

pliki .yaml

Omówienie podstawowych rzeczy dla złapania intuicji:

```
name: run-unit-tests <- 1
run-name: ${ github.actor } is running unit tests <- 2
on: <- 3
  push:
    paths: <- 4
      - '**.py'
jobs: <- 5
  run-tests:
    name: run-tests <- 6
    runs-on: ubuntu-latest <- 7
    steps: <- 8
      - name: build project <- 9
        run: (cos tutaj) <- 10
      - name: run tests
        run: (cos tutaj)
```

Plain text

[Screeny przykładu z SO niżej.]

1. To jest nazwa, która będzie się wyświetlać w panelu po lewej pod "Workflows".
2. 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. Definiuje, 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:

The screenshot shows the GitHub Actions interface for a repository named 'ii-ask / malloc23-KalBia'. The 'Actions' tab is selected, displaying a list of workflow runs for the 'GitHub Classroom Workflow' (classroom.yml). The list shows 13 workflow runs, with the most recent one being 'Update mm.c format' (Commit 5639219 pushed by KalBia) which is in a 'Success' state. The interface includes a search bar for workflow runs and a sidebar with navigation options like 'All workflows', 'Workflows', 'Management', 'Caches', and 'Runners'.

Event	Status	Branch	Actor
Update mm.c format	Success	master	KalBia
fourth version - added checkheap function	Failure	master	KalBia
fourth version - make format	Success	master	KalBia
fourth version - explicit free list address ordered [clear of testing...	Failure	master	KalBia
third version of mm.c - explicit list LIFO	Success	master	KalBia
second version - after make format	Success	master	KalBia

The screenshot shows the details of a specific workflow run for the 'GitHub Classroom Workflow' (classroom.yml). The run is titled 'Update mm.c format #13' and is in a 'Success' state. The interface includes a search bar for workflow runs and a sidebar with navigation options like 'Summary', 'Jobs', 'Run details', 'Usage', and 'Workflow file'. The main content area shows the workflow file 'classroom.yml' and the job 'Autograding' which has a duration of 2m 58s. There are also annotations showing 2 warnings and 1 notice, with the last one being 'Autograding complete'.

Triggered via	Status	Total duration	Artifacts
KalBia pushed 5639219	Success	3m 8s	-

classroom.yml
on: push

Autograding 2m 58s

Annotations
2 warnings and 1 notice

Autograding complete

```

1   name: GitHub Classroom Workflow
2
3   on: [push]
4
5   jobs:
6     build:
7       name: Autograding
8       runs-on: ubuntu-latest
9       container: cahirwpz/ii-so:latest
10      steps:
11        - uses: actions/checkout@v2
12        - name: Check code formatting
13          run: ./run-clang-format.sh
14        - name: Check for unauthorized modifications
15          run: ./check-files.py
16        - 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.

```

```

steps:
  - name: My first step
    if: ${ github.event_name == 'pull_request' && github.event.action ==
'unassigned' }}
    run: echo This event is a pull request that had an assignee removed.

```