

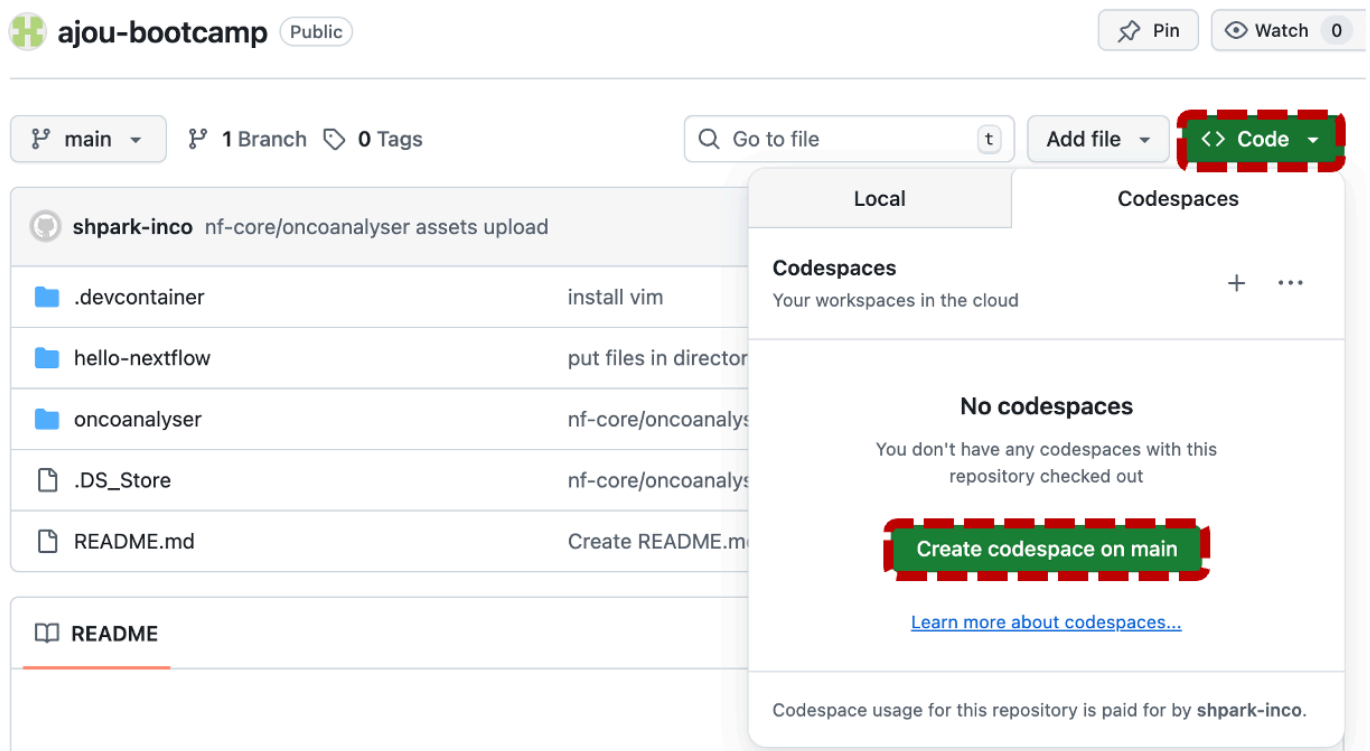
아주대 인공지능 부트캠프 2회차 3강

- 1. 실습 환경 세팅
  - 2. Nextflow 실습
    - 2.1 Nextflow 기본
    - 2.2 Nextflow 출력 경로 설정
    - 2.3 Nextflow 변수 설정
- 3. nf-core/oncoanalyser 실습

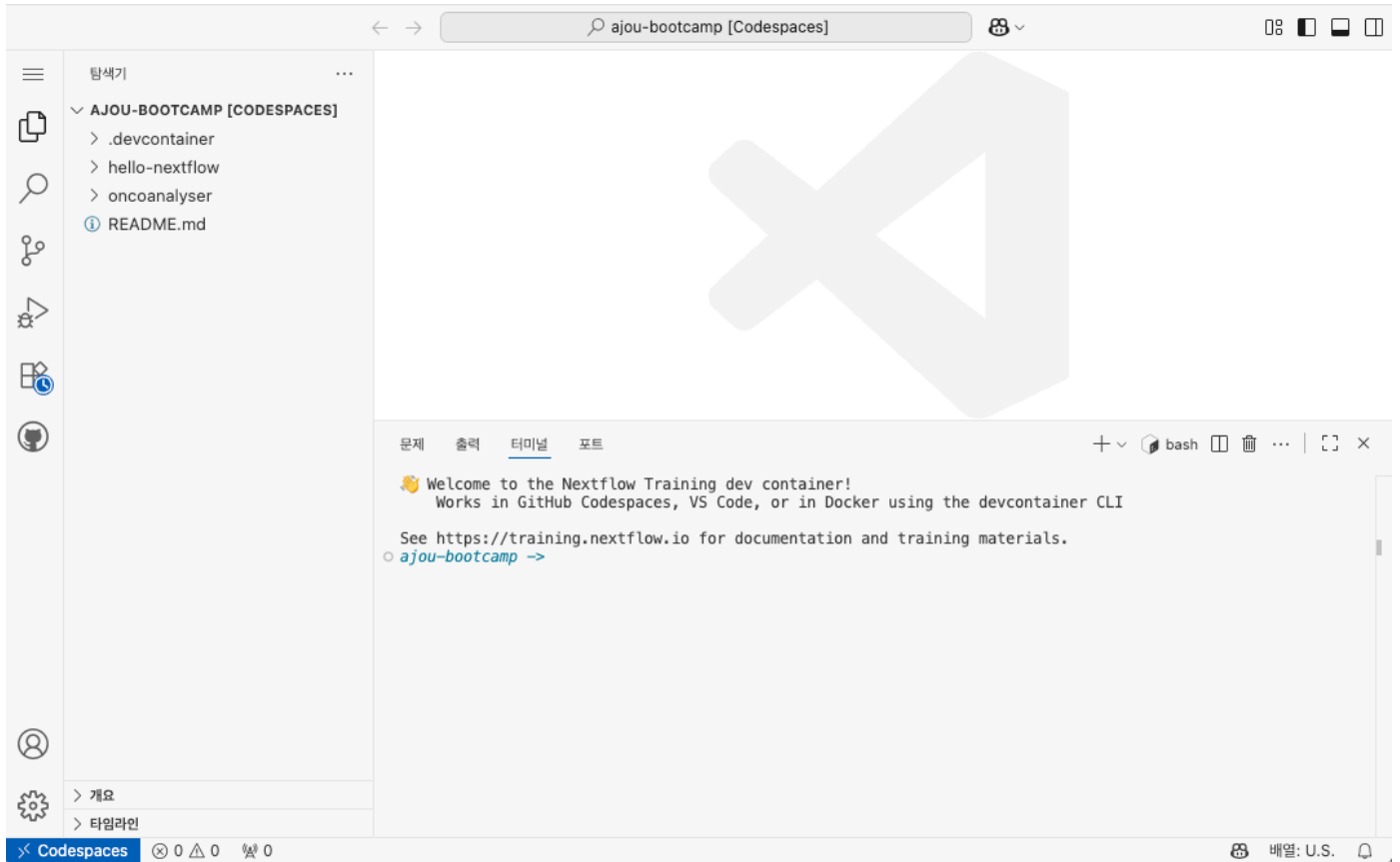
# 아주대 인공지능 부트캠프 2회차 3강

## 1. 실습 환경 세팅

- 1. 오른쪽 링크에 접속합니다. <https://github.com/shpark-inco/ajou-bootcamp>
- 2. [<> Code] 클릭 - [Create codespace on main] 클릭



- 3. 아래와 같은 창이 뜹니다.



## 2. Nextflow 실습

### 2.1 Nextflow 기본

**Step 1.** 실습 디렉토리 살펴봅니다. (리눅스 명령어는 아래 참고)

```
11
```

리눅스 명령어	기능
ll	현재 디렉토리 확인
cd	디렉토리로 이동
cd ..	하위 디렉토리로 이동
vim	텍스트 파일 수정

**Step 2.** hello-world.nf 스크립트를 살펴봅니다. (process, workflow, output, script)

```
# hello-nextflow 디렉토리로 접근합니다.
cd hello-nextflow

# hello-nextflow 디렉토리의 하위 항목을 출력합니다.
ll

# hello-world.nf 파일을 읽어봅니다.
vim hello-world.nf
```

**Step 3.** hello-world.nf 스크립트를 나갑니다. [ESC] - `:q!` - [ENTER]

**Step 4.** hello-world.nf 스크립트를 구동합니다. (Step, Hash)

```
# nextflow run [nf 스크립트]
nextflow run hello-world.nf
```

```
● ajou-bootcamp/hello-nextflow -> nextflow run hello-world.nf
Nextflow 25.04.6 is available - Please consider updating your version to it

NEXTFLOW ~ version 25.04.3

Launching `hello-world.nf` [exotic_feynman] DSL2 - revision: c33d41f479

executor > local (1)
[61/2dccb1] sayHello | 1 of 1 ✓
```

**Step 5.** 결과를 확인합니다. (work)

```
# hello-world.nf 구동 결과 확인
ll

# work 디렉토리 접근
cd work

# work 디렉토리 결과 확인
tree

# 상위 디렉토리로 이동합니다.
cd ..
```

## 2.2 Nextflow 출력 경로 설정

**Step 1.** hello-world.nf 파일을 수정합니다.

```
# hello-world.nf 파일 열기
vim hello-world.nf
```

**Step 2.** 텍스트 수정 모드로 전환 (키보드 `a`) 후, `publishDir` function 설정합니다. (indentation)

```
process sayHello {
  publishDir 'results', mode: 'copy'

  output: # 이하 생략
```

**Step 3.** hello-world.nf 스크립트를 저장 후 나갑니다. [ESC] - `:wq!` - [ENTER]

**Step 4.** hello-world.nf 스크립트를 구동 후, 결과를 확인합니다. (Resume)

```
# nextflow run [nf 스크립트]
nextflow run hello-world.nf -resume
```

**Step 5.** `results` 디렉토리에서 결과를 확인합니다.

## 2.3 Nextflow 변수 설정

**Step 1.** hello-world.nf 파일을 수정합니다.

```
# hello-world.nf 파일 열기
vim hello-world.nf
```

**Step 2.** 텍스트 수정 모드로 전환 (키보드 `a`) 후, `input` 설정, `script` 수정, `workflow` 수정 합니다.

```
process sayHello {
  publishDir 'results', mode: 'copy'

  input: # 추가(1)
    val greeting # 추가(2)

  output:
    path 'output.txt'

  script:
    """
    echo "$greeting" > output.txt # 수정
    """
}

workflow {
  sayHello(params.greeting) # 수정
}
```

**Step 3.** hello-world.nf 스크립트를 저장 후 나갑니다. [ESC] - `:wq!` - [ENTER]

**Step 4.** hello-world.nf 스크립트를 구동 후, 결과를 확인합니다. (Resume)

```
# nextflow run [nf 스크립트]
nextflow run hello-world.nf --greeting "Hello, my name is Sohee"
```

**Step 5.** `results` 디렉토리에서 결과를 확인합니다.

## 3. nf-core/oncoanalyser 실습

**Step 1.** `ajou-bootcamp` 디렉토리로 이동합니다. (`cd ..` 활용)

**Step 2.** nf-core 에서 파이프라인을 찾아봅니다.

```
nf-core pipelines list
```

**Step 3.** nf-core/oncoanalyser 실행을 위해 파이프라인을 내려받습니다.

```
nextflow pull nf-core/oncoanalyser -r 2.1.0
```

**Step 4.** nf-core/oncoanalyser를 구동합니다.

```
nextflow run nf-core/oncoanalyser -r 2.1.0 -profile docker,test --outdir ./
```