# pliki .yaml

#### Omówienie podstawowych rzeczy dla złapania intuicji:

## [Screeny przykładu z SO niżej.]

- 1. To jest nazwa, która będzie się wyświetlać w panelu po lewej pod "Workflows".
- Jeśli nie będzie tego wiersza, to w miejscu na nazwę odpalenia będzie nazwa commita tak jak niżej.
   Jeśli coś się tu wpisze, to zawsze taki tytuł będzie miało odpalenie.
  - W tym wypadku github.actor to nick użytkownika, który odpalił workflow.
- 3. Definiuje, na jakie eventy w repozytorium ma reagować workflow. W tym wypadku push.
- 4. Defniuje, jakie pliki w pushu odpalają workflow.
- 5. Definiuje, co ma się po kolei wykonać w workflow.
- 6. Nazwa, która wyświetli się po wejściu w odpalony workflow. [drugi screen, lewy panel, "Autograding"]
- 7. Definiuje, jaką wirtualną maszynę i którą wersję ma odpalić GitHub.
- 8. Definiujesz konkretne kroki, które będą się od góry odpalać po kolei.

Steps can run commands, run setup tasks, or run an action in your repository, a public repository, or an action published in a Docker registry. Not all steps run actions, but all actions run as a step. Each step runs in its own process in the runner environment and has access to the workspace and filesystem. Because steps run in their own process, changes to environment variables are not preserved between steps. GitHub provides built-in steps to set up and complete a job.

- 9. Nazwa danego kroku.
- 10. Komendy, które mają się wykonać w tym kroku.

[Więcej szczegółów: https://docs.github.com/en/actions]

```
Each run keyword represents a new process and shell in the runner environment. When you provide multi-line commands, each line runs in the same shell. For example:

• A single-line command:

• name: Install Dependencies run: npm install

• A multi-line command:

• name: Clean install dependencies and build run: | npm ci npm run build
```

```
Jobs. <job_id>.steps[*].working-directory 

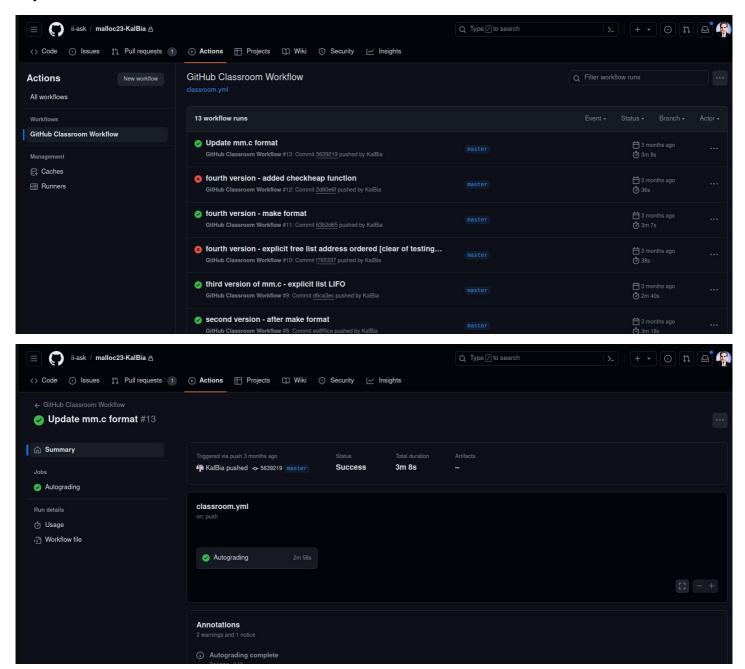
Using the working-directory keyword, you can specify the working directory of where to run the command.

- name: Clean temp directory 
    run: rm -rf * 
    working-directory: ./temp

Alternatively, you can specify a default working directory for all run steps in a job, or for all run 
steps in the entire workflow. For more information, see "defaults.run.working-directory" and 
"jobs.<job_id>.defaults.run.working-directory."

You can also use a run step to run a script. For more information, see "Essential features of GitHub 
Actions."
```

### Przykład z SO:



```
name: GitHub Classroom Workflow

on: [push]

jobs:
build:
name: Autograding
runs-on: ubuntu-latest
container: cahirwpz/ii-so:latest
steps:

- uses: actions/checkout@v2
- name: Check code formatting
run: ./run-clang-format.sh
- name: Check for unauthorized modifications
run: ./check-files.py
- uses: education/autograding@v1
```

## Inne przykłady:

```
name: Greeting from Mona

on: push

jobs:
    my-job:
    name: My Job
    runs-on: ubuntu-latest
    steps:
    - name: Print a greeting
    env:
        MY_VAR: Hi there! My name is
        FIRST_NAME: Mona
        MIDDLE_NAME: The
        LAST_NAME: Octocat
    run: |
        echo $MY_VAR $FIRST_NAME $MIDDLE_NAME $LAST_NAME.
```