action with Python**

- Workflow 예시 중 Python으로 작성된 것은 Python Package using Anaconda, Publish Python Package, Python application, Python package, Django, Pylint 등이 있음
- 여기서 Python package를 예시로 사용
- Github Repository 생성 후, hello.py 생성 후 Push

```
# hello.py  
print("Hello world")
```

- Github Repository에서 Actions 클릭
  - New workflow
  - Continuous integration workflows
  - 더보기
  - Python package
  - Configure

- 파일이 Push된 경우에는 Suggested for this repository와 같은 항목이 New workflow 대신 등장함
- 이 경우에는 Push된 파일의 종류에 따라 선택 항목이 반영된 카드가 제시됨



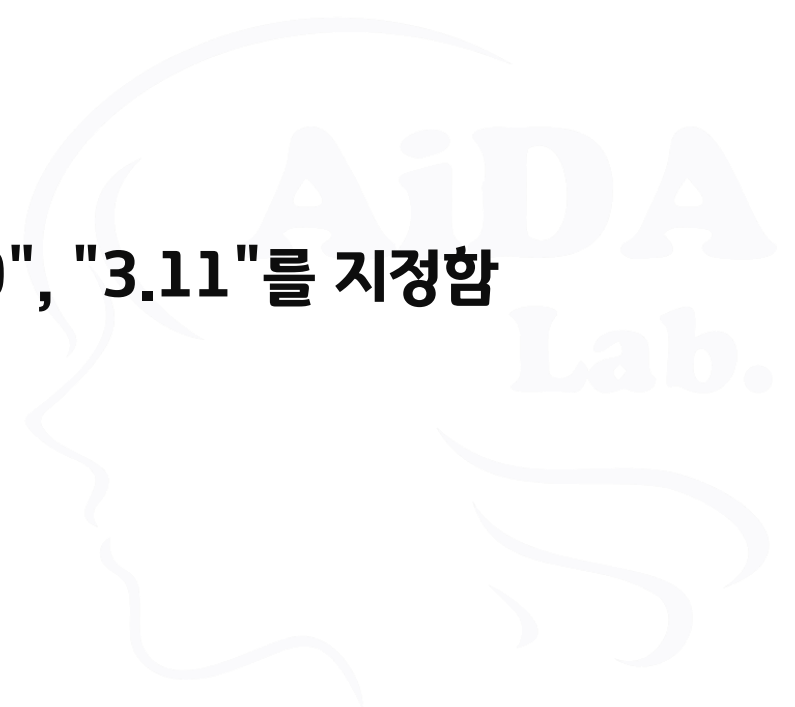
The screenshot shows a GitHub repository page for 'SeokhwanYang / GitRepoTest'. The main content is a workflow file named 'python-package.yml' in the 'main' branch. The workflow is configured to run on 'ubuntu-latest' and uses a matrix strategy to test Python versions '3.9', '3.10', and '3.11'. The workflow steps include checking out the code and installing dependencies.

```
1 # This workflow will install Python dependencies, run tests and lint with a variety of Python versions
2 # For more information see: https://docs.github.com/en/actions/automating-builds-and-tests/building-and-testing-python
3
4 name: Python package
5
6 on:
7   push:
8     branches: [ "main" ]
9   pull_request:
10    branches: [ "main" ]
11
12 jobs:
13   build:
14
15     runs-on: ubuntu-latest
16     strategy:
17       fail-fast: false
18       matrix:
19         python-version: ["3.9", "3.10", "3.11"]
20
21     steps:
22     - uses: actions/checkout@v4
```

The right sidebar shows the 'Marketplace' section with a search bar and a list of featured actions:

- Setup Node.js environment** (3.8k stars): Setup a Node.js environment by adding problem matchers and optionally downloading and adding it to the PATH.
- Upload a Build Artifact** (3k stars): Upload a build artifact that can be used by subsequent workflow steps.
- Setup Go environment** (1.4k stars): Setup a Go environment and add it to the PATH.
- Close Stale Issues** (1.3k stars): Close issues and pull requests with no recent activity.
- Setup .NET Core SDK** (932 stars): Setup the .NET Core SDK.

- 생성된 yaml 파일 아래의 pip install 부분과 pytest를 지우고
- python hello.py를 실행하도록 수정
- 템플릿 코드를 조금만 수정함
- python hello.py를 다양한 파이썬 버전에서 실행함
- strategy: matrix: python-version에서 "3.9", "3.10", "3.11"를 지정함
- 수정 후 “Commit Change” 실행



```
Edit Preview Code 55% faster with GitHub Copilot Spaces 2 No wrap
```

```
14
15 runs-on: ubuntu-latest
16 strategy:
17   fail-fast: false
18   matrix:
19     python-version: ["3.9", "3.10", "3.11"]
20
21 steps:
22 - uses: actions/checkout@v4
23 - name: Set up Python ${{ matrix.python-version }}
24   uses: actions/setup-python@v3
25   with:
26     python-version: ${{ matrix.python-version }}
27 - name: Install dependencies
28   run: |
29     python -m pip install --upgrade pip
30     python -m pip install flake8 pytest
31     if [ -f requirements.txt ]; then pip install -r requirements.txt; fi
32 - name: Lint with flake8
33   run: |
34     # stop the build if there are Python syntax errors or undefined names
35     flake8 . --count --select=E9,F63,F7,F82 --show-source --statistics
36     # exit-zero treats all errors as warnings. The GitHub editor is 127 chars wide
37     flake8 . --count --exit-zero --max-complexity=10 --max-line-length=127 --statistics
38 - name: Test with pytest
39   run: |
40     pytest
41
```

여러 버전으로 실행하도록 전략 설정

종속성에 따른 설치 작업 제거

이것도 지우자...

Pytest 대신 hello.py를 실행하도록 수정

```
name: Python package
on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]
jobs:
  build:
    runs-on: ubuntu-latest
    strategy:
      fail-fast: false
      matrix:
        python-version: ["3.9", "3.10", "3.11"]
    steps:
      - uses: actions/checkout@v4
      - name: Set up Python ${{ matrix.python-version }}
        uses: actions/setup-python@v3
        with:
          python-version: ${{ matrix.python-version }}
      - name: Run Python
        run: |
          python hello.py
```

The screenshot shows a web browser window displaying a GitHub repository page for 'SeokhwanYang / GitRepoTest'. The URL is `https://github.com/SeokhwanYang/GitRepoTest/new/main?filename=.github%2Fworkflows%2Fpython-package.yml&workflow_template=ci%2F...`. The page shows the workflow file `python-package.yml` in the `main` branch. The workflow content is as follows:

```
1 # This workflow will install Python dependencies, run tests and lint with a variety of Python versions
2 # For more information see: https://docs.github.com/en/actions/automating-builds-and-tests/building-and-testing-python
3
4 name: Python package
5
6 on:
7   push:
8     branches: [ "main" ]
9   pull_request:
10    branches: [ "main" ]
11
12 jobs:
13   build:
14
15     runs-on: ubuntu-latest
16     strategy:
17       fail-fast: false
18       matrix:
19         python-version: ["3.9", "3.10", "3.11"]
20
21     steps:
22     - uses: actions/checkout@v4
```

The `Commit changes...` button is highlighted with a red box. The page also features a sidebar with 'Featured Actions' and a footer with navigation instructions.

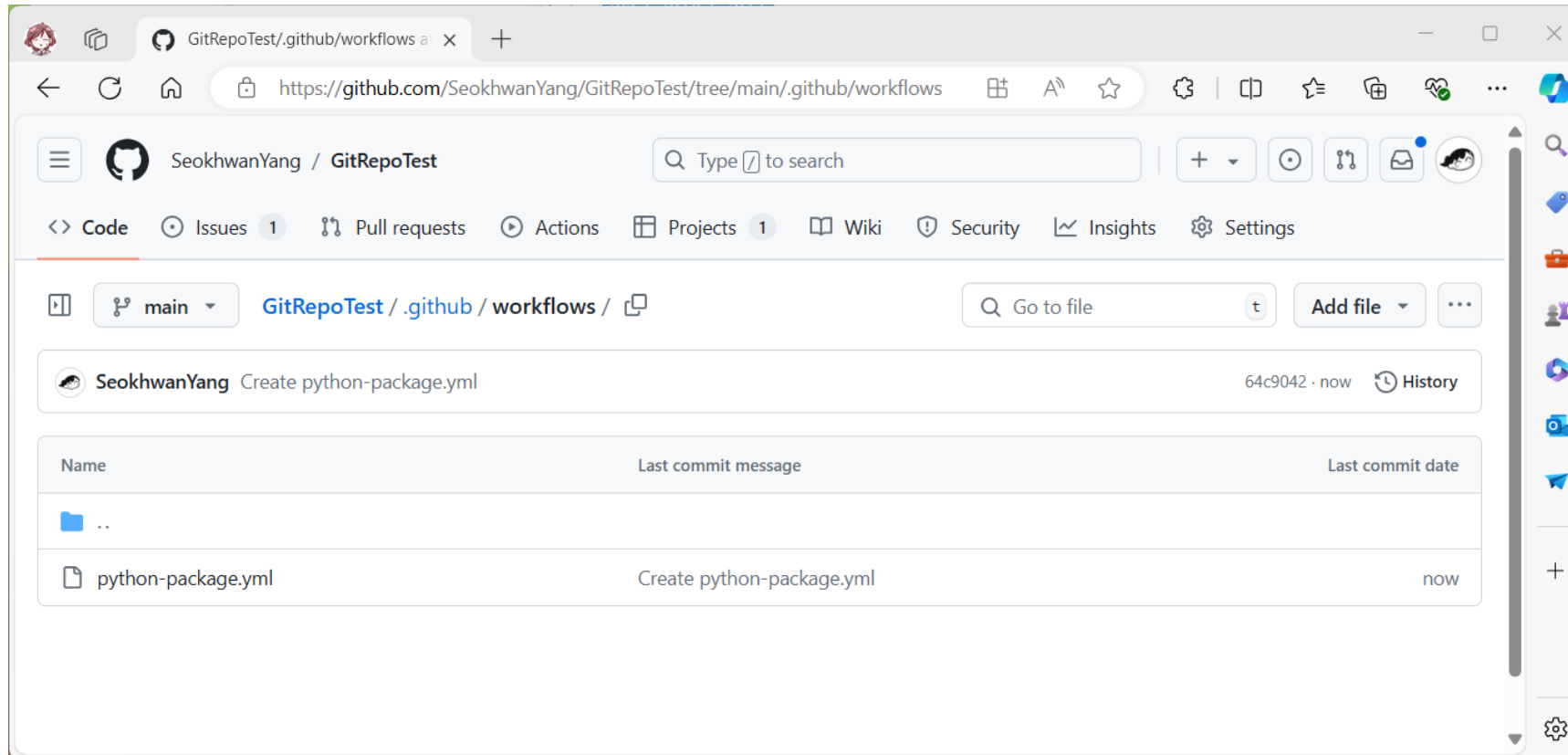
The screenshot shows a GitHub repository page for 'SeokhwanYang / GitRepoTest'. The file being edited is `python-package.yml` in the `.github/workflows` directory. The workflow content is as follows:

```
1 # This workflow will install Python dependencies, run tests and
2 # For more information see: https://docs.github.com/en/actions/a
3
4 name: Python package
5
6 on:
7   push:
8     branches: [ "main" ]
9   pull_request:
10    branches: [ "main" ]
11
12 jobs:
13   build:
14
15     runs-on: ubuntu-latest
16     strategy:
17       fail-fast: false
18       matrix:
19         python-version: ["3.9", "3.10", "3.11"]
20
21     steps:
22     - uses: actions/checkout@v4
```

A 'Commit changes' dialog box is open, showing the commit message 'Create python-package.yml' and the commit email 'yangseokhwan@naver.com'. The 'Commit directly to the main branch' option is selected. A red box highlights the 'Commit changes' button. The extended description field contains the Korean text '내용을 적절히 작성' (Write the content appropriately).

On the right side of the page, there is a 'Featured Actions' section with the following items:

- Setup Node.js environment (3.8k stars)
- Upload a Build Artifact (3k stars)
- Setup Go environment (1.4k stars)
- Close Stale Issues (1.3k stars)
- Setup .NET Core SDK (932 stars)



SeokhwanYang / GitRepoTest

Search: Type / to search

Code Issues 1 Pull requests Actions Projects 1 Wiki Security Insights Settings

GitRepoTest Public

Pin Unwatch 1 Fork 0 Star 0

main 4 Branches 0 Tags

Go to file Add file Code

File	Commit	Time
SeokhwanYang Update python-package.yml	5aca448 · now	9 Commits
.github/workflows	Update python-package.yml	now
README.md	Pull Request 실습	2 days ago
Readme02.md	Create Readme02.md	2 days ago
curr001.txt	Pull Request 실습	2 days ago
curr002.txt	Pull Request 실습	2 days ago
hello.py	Github Actions Test	1 hour ago

작업중이라는 표시

클릭하면 작업상태를 확인할 수 있음

About: No description, website, or topics provided.

Releases: No releases published. [Create a new release](#)

Packages: No packages published. [Publish your first package](#)

SeokhwanYang / GitRepoTest

Code Issues 1 Pull requests Actions Projects 1 Wiki Security Insights Settings

GitRepoTest Public

main 4 Branches 0 Tags

Go to file Add file Code

SeokhwanYang Create python package.yml 24c9042 · 3 minutes ago 8 Commits

File	Commit Message	Time
.github/workflows	Create python-package.yml	3 minutes ago
README.md	Pull Request 실습	2 days ago
Readme02.md	Create Readme02.md	2 days ago
curr001.txt	Pull Request 실습	2 days ago
curr002.txt	Pull Request 실습	2 days ago
hello.py	Github Actions Test	1 hour ago

README

## GitRepoTest

Test Curr 브랜치에서의 수정 내용

About: No description, website, or topics provided.

Releases: No releases published. [Create a new release](#)

Packages: No packages published. [Publish your first package](#)

Languages: Python 100.0%

Commits · SeokhwanYang/GitRepoTest

main

Commits on Jul 28, 2024

- Create python-package.yml** (Verified) 64c9042  
SeokhwanYang committed 5 minutes ago · 0 / 3
- Github Actions Test** 2178f2e  
SeokhwanYang committed 1 hour ago

Commits on Jul 26, 2024

- Merge pull request #2 from SeokhwanYang/curr** (Verified) cbff84a  
SeokhwanYang committed 2 days ago
- Merge pull request #3 from SeokhwanYang/sub01** (Verified) 88b577d  
SeokhwanYang committed 2 days ago
- Pull Request 실습** cec09be  
SeokhwanYana committed 2 days ago

*Workflow 커밋 이력*

The screenshot shows the GitHub interface for the repository 'SeokhwanYang / GitRepoTest'. The 'Commits' section is active, showing a list of commits on the 'main' branch. The first commit is 'Create python-package.yml' by SeokhwanYang, committed 7 minutes ago. A red box highlights the three-dot menu icon for this commit. A red arrow points from this menu to the 'View checks' option in the dropdown menu. Handwritten Korean text '상태 체크' (Status Check) is written next to the menu icon. Other commits include 'Github Actions Test', 'Merge pull request #2 from SeokhwanYang/curr', 'Merge pull request #3 from SeokhwanYang/sub01', and 'Pull Request 실습'.

Commits

main

All users All time

Commits on Jul 28, 2024

Create python-package.yml  
SeokhwanYang committed 7 minutes ago · 0 / 3

Github Actions Test  
SeokhwanYang committed 1 hour ago

Commits on Jul 26, 2024

Merge pull request #2 from SeokhwanYang/curr  
SeokhwanYang committed 2 days ago

Merge pull request #3 from SeokhwanYang/sub01  
SeokhwanYang committed 2 days ago

Pull Request 실습  
SeokhwanYana committed 2 days ago

Verified 64c9042

View commit details  
Copy full SHA for 64c9042  
Browse repository at this point in time  
View checks

상태 체크

The screenshot shows a GitHub repository page for 'SeokhwanYang / GitRepoTest'. The browser address bar shows the URL 'https://github.com/SeokhwanYang/GitRepoTest/commits/main/'. The repository name and a search bar are visible at the top. Below the repository name, there are navigation links for Code, Issues (1), Pull requests, Actions, Projects (1), Wiki, Security, Insights, and Settings. The main content area is titled 'Commits' and shows a commit history for the 'main' branch. A modal window is open over the commit history, displaying the message 'All checks have failed' and listing three failing checks: 'Python package / build (3.9) (push)', 'Python package / build (3.10) (push)', and 'Python package / build (3.11) (push)'. The 'Details' link for the first failing check is highlighted with a red box. The commit history shows several commits, including 'Create python-package.yml', 'Github Actions Test', 'Merge pull request #2 from SeokhwanYang/curr', 'Merge pull request #3 from SeokhwanYang/sub01', and 'Pull Request 실습'. Each commit entry includes the commit message, the author, the commit time, and a 'Verified' badge with a commit hash and icons for cloning and viewing the commit details.

Commits

main

Commits on Jul 28, 2024

Create python-package.yml

SeokhwanYang committed 8 minutes ago · X 0 / 3

Python package / build (3.9) (push) Failing after 10s

Python package / build (3.10) (push) Failing after 6s

Python package / build (3.11) (push) Failing after 5s

Commits on Jul 26, 2024

Merge pull request #2 from SeokhwanYang/curr

SeokhwanYang committed 2 days ago

Merge pull request #3 from SeokhwanYang/sub01

SeokhwanYang committed 2 days ago

Pull Request 실습

SeokhwanYana committed 2 days ago

All checks have failed

3 failing checks

Details

Verified 64c9042

Verified 2178f2e

Verified cbff84a

Verified 88b577d

cec09be

The screenshot shows a GitHub Actions workflow run for the job 'Update python-package.yml #2'. The workflow is in a failed state, indicated by a red 'x' icon. The left sidebar shows the workflow file and a list of jobs, with 'build (3.9)' selected. The main panel displays the logs for the 'build (3.9)' job, which failed 2 minutes ago in 3 seconds. The logs show the following steps:

- Set up job (1s)
- Run actions/checkout@v4 (1s)
- Set up Python 3.9 (0s)
- Run Python (0s) - Failed
- Post Set up Python 3.9 (0s)
- Post Run actions/checkout@v4 (0s)
- Complete job (0s)

The error message for the 'Run Python' step is:

```
1 ▶ Run python3 hello.py
7   File "/home/runner/work/GitRepoTest/GitRepoTest/hello.py", line 3
8
9   ^
10  SyntaxError: unexpected EOF while parsing
11  Error: Process completed with exit code 1.
```

# Github Action 실습

## • 작업 순서

1. Github에 저장소 만들기
2. 파이썬 가상환경 만들기
3. Git 초기화
4. Git 원격 저장소 설정하기
5. 파이썬 가상환경 활성화 및 Library 설치
6. 코드 작업
7. Git Push
8. Actions 사용
9. Git Pull

### 명령어 이력

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

```
git config user.name "SeokhwanYang"
git config user.email "yangseokhwan@naver.com"
```

```
git remote add origin https://github.com/SeokhwanYang/actions.git
git pull origin main
git status
```

```
source ./bin/activate
```

```
pip install --upgrade pip
pip install pytest
```

[ test\_action.py 만들기 ]

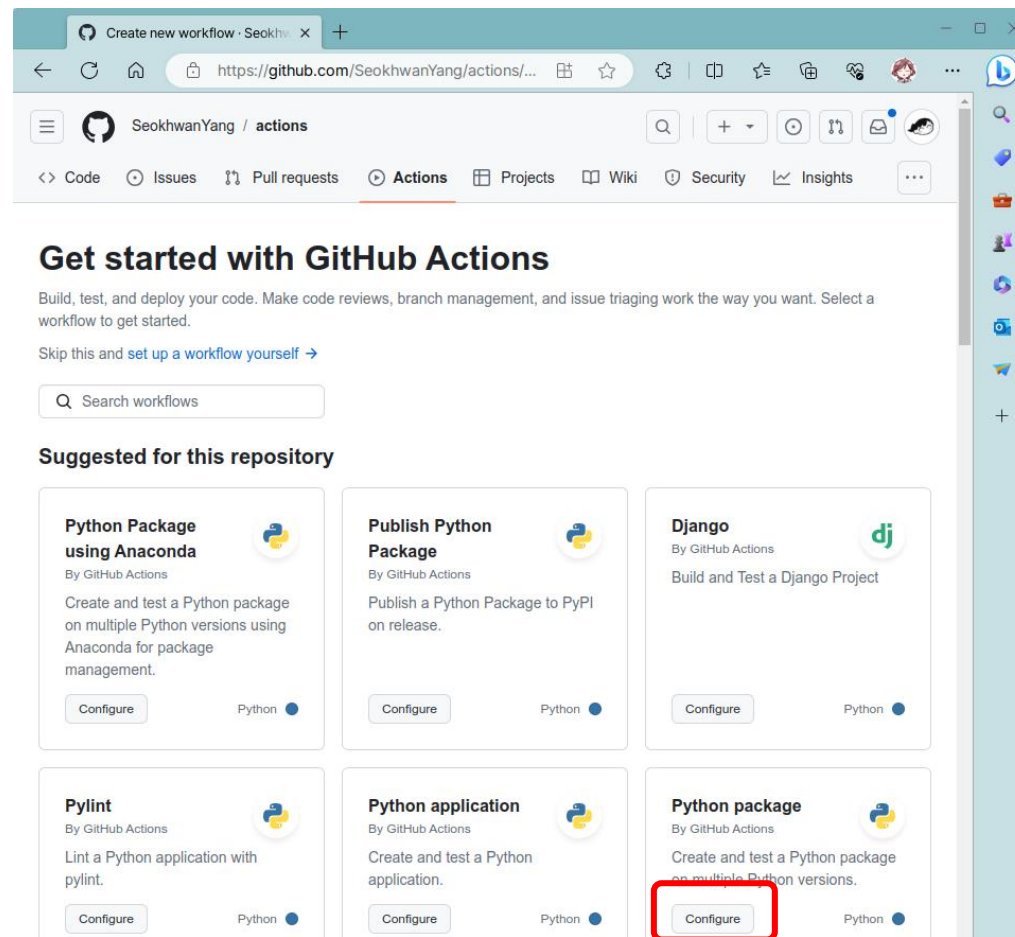
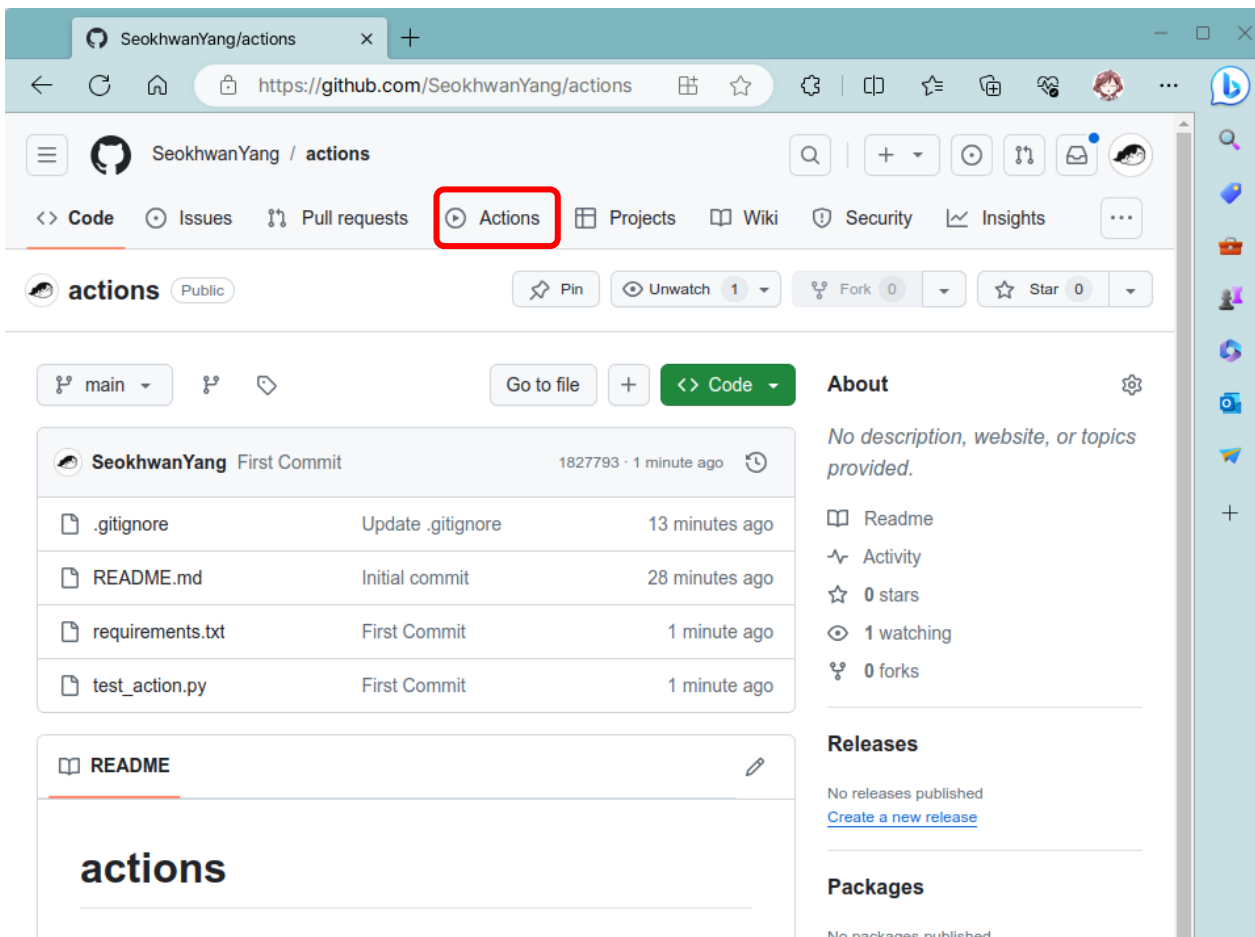
```
python -m pytest -s -v
pip freeze > requirements.txt
```

```
git status
git add .
git commit -m "First Commit"
git push origin main
```

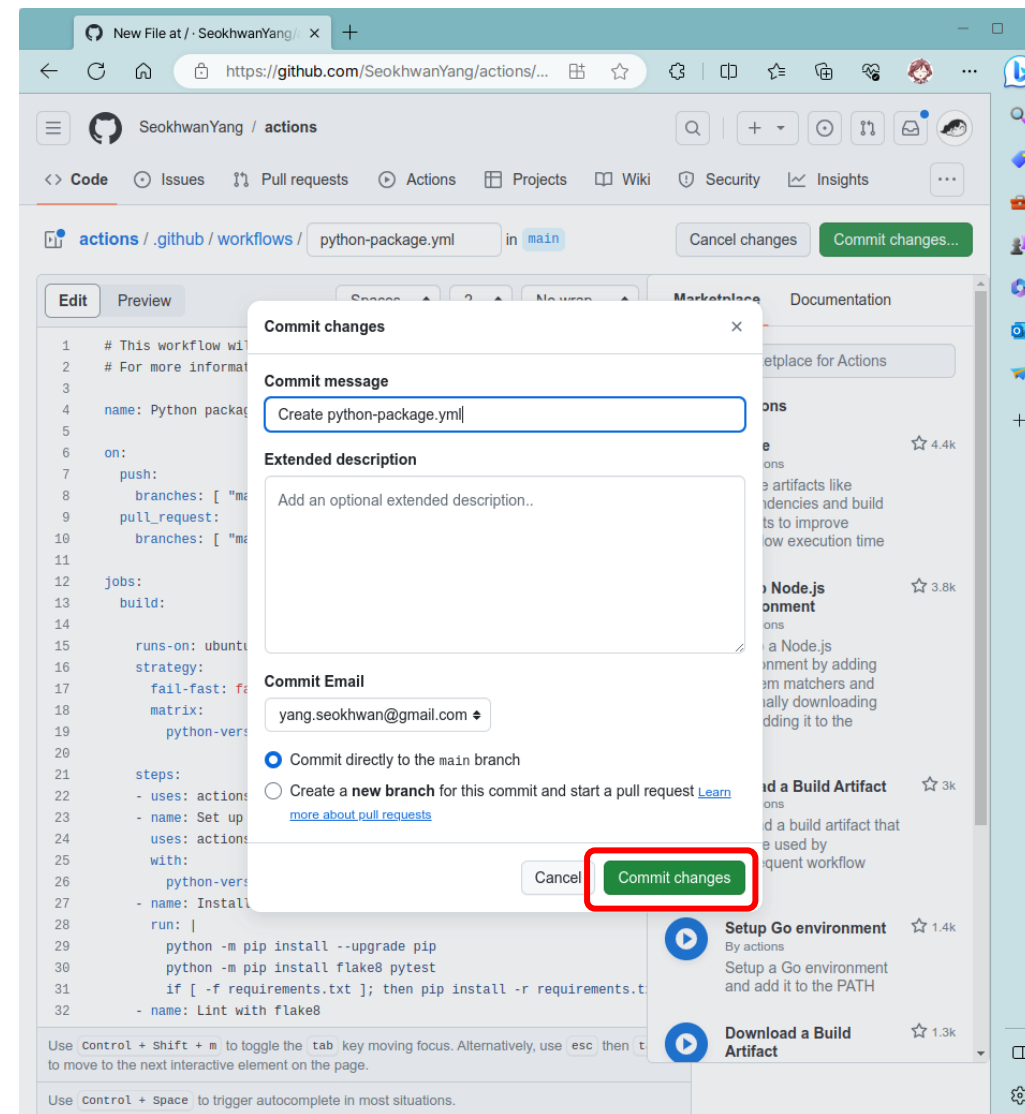
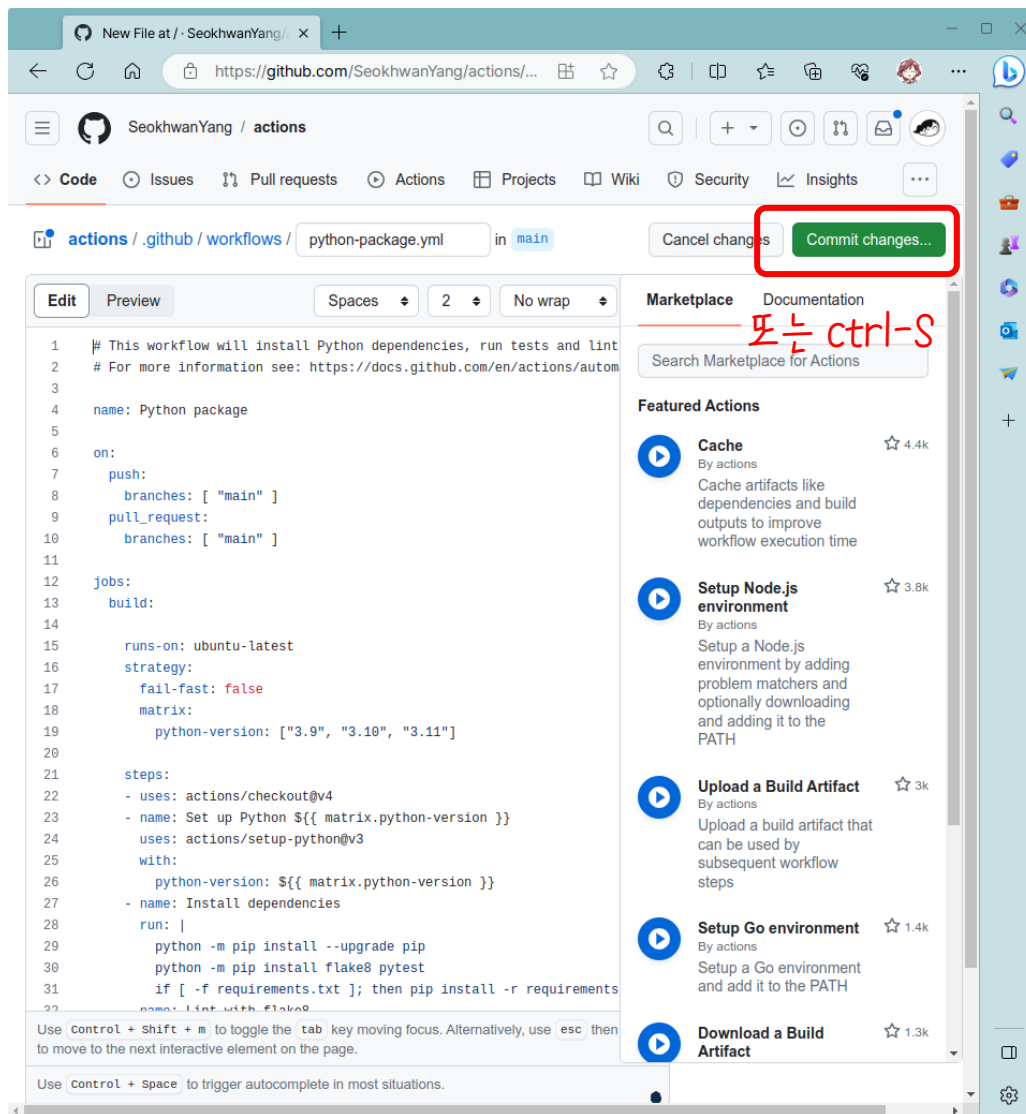
[ Github Site에서 Actions 사용 ]

```
git pull origin main
```

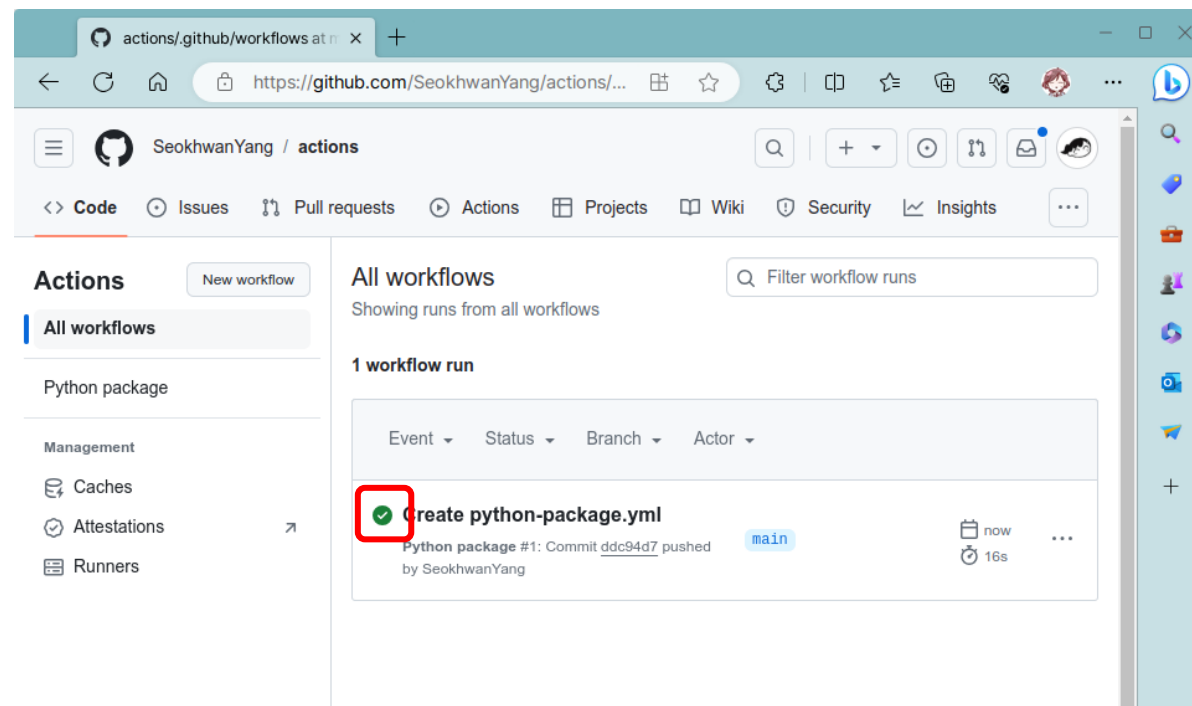
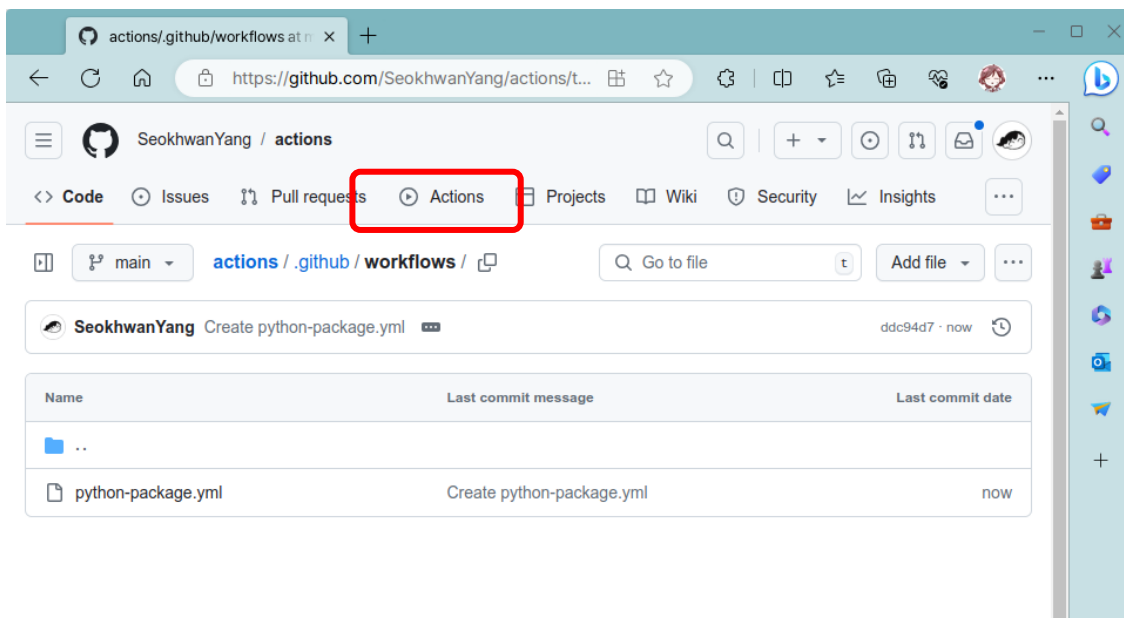
## • Github Actions 작업 내용



# Github Action 기본 설정을 이용하여 Test 실행 시키기



# Github Action 기본 설정을 이용하여 Test 실행 시키기



# Github Action 기본 설정을 이용하여 Test 실행 시키기

The screenshot shows the GitHub Actions interface for a workflow named 'Create python-package.yml #1'. The workflow is in a 'Success' state, triggered by a push to the 'main' branch. The total duration is 16s. The 'Summary' section shows three jobs, all of which are 'build' jobs for versions 3.9, 3.10, and 3.11, all of which are completed. The 'Run details' section shows the workflow file and a matrix build with 3 jobs completed. The 'Annotations' section shows three warnings, each related to a deprecated Node.js version (3.10, 3.11, and 3.9) used in the build jobs.

The screenshot shows the 'All workflows' page for the repository. It displays two workflow runs for the 'python-package' workflow. The first run is 'Update python-package.yml', which is currently in progress. The second run is 'Create python-package.yml', which is completed and took 16s. The page also shows a search bar for workflow runs and a list of workflow runs with columns for Event, Status, Branch, and Actor.

## • 주기적으로 파이썬 실행 시키기

### 기본 YML 파일

```
name: Python package

on:
  push:
    branches: [ "main" ]
jobs:
  build:
    runs-on: ubuntu-latest
    strategy:
      fail-fast: false
      matrix:
        python-version: ["3.9", "3.10", "3.11"]
    steps:
      - uses: actions/checkout@v4
      - name: Set up Python ${ matrix.python-version }
        uses: actions/setup-python@v3
        with:
          python-version: ${ matrix.python-version }
      - name: Install dependencies
        run: |
          python -m pip install --upgrade pip
          python -m pip install flake8 pytest
          if [ -f requirements.txt ]; then pip install -r requirements.txt; fi
      - name: Test with pytest
        run: pytest
```

### 5분마다 실행

```
name: Python package

on:
  push:
    branches: [ "main" ]
  schedule:
    - cron: '0 0/5 * * *'

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - name: Set up Python 3.11
        uses: actions/setup-python@v3
        with:
          python-version: "3.11"
      - name: Install dependencies
        run: |
          python -m pip install --upgrade pip
          python -m pip install flake8 pytest
          if [ -f requirements.txt ]; then pip install -r requirements.txt; fi
      - name: Test with pytest
        run: pytest
```

## • Cron 표현식

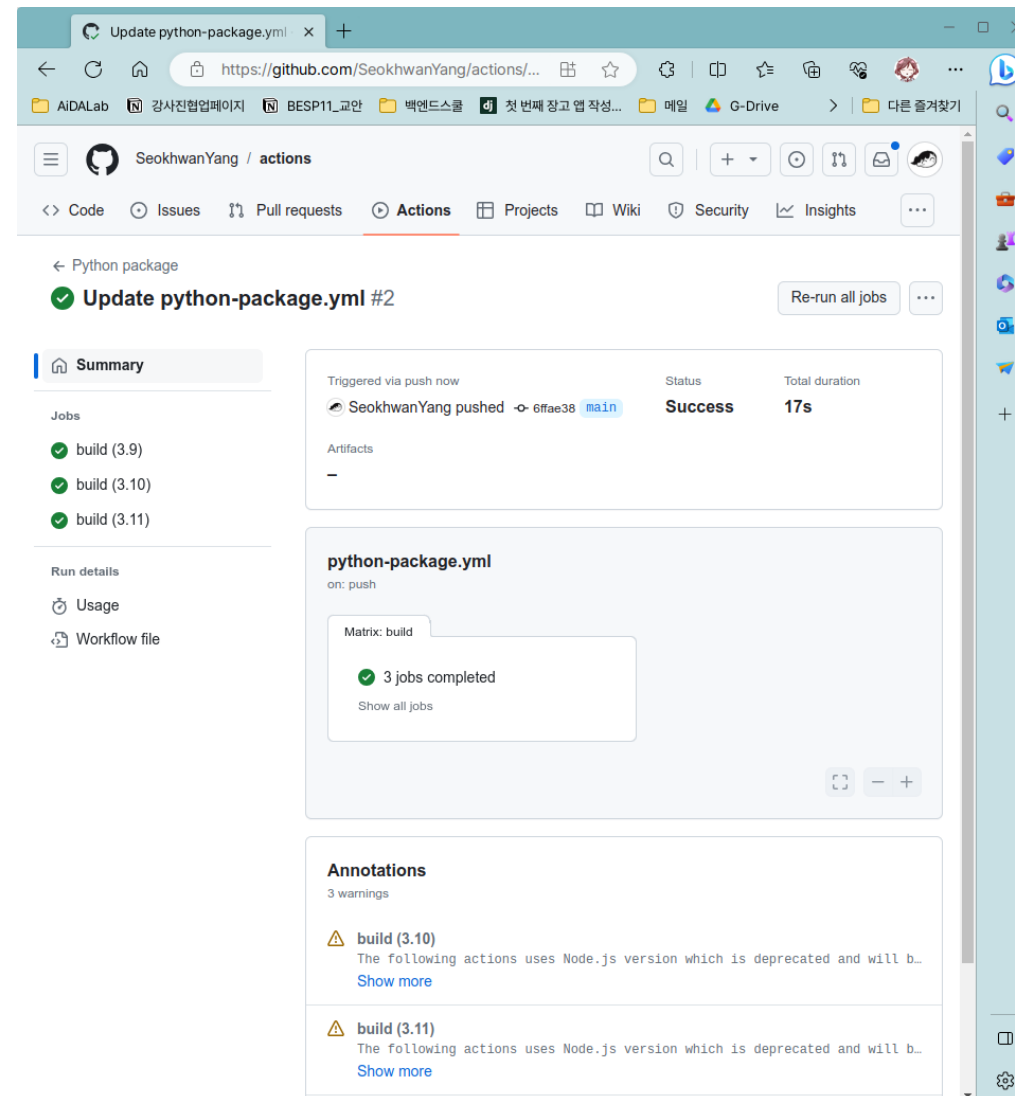
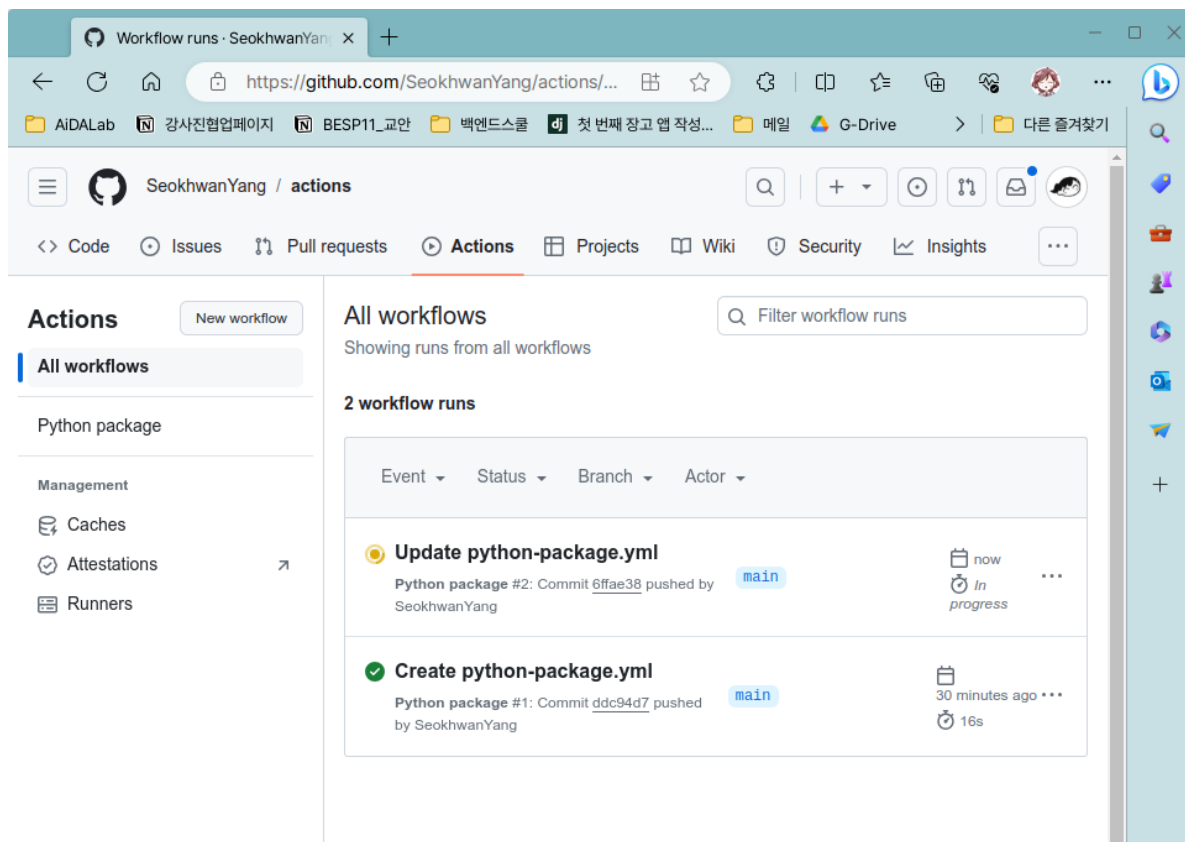
### • 필드 값

필드명	값의 허용 범위	허용된 특수문자
초 (Seconds)	0 ~ 59	, - * /
분 (Minutes)	0 ~ 59	, - * /
시 (Hours)	0 ~ 23	, - * /
일 (Day)	1 ~ 31	, - * ? / L W
월 (Month)	1 ~ 12 or JAN ~ DEC	, - * /
요일 (Week)	0 ~ 6 or SUN ~ SAT (7도 일요일)	, - * ? / L #
연도 (Year)	empty or 1970 ~ 2099	, - * /

Cron 표현식은 7개의 필드로 구성된 문자열, 각 필드는 공백으로 구분함

### • 특수문자

- \* : 모든 값
- ? : 특정한 값이 없음
- - : 범위 (예: 월요일에서 수요일까지는 MON-WED로 표현)
- , : 특별한 값일 때만 동작 (예: 월,수,금 MON,WED,FRI 등)
- / : 시작시간 / 단위 (예: 0분부터 매 5분 0/5)
- L : 일에서 사용하면 마지막 일, 요일에서는 마지막 요일(토)
- W : 가장 가까운 평일 (예: 15W: 15일에서 가장 가까운 평일 (월~금))
- # : 몇째주의 무슨 요일을 표현 (예: 3#2 : 2번째주 수요일)



## • Index.html 파일의 시각 갱신하기

1. Update\_time.py 파일 만들기
2. Update\_test\_html.yml 파일 만들기
3. Git Push
4. Actions 확인

```
name: Update index.html with current time
```

```
on:
```

```
  push:
```

```
    branches: [ main ]
```

```
permissions:
```

```
  contents: write
```

```
jobs:
```

```
  build:
```

```
    runs-on: ubuntu-latest
```

```
    steps:
```

```
      - uses: actions/checkout@v4
```

```
      - name: Set up Python 3.11
```

```
        uses: actions/setup-python@v3
```

```
        with:
```

```
          python-version: 3.11
```

```
      - name: Install dependencies
```

```
        run: |
```

```
          python -m pip install --upgrade pip
```

```
          python -m pip install flake8 pytest
```

```
          if [ -f requirements.txt ]; then pip install -r requirements.txt; fi
```

```
      - name: Run Python script # 파이썬 코드 실행
```

```
        run: python update_time.py
```

```
      - name: Commit updated index.html # 변경된 index.html 파일 커밋
```

```
        uses: EndBug/add-and-commit@v7
```

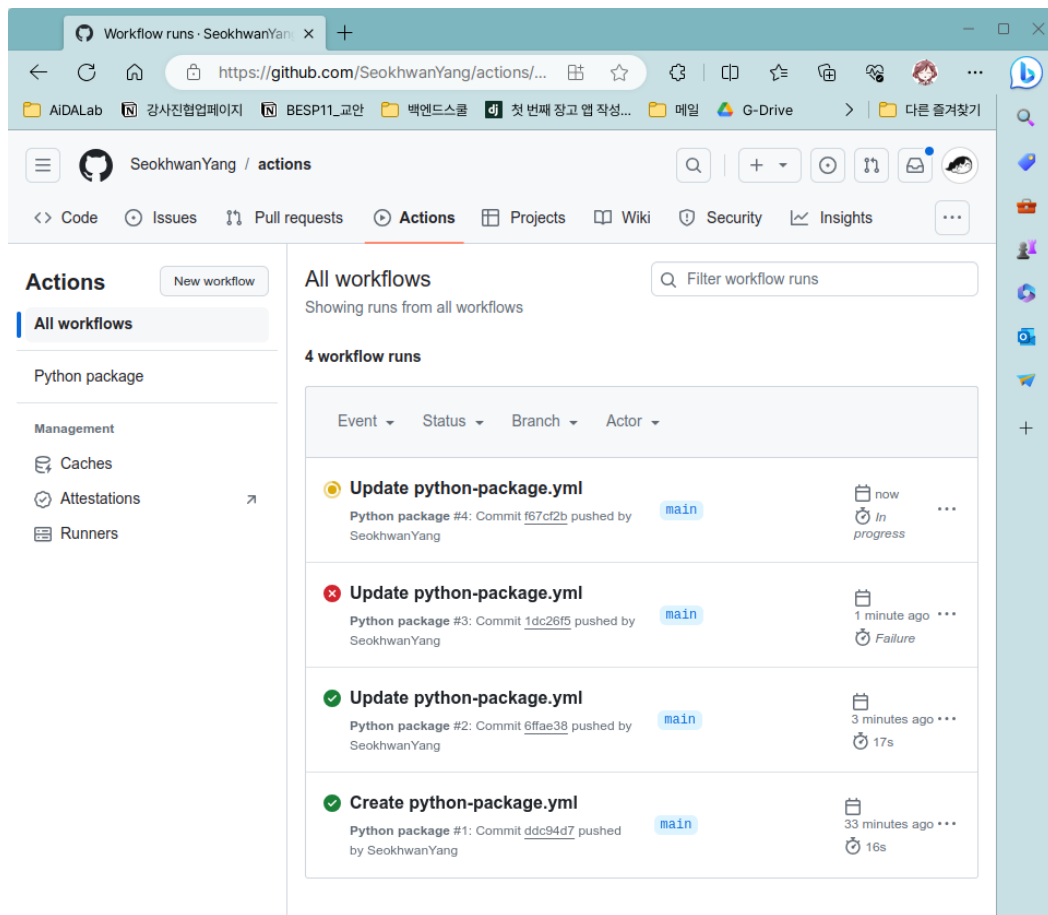
```
        with:
```

```
          author_name: SeokhwanYang
```

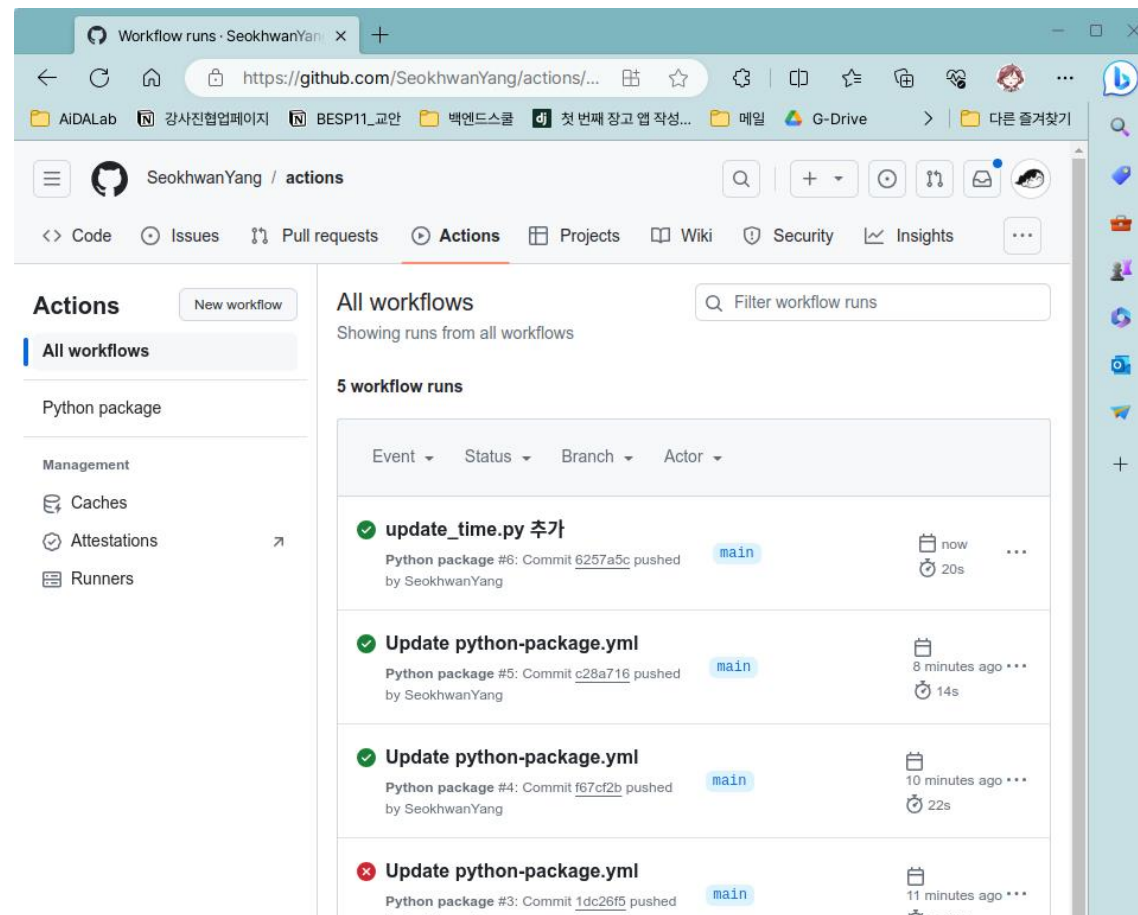
```
          author_email: yangseokhwan@naver.com
```

```
          message: Update index.html with current time # 커밋 메시지
```

```
          add: "index.html" # 변경된 파일 추가
```



권한 문제 발생



The screenshot shows the GitHub Actions interface for user SeokhwanYang. The 'All workflows' section displays 8 workflow runs. The runs are as follows:

Event	Status	Branch	Actor	Time
update_test_html.yml 추가	Failed (red X)	main	SeokhwanYang	now (13s)
update_test_html.yml 추가	Running (yellow circle)	main	SeokhwanYang	now (In progress)
update_time.py 추가	Completed (green check)	main	SeokhwanYang	20 minutes ago (20s)
Update python-package.yml	Completed (green check)	main	SeokhwanYang	28 minutes ago (14s)
Update python-package.yml	Completed (green check)	main	SeokhwanYang	30 minutes ago (22s)
Update python-package.yml	Failed (red X)	main	SeokhwanYang	...
Update python-package.yml	Completed (green check)	main	SeokhwanYang	...

여러 버전에 대한 중복 실행 문제 발생

The screenshot shows the GitHub Actions interface for user SeokhwanYang. The 'All workflows' section displays 10 workflow runs. The runs are as follows:

Event	Status	Branch	Actor	Time
update_test_html.yml 수정-> No...	Running (yellow circle)	main	SeokhwanYang	now (In progress)
update_test_html.yml 수정-> No...	Running (yellow circle)	main	SeokhwanYang	now (In progress)
update_test_html.yml 추가	Failed (red X)	main	SeokhwanYang	2 minutes ago (13s)
update_test_html.yml 추가	Completed (green check)	main	SeokhwanYang	2 minutes ago (22s)
update_time.py 추가	Completed (green check)	main	SeokhwanYang	22 minutes ago (20s)
Update python-package.yml	Completed (green check)	main	SeokhwanYang	30 minutes ago (14s)
Update python-package.yml	Completed (green check)	main	SeokhwanYang	37 minutes ago (...

The screenshot shows the details of a failed workflow run. The title is "update\_test\_html.yml 수정-> Permission 추가 #3". The status is "Failure" with a total duration of "20s". The workflow was triggered by a push from "SeokhwanYang" on the "main" branch. The "Summary" section lists three jobs: "build (3.9)" (success), "build (3.10)" (failure), and "build (3.11)" (failure). The "Run details" section shows a matrix build with three jobs: "build (3.10)" (9s, failure), "build (3.11)" (9s, failure), and "build (3.9)" (6s, success). The "Annotations" section shows 4 errors and 21 warnings, with a specific error for "build (3.11)": "Error: error: Your local changes to the following files would be overwritten by checkout: index.html".

버전 중복 해결 후 완료

The screenshot shows the "All workflows" page for the repository. It displays a list of 16 workflow runs. The runs are filtered by "All workflows". The list includes runs for "rebase 후 수정", "update\_test\_html.yml 수정-> 동...", and "update\_test\_html.yml 수정-> No...". The statuses of the runs are: "In progress", "In progress", "Failure", "Success", "Failure", "Success", "Failure", "Success", "Failure", "Success", "Failure", "Success", "Failure", "Success", "Failure".

**THANK  
YOU**

