

# ENAC-IT4R Data & Code services

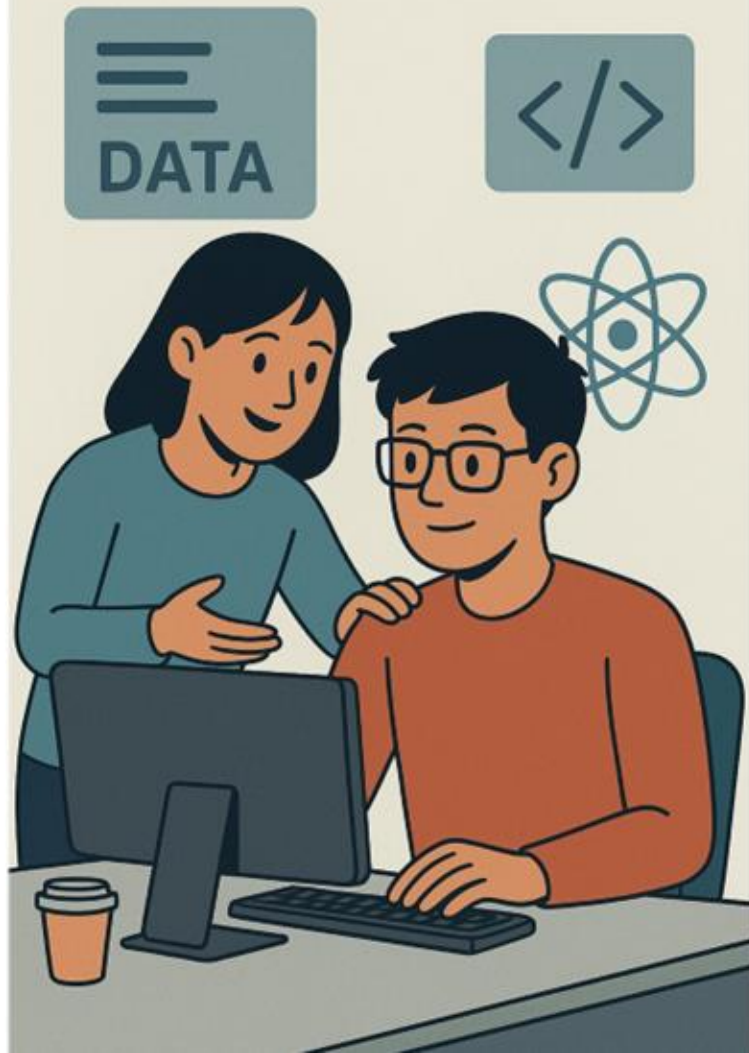
Tech talk x  
RSE  
Romandie

02.06.2026

# Tuesday Tech talks?

# Tuesday Tech talks!

- Informal lunchtime gatherings (~~bring your own lunch!~~) designed to foster interdisciplinary knowledge sharing within the ENAC community (and beyond!) on **data and coding topics**, across academic domains and career levels.



# Agenda

- Welcome
- Literate programming
  - Pierre-Olivier Valès, CEDE, EPFL
  - Simon Dürr, HES-SO Valais-Wallis
- From docstrings to live docs
  - Hugo Solleder, ENAC-IT4R (10 min)
- Static site generation for project & code documentation
  - Alexander Lepper, LASUR (10 min)
- Open discussion

# ENAC-IT4R Data & software engineers

Data & Code support services [enac-it4r.epfl.ch](http://enac-it4r.epfl.ch)



swiss academies of arts and sciences  
Open Research Data National Prize 2024

Ask us anything about your  
Data & Code !  
[enacit4research@epfl.ch](mailto:enacit4research@epfl.ch)



Interns



# Supporting researchers with their data and code

## Fostering Collaborative, FAIR, Open Research Data & Code for Scientific Valorization

### Research Data Lifecycle



#### Software dev & packaging

- Code packaging, refactoring, migration, deployment
- Collaborative repository mgt

Welcome to GPM-API  
A python package to access and analyze satellite precipitation data: radar, passive microwave and IMERG products are at your fingertips!

Start exploring the Global Precipitation Measurement Mission archive now!

Deployment: `pip install gpm-api` or `conda-forge`

Activity: `PyPI downloads` | `conda downloads`

Python Versions: `Python 3.8-3.11 3.12-3.13`

Supported Systems: `Linux` | `macOS` | `Windows`

Project Status: `Repo status`

Build Status: `Tests` | `Lint` | `docs`

Linting: `flake8` | `black` | `ruff` | `Codetools`

Code Coverage: `coverage` | `Coverage`

Code Quality: `codeclimate` | `codebase` | `code quality`

License: `license`

Community: `Stack` | `GitHub`

Citation: `DOI`

[Slack](#) | [Documentation](#)

#### Data valorization

- Web application development
- Database set-up
- Data pipelines automation

Un atlas des paysages de mobilité à Vernier  
Vers une lecture affective de l'espace de la Commune de Vernier, Genève

Cartographie affective de Vernier  
Ces cartographies affectives représentent l'espace urbain vécu dans Vernier à travers quatre entretiens réalisés avec des habitants. L.e.s nous guidant à travers la commune.

- 01 La forêt tropicale
- 02 L'envers des colonnes
- 03 Une barrière bleue
- 04 Passages secrets
- 05 Archipels et deltas
- 06 Territoires DOM-TOM
- 07 Jeu de pistes
- 08 Au plateau des pins
- 09 Sous les pavés, la Cité
- 10 La non des rues
- 11 Une histoire de calloas
- 12 L'invention d'un ritournelle
- 13 La campagne Naïlle
- 14 La non de terre
- 15 Là où l'on revient

Infos | Tutoriel | A Propos

# Pilot projects

- Open-source software or research scripts for packaging and distribution
- Dataset valorization through tailored visualization

# Literate programming & computing

# Literate programming

## What is it

- Introduced 1984 by Donal Knuth
- Code should be written for humans, not machines

*Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do – Literate Programming (1984)*

- Originally TeX as document formatting language, PASCAL as programming language
- Two main operations:
  - Tangle: produces compilable source command
  - Weave: generates documentation files from the literate file

# Literate programming

R - Sweave, knitr,

Quarto

- *The traditional way of writing a report as part of a statistical data analysis project uses two separate steps: running the analysis using some software, and then copy and pasting the results into a word processing tool - Leisch (2002)*
- Literate programming as a way to avoid copy-pasting
- Vignettes as long-form guide to packages, framed around a problem that the package is designed to solve.

# Literate computing

## IPython, Jupyter

*A literate computing environment is one that allows users not only to execute commands interactively but also to store in a literate document format the results of these commands along with figures and free-form text that can include formatted mathematical expressions.*

- Interactive computing across multiple programming languages
  - **Julia, Python, R**
- Focus on **computing** rather than **writing**
- Execute commands interactively and store the results, figures and free-form text in literate document

# Literate computing in practice

- (G)note at EPFL
- Quarto at HES-SO Valais Wallis

- Execution order:
  - Can execute cells out of order, making the analysis not reproducible
  - Safest approach: restart & run
- Version-control
  - JSON behind the scenes
  - Very verbose diffs
  - Can use external tools (nbsstripout, nbdime, ...)
- Dependency issues

```

286   "output_type": "execute_result"
287   }
288   @@ -292,7 +299,7 @@
289   },
290   {
291     "cell_type": "code",
292     "execution_count": 37,
293     "id": "9b278e48",
294     "metadata": {},
295     "outputs": [],
296     @@ -322,9 +319,18 @@
297     "\n",
298     "landuse_with_grid_reset = landuse_with_grid_reset_index()\n",
299     "\n",
300     "# count occurrences of each landuse class per 500m grid\n",
301     "counts = landuse_with_grid_reset\n",
302     "    .groupby(['MUT_RIGHT', 'lu_col'])\n",
303     "    .size()\n",
304     "    .unstack(fill_value=0)\n",
305     "    }\n",
306     @@ -338,7 +338,7 @@
307   },
308   {
309     "cell_type": "code",
310     "execution_count": 39,
311     "id": "8d7f8c41",
312     "metadata": {},
313     "outputs": [
314     @@ -355,6 +363,7 @@
315     " <thead>\n",
316     "   <tr style='text-align: right;'>\n",
317     "     <th>/th>\n",
318     "     <th>geometry/th>\n",
319     "     <th>/th>\n",
320     "   "
321   "output_type": "execute_result"
322   }
323   @@ -294,7 +294,7 @@
324   )
325   },
326   {
327     "cell_type": "code",
328     "execution_count": 22,
329     "id": "9b278e48",
330     "metadata": {},
331     "outputs": [],
332     "\n",
333     "landuse_with_grid_reset = landuse_with_grid_reset_index()\n",
334     "\n",
335     "# count occurrences of each landuse class per 500m grid\n",
336     "counts = landuse_with_grid_reset\n",
337     "    .groupby(['index_right', 'lu_col'])\n",
338     "    .size()\n",
339     "    .unstack(fill_value=0)\n",
340     "    }\n",
341   },
342   {
343     "cell_type": "code",
344     "execution_count": 23,
345     "id": "8d7f8c41",
346     "metadata": {},
347     "outputs": [
348     @@ -363,7 +364,7 @@
349     " <thead>\n",
350     "   <tr style='text-align: right;'>\n",
351     "     <th>/th>\n",
352     "     <th>geometry/th>\n",
353     "     <th>/th>\n",
354     "   "

```

# Alternatives

- Marimo – Python
- Pluto.jl – Julia
  
- Perks
  - Reactive notebooks: input change → dependent cells re-run automatically
  - Version-control friendly: pure Python/Julia!
  - Built-in dependency management
  - Can deploy as app
  
- DEMO TIME

# Documentation generation

- In addition to structured code and clear variable names, adding documentation to functions/classes/methods helps other understand (how to use) your code
  - (Future you will also be grateful)
  
- Each language has its own approach/styles
  - C++: Javadoc/QT
  - Python: Google, Numpydoc, Javadoc-like

# Source-based documentation

## Two approaches

- Docstrings
  - String literal, part of the the runtime object and can be inspected programmatically
  - Julia, Python,...
- Documentation comments
  - Serves a similar purpose
  - Processed by external documentation tools → comment extraction systems
  - R, C/C++, ...

```
julia> """
    Return a greeting for the given name.
    """
    function greet(name)
        return "Hello, $name"
    end
greet

julia> Docs.doc(greet)
    Return a greeting for the given name.

help?> greet
search: greet get reset parent Base.rest repeat

    Return a greeting for the given name.
```

# Demo-time

# Beyond code documentation

## MkDocs

- MkDocs at the Urban Sociology Laboratory (LASUR)

# MkDocs

## Open source horror story?

- Original maintainer AWOL, then works on v2.0, completely breaking plugin compatibility
- Internal infighting between *de facto* maintainers
- Community split
  - ProperDocs – drop-in replacement
  - Zensical –
  - MkDocs 2.0 - ?

What the actual fuck? I'm the author and license holder.



@lovelydinosaur · Mar 9, 2026

Now please revert the PyPI changes and fuck off.



@lovelydinosaur · Mar 9, 2026