

# SUPPORTING STUDENT PYTHON FLUENCY AT GFI

An all-in-one framework using the integrated  
**GitHub+VSC** solution and **Jupyter** notebooks

*Vår Dundas & Julien-Pooya Weihs  
Teachers breakfast meeting 22/3/2023*

# STATUS-QUO

- Programming is necessary in scientific life, studies, career, etc.
- Students feel overwhelmed
  - Single course formally covering the topic: INF100
  - Educational research shows that repetition and continuity is key, to learning — not a one-time training program
  - Different competing environments for Python: CoCalc, Anaconda, ActivePython...
- Teaching staff struggle too
  - With lectures including programming
  - Because of experience bias expecting students to know all about Python,
  - Programming complex topic to teach (linguistics, scientific semantics, use cases...)

## A SUSTAINABLE SOLUTION FOR ALL USERS

- Introduce programming in an incremental way
- Help students out of their struggles
- Provides students with professional skills (programming + data management)
- Offer sustainable support for teaching staff
- Create collaboration at all levels of teaching and learning environment
- Support integrating programming into all courses


# THE TOOLS

**GitHub** and **Visual Studio Code** as a framework

# GITHUB

- Web-based platform for ~~software~~ project development using version control
- Allows multiple people to work on project simultaneously by tracking changes
- Over 100 million developers and more than 386 million repositories (3/23)
- New industry standard for programming, data management
- Commonly used for open-source projects, 40 million public repositories (3/23)
- Public repositories: 1700 hit for “climate models”, 1000 hits for “oceanography”, 2600 hits for “meteorology” (3/23)

# GITHUB PROFILE



## Kristian Agasøster Haaga

kahaaga

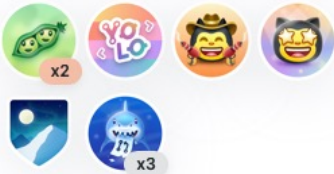
Unfollow

Post-doc at the Department of Earth Science, University of Bergen, Norway. I work on the detection of causal relationships from time series.

19 followers · 4 following

@earthsystemevolution  
Bergen, Norway

### Achievements



Beta [Send feedback](#)


### Highlights

PRO

### Organizations


OverviewRepositories36ProjectsPackagesStars26

### Pinned

 [JuliaDynamics/TimeseriesSurrogates.jl](#) Public


A Julia package for generating timeseries surrogates

Julia 35 7

 [JuliaDynamics/CausalityTools.jl](#) Public


Algorithms for detecting associations, dynamical influences and causal inference from data.

Julia 88 7

 [UncertainData.jl](#) Public

Working efficiently with datasets where observations have associated uncertainties.

Julia 15 4

 [JuliaDynamics/ComplexityMeasures.jl](#) Public

Estimators for probabilities, entropies, and other complexity measures derived from observations in the context of nonlinear dynamics and complex systems

Julia 29 7

### 1,081 contributions in the last year

	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar
Mon													
Wed													
Fri													

Learn how we count contributions

Less More

### Contribution activity

March 2023

Created 8 commits in 2 repositories

[JuliaDynamics/CausalityTools.jl](#) 7 commits

[JuliaDynamics/ComplexityMeasures.jl](#) 1 commit

Created a pull request in JuliaDynamics/ComplexityMeasures.jl that received 3 comments

Mar 1

Fix log units

Fixes #256 Summary: Introduces a convenience method convert\_logunit that converts between numbers computed with different logarithm bases. This ...

+94 -49 3 comments

2023

2022

2021

2020

2019

2018

2017

2016

2015

# VISUAL STUDIO CODE

- Free and open-source code editor developed on... GitHub!
- Wide range of features (code completion, debugging tools, integrated terminal)
- Full integration and synchronization of projects with GitHub
- Most popular coding environment (74% of 71000 respondents, 2022)
- Supporting more than 100 programming languages
- Over 45000 extensions (3/23) to add features and customize workflow with language interpreters, compilers, renderers, etc.

# USER INTERFACE

**A** Activity Bar

**C** Editor Groups

**D** Panel

**E** Status Bar

**B** Side Bar

The screenshot displays the Visual Studio Code interface with the following components:

- Activity Bar (A):** Located on the left, it contains icons for Explorer, Search, Source Control, Run and Debug, and Extensions. The Explorer view is active, showing the file structure of the workspace.
- Side Bar (B):** Located below the Activity Bar, it displays the contents of the selected file, 'package.json'.
- Editor Groups (C):** The main area for editing code. It contains several editor windows, including 'findModel.ts', 'findOptionsWidget.ts', 'CONTRIBUTING.md', 'contextmenu.ts', and 'package.json'.
- Panel (D):** Located at the bottom, it displays the output of the terminal, showing the command 'npm install' and its output.
- Status Bar (E):** Located at the very bottom, it displays the current file's path, the active editor's name, and the current line and column numbers.

The Explorer view shows the following file structure:

- VS CODE PROJECTS (WORKSPACE)
- .mention-bot
- .travis.yml
- .yarnrc
- appveyor.yml
- CODE\_OF\_CONDUCT.md
- CONTRIBUTING.md
- gulpfile.js
- LICENSE.txt
- npm-debug.log
- OSSREADME.json
- package.json
- product.json
- README.md
- ThirdPartyNotices.txt
- tsfmt.json
- tslint.json
- yarn.lock
- vscode-docs
  - .vscode
  - blogs

The terminal output shows the command 'npm install' and its output:

```
PS C:\Users\gregvanl\vscode> npm install
1133 LICENSE.txt
607796 npm-debug.log
42422 OSSREADME.json
3699 package.json
683 product.json
3732 README.md
103675 ThirdPartyNotices.txt
729 tsfmt.json
11050 tslint.json
203283 yarn.lock
```



# HOW IT WORKS - DEMONSTRATION

**Coding** and **synchronisation** between collaborators

**Uploading files** into a repository

**Terminal** vs **GitHub Desktop**



# GITHUB + VSC FOR TEACHING

**Advantages and potential**

# INSIGHTS FROM THE LITERATURE

## ADVANTAGES

- *Collaboration*: improves communication and teamwork skills
- *Version control*: track changes and revert to previous versions if necessary
- *Feedback*: pull-request feature is a platform for discussion between users
- *Transparency of activity*: holds users accountable for their contributions
- *Opportunity*: students develop technical skills at industry-standard

## LIMITATIONS

- *Learning curve* for beginners unfamiliar with Git-environment
  - Unthinkable to have a degree in geophysics without robust coding skills
  - Develop shared base of knowledge
- May not be suitable for *grading assignments*
  - Grade student work on mitt.uib
- Some may be *uncomfortable sharing work*
  - Possibility to use private repositories at any time

# WHAT IS CURRENTLY DONE AT GFI

- Existing repositories for GFI courses (3/23):
  - *Public*: GEOF211, GEOF212, GEOF321 (in development), GEOF337 (user's personal), GFPy toolbox
  - *Private*: GEOF105 (probably many more)
- What some course repository owners say:
  - “strong tool for shared projects [...] with well-structured way to decide of changes”
  - “shifting the access of the course material to GitHub would certainly be a great step”
  - “the repository can live on, beyond my association with the course”
  - “I can more easily track the development of the course materials”
  - “perfect for sharing my Jupyter notebooks”

# MOTIVATION FOR IMPLEMENTATION

- Complete project folders available on single web-based platform
- Shared documents always up to date for everyone
- Teaching staff can access and use common data/toolboxes/libraries
- Streamlines course management for both students and teaching staff
- Transparency about workflow and individual contributions
- Possible private repositories for data protection and security

# APPLICATION OF THE FRAMEWORK: AN EXAMPLE OF TEACHING RESOURCE

Introduction to **Python** for geophysicists

A **tailored** set of **Jupyter notebooks** for beginners and  
intermediate users





# OUR SUGGESTION

Developing a sustainable **teaching** and **learning**  
environment for **programming** at GFI

## COMMON PLATFORM FOR COURSES AT GFI

- GFI-level GitHub environment for courses — GEOFIxx, GEOF2xx, GEOF3xx
- For students
  - support learning with modern tools for scientific programming & collaborative project management
  - address programming anxiety
  - develop new skills
- For teaching staff
  - support teaching by creating continuity between courses and over several years
  - foster collaborative approach on course design (peer-peer, Students as Partners)
  - develop new *culture of programming* to transmit to students

# POSSIBLE FUTURE STEPS

## Short-term implementation:

- Creation of tutorials + handbook for installation/setup/use/collaboration with GitHub + VSC
- User workshops for students
- User workshops for teaching staff
- Jupyter notebooks to familiarise new students with python scientific coding

## Medium-term evolution:

- Opening of space for more creativity in teaching (lectures, assessment, activities)
- Expansion to MatNat? (can be research-based)

## Long-term developments:

- New *culture of programming* for students
- ???



*WHAT DO YOU THINK?*

1. What would you need to implement GitHub + VSC in your teaching?
2. What would possible barriers be, what to avoid?
3. Further thoughts and comments are warmly welcome!