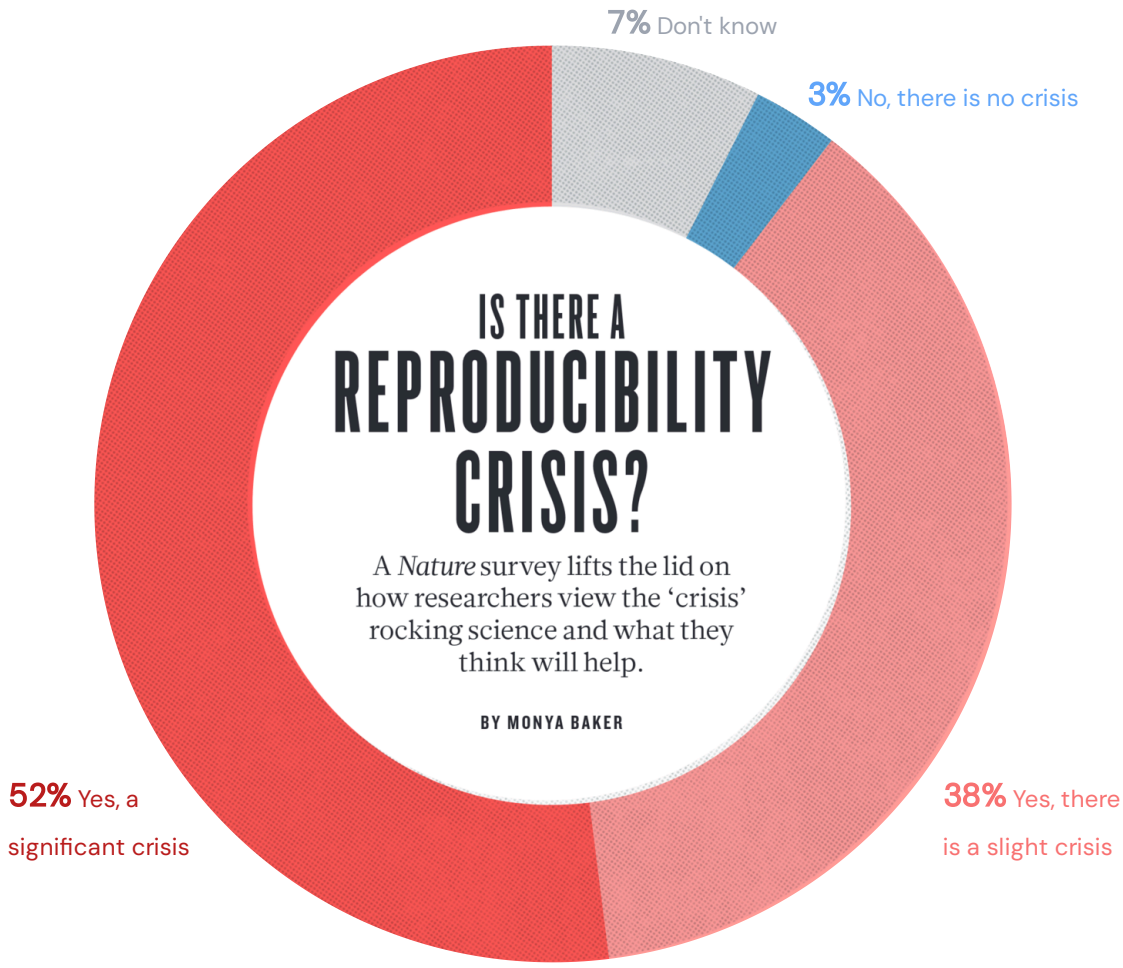

CODING SPACES
Why a Good Coding Practice Matters

Why are you here?

What do you expect?


7% Don't know
3% No, there is no crisis
IS THERE A
REPRODUCIBILITY
CRISIS?
A *Nature* survey lifts the lid on how researchers view the 'crisis' rocking science and what they think will help.
BY MONYA BAKER
52% Yes, a significant crisis
38% Yes, there is a slight crisis

How do Code & Science relate?

- We typically see a paper/thesis as the way of communicating our science.
- Code is more than just a tool you use for your science. It is the "true" documentation of *how* you obtained your results!
- Code is more than only instructions for the computer. It is interpreted by the computer but read and written by humans.
- It can help your readers comprehend, reproduce, and trust your science.

→ *Do we handle code in an adequate way?* 🤔

A typical situation

We read a {paper, student report, thesis} and are curious about how the results were achieved.

> on the temperature data. We cleansed the data following standardized procedures and performed a polynomial fit to the resulting data set. The statistical analysis, which includes standardized statistics of the data set as well as a student-t test, was performed with Python and the `statsmodels` package. In the following, we will go through the re-

Were you able to reproduce the results or comprehend how the data were processed?

Probably no. The obvious next step would be to look at the code.

From here, what typically hinders you from reproducing the results?

There is no code at all.

On request, the authors send you their scripts.

All the code is put in one very loooong Jupyter notebook.

The code is of lousy quality (poorly documented, not self-explanatory, etc.).

etc.

How can we improve the situation?

Share your code!

The worst code shared is better than having it not shared at all.

"But my code is ugly. I don't want to put it online!" 🥺

Mmm... do you suffer from code shaming? 🤔

You can have fun improving it! Make the fruits of your work more understandable, modular, reproducible, reusable, ...

What good code should look like

- Good code should be
  - easy to understand and to use
  - well documented
  - reproducible
- Good code should also
  - have a clean layout
  - be robust
  - run reliably
  - produce consistent results

☑☰ **Quick check:**
How many of these points typically apply to your code? 😉

Easier said than done? Maybe.

But you *can* get there 🚀

Know and master your tools (coding is a craft).

Have a reasonable workflow.

Be aware of and follow conventions and concepts.
(They exist for a reason 😉)

Build habits and be consistent in what you do!

The purpose of this series

- Not yet another coding class but rather an introduction to concepts, tools, workflows
- A space for learning, trying things out, discussions, socializing
- Jointly create a new and better coding culture
- Open for all skill levels

Concept: A loose series of different topics

Each coding space session with a different focus

Playgrounds we have in mind

1. Code Management
2. Principles and Practices
3. Quality and Testing
4. Development Tools
5. Data Management
6. Special Topics

Code Management

Unleash the power of organized coding and collaboration magic!

Version Control & Collaborative Development

↳ How to keep track of your own and other people's changes.

Virtual Environments / Python Environments

↳ Make your installation environment reproducible.

Continuous Integration

↳ Automate the execution of repetitive tasks.

Packaging your Code

↳ Make it installable, and how to publish it.

Principles and Practices

Crack the code to clean, elegant, and rock-solid programming!

Programming Principles / Clean Code

↳ How to write clear and understandable code.

Code Architecture & Design

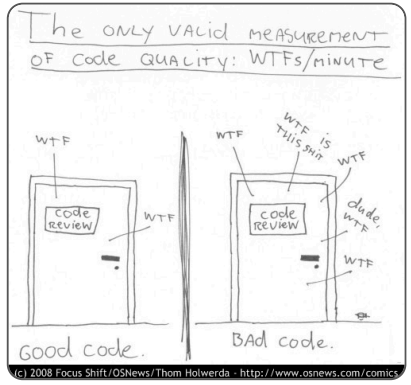
↳ How to outline and structure your code.

Object-orientation

↳ Learn how to program with classes.

Quality and Testing

Turn your code into a fortress of perfection with bulletproof quality and testing!



Code Documentation

↳ How to make documenting your code a breeze.
Turn your documentation into an interactive website.

Test Driven Development

↳ How to test your code automatically and ensure robust outcomes.

Code Reviews & Walkthroughs

↳ Let's go together through our codes and improve them.

Development Tools

Turbocharge your coding with the coolest tools and AI sidekicks!

Integrated Development Environments

↳ Editors on stereoids

AI-assisted Programming

↳ Let AI support you in your programming

Python libraries in Earth System Sciences

↳ Get pro with Pandas, Xarray, Cartopy, ...

Data Management

Become a data wizard, mastering the magic of FAIR and reproducible data handling!

- FAIR Data Principles in Practice
- Standards in Earth System Sciences
- Data Versioning

Special Topics

Step into the adventure zone of coding with exciting and unexpected challenges!

- Jupyter Notebooks vs. Standalone Applications
- Developing CLI interfaces
- Your wishes ...

What's next?

- **Find a date** for the following meetings
- **Join us on Mattermost**:

  Team *"All-IfM-social"* → Channel *"Coding Spaces"*

Questions?

Drop us an email:

yves.sorge@uni-hamburg.de

markus.ritschel@uni-hamburg.de

markusritschel ysorge