



PhD Course

Python for Empirical Economics: Web Data, Large Databases & Automation

block course: 11 – 13 January 2027; each 09h00 through 16h00
ESP 36, room 4030/31

Course Instructor: Dr. Jan Rabenseifner (UHH)

Course Value: 5 LP; for graduate program in section B=specialization

Assessment/Student evaluation: Practical coding project (group assignment submitting executable Python code and Quarto-generated PDF reports). Grading is pass/fail.

Course Language: English

Software: Python, VS Code; please bring your laptop to the course.

Prerequisites: Economics PhD students working empirically, with prior programming experience in statistical software. No prior Python knowledge is required.

Registration: Please register via STiNE. The class is limited to 20 people.

If there are more registrations than spots available, HCHE members will be given priority.

For all organizational matters or if you are unable to register via STiNE please contact andrea.buekow@uni-hamburg.de.

Course Overview:

Designed specifically for empirical economics PhD students, this course focuses exclusively on modern Python workflows for automated data collection, text processing, large database management, and interactive web deliverables. The curriculum moves from foundational Python syntax into more advanced tools.

Day 1-2: Syntax Transition, Tooling & AI Workflows

- **Modern Environment Configuration:** Setting up robust development environments using VS Code and the Positron IDE.
- **Project Architecture:** Initializing projects using the UV package manager, pyproject.toml configurations, and dependency locking.

- **AI Integration:** Accelerating development with AI coding tools inside the IDE, alongside responsible prompt engineering workflows.
- **Syntax & Core Data Structures:** Mastering foundational Python types including lists, dictionaries, sets, and tuples.
- **Control Flow & Strings:** Writing efficient loops, conditional statements, custom functions, and executing robust text manipulation (f-strings, slicing, indexing).
- **Modular Design & Error Handling:** Structuring code across modules and implementing defensive exception-handling frameworks.

Day 2-3: Web Scraping, Automation & Interactive Deliverables

- **API Automation:** Querying HTTP web APIs, handling authentication, managing rate limits, and parsing JSON responses.
- **Data Mining:** Scraping unstructured data from static and dynamic web pages.
- **Text Processing:** Cleaning scraped text data and implementing basic text classification pipelines.
- **Interactive Web Reporting:** Creating dynamic HTML charts and exploratory hover/zoom graphics using for dashboards and working paper previews.

Day 3: Large Databases & Out-of-Core Processing

- **Distributed Processing Concepts:** Handling large-scale data workflows using tools like Spark.
- **Out-of-Core Pipelines:** Managing and reshaping multi-gigabyte files (wide vs. long transformations) that exceed local RAM constraints.
- **Automated Reporting:** Synthesizing final data structures and code into reproducible Quarto-generated PDF deliverables.

During the course, the content can still be adjusted to reflect students' current level of knowledge or their progress.