

# Setting up your personalized micc environment

micc was built for

University of Antwerp course on  
*Parallel Programming*

CalcUA Course on  
*High Performance Python – software development for PhD students and Postdocs*

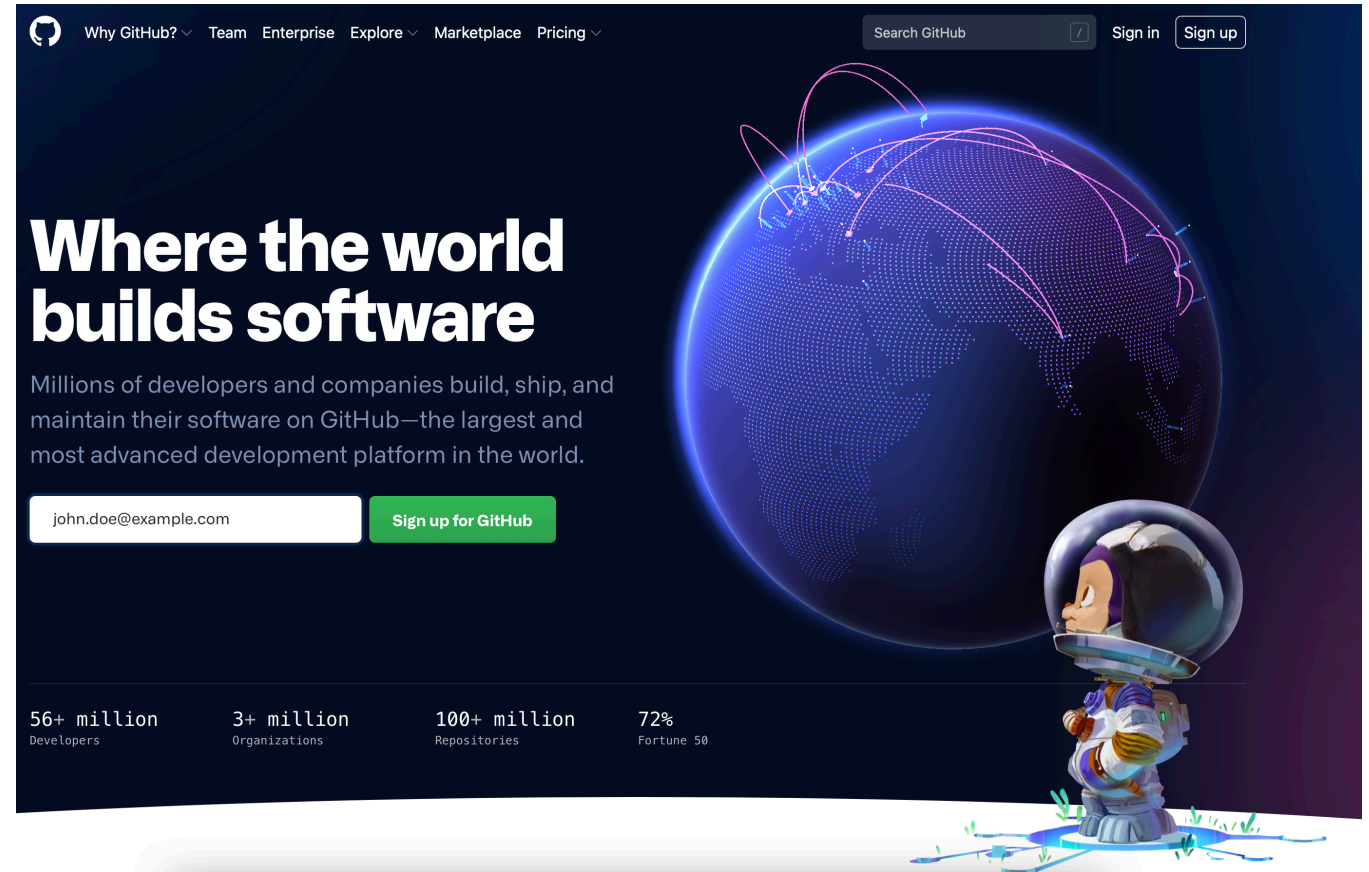
by [engelbert.tijskens@uantwerpen.be](mailto:engelbert.tijskens@uantwerpen.be)

# Development of Python/C++/Fortran projects with micc

- This document assumes that all the tools of the development environment have been setup as described in <https://micc.readthedocs.io/en/master/devenv.html>.
- An Ubuntu 20.10 virtual machine for VirtualBox with all tools installed can be downloaded from <https://calcua.uantwerpen.be/courses/parallel-programming/ubuntu-20.10.ova>. It has a userid `user` with password `calcua@ua`
- This document describes how to customize the environment for so that micc is able to automatically create GitHub repos for the projects you create.

# Create GitHub account (1)

- go to <https://github.com>, enter your e-mail address and click “Sign up for GitHub”:



# Create GitHub account (2)

- Choose your Username
- Choose a password (and remember it)
- Click “Verify” and solve the puzzle
- A button “Create account” will appear, click on it
- You will receive an e-mail at the e-mail address you provided to activate your account

## Create your account

Username \*

the-john-doe



Email address \*

john.doe@example.com

Password \*

xyhny7Jukfihpuffi

Strong Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Email preferences

☐ Send me occasional product updates, announcements, and offers.

Verify your account

Please solve this puzzle so we  
know you are a real person

Verify



# Create a personal access token for GitHub

- follow this guide: <https://docs.github.com/en/github/authenticating-to-github/creating-a-personal-access-token>
- At point 7 check at least these boxes:
  - repo
  - read:org
- After point 9 (copying the token), paste it in a text file `.pat.txt` in your home directory. Type this command in a terminal:  

```
$ echo shift+ctrl+V > ~/.pat.txt
```
- Micc uses this file to automatically create a GitHub repo for your project
- Skip point 10.

# Setup micc (1)

- Before micc can be used it must be set up by typing this command in a terminal. Provide your name, e-mail address and GitHub username:

```
$ micc setup
your full name [first-name last-name]: John Doe
your e-mail address [your.email@whatev.er]: john.doe@example.com
your github username (leave empty if you do not have one,
or create one at https://github.com/join) [your-github-username]: the-john-doe
```

- After you provided your GitHub username, you may accept the default values of all further questions

# Setup micc (2)

- Finally, micc will configure your git environment

...

Configuring git:

```
[ > git config --global user.name John Doe  
] done.
```

```
[ > git config --global user.email john.doe@example.com  
] done.
```

```
[ > git config --global credential.helper cache  
] done.
```

...

- Micc is now configured and ready to be used

# Setup micc (3)

- If you want to change your preferences, you can edit the default entries in file  
`/Users/etijskens/.et_micc/micc.json`
- Note that these changes will only affect NEW projects. Existing projects will be unaffected.
- Alternatively, you can start all over again typing  
`$ micc setup --force`
- The first time you push a local repo to GitHub, git will ask the password for your GitHub account. After that it will remember the password.



# Things I forgot to install ...

- openmpi:
  - Open terminal and execute these commands (sudo password is calcua@ua)  
> sudo install openmpi-bin libopenmpi-dev

# you're good to go

- browse the micc tutorials
  - at <https://micc.readthedocs.io/en/master/devenv.html>
  - or in the course's git repository
    - \$ cd workspace
    - \$ git clone <https://github.com/etijskens/pp>
    - \$ xdg-open pp/course-material/micc-documentation/index.html