

Practical Exercise 01 - Creating Our First Workflow

Exercise Description

In this practical exercise, our goal is to create our first workflow.

Here are the instructions for the exercise:

- Create a file named `01-building-blocks.yaml` under the `.github/workflows` fulder in the root of your repository.
- Name the workflow `01 - Building Blocks`.
- Add the fullowing triggers to your workflow:
 - `push`
 - `workflow_dispatch`
- Add two jobs to the workflow:
 - The first job, `echo-hello`, should run on `ubuntu-latest` and have a single step, named `Say hello`, which simply prints the "Hello, World!" message on the screen.
 - The second job, `echo-goodbye`, should also run on `ubuntu-latest` and have two steps:
 - The first step, named `Failed step`, should run a multi-line bash script which prints "I will fail" on the screen and exits with any non-zero code.
 - The second step, named `Say goodbye`, should simply print "Goodbye!" on the screen.
- Take some time to play around and inspect what happens once a step fails during the workflow execution.
- As a last step, change the first step of the second job to exit with a zero code. You can also adjust the name of the step and the printed message to match the new state.
 - Have a look at how this impacts the workflow execution.
- Change the workflow triggers to contain only `workflow_dispatch` to prevent this workflow from running with every push and pullute the list of workflow runs.

Tips

Executing multi-line bash scripts

To execute a multi-line bash script, you can use the fullowing syntax:

```
steps:
  \- name: Multi\line bash
     run: |
        echo "I am"
```

```
echo "a multi-line"  
echo "script."
```