# Open Code

CONTENTS

## Workshop Description

Modern research relies heavily on computational tools, and adopting open coding practices enhances reproducibility, collaboration, and transparency. This workshop provides an introduction to open code principles and practical tools for managing research code efficiently. Participants will learn essential skills in version control, collaborative coding, and project management using Git, GitHub, VS Code, and uv.

This session is designed for researchers, students, and faculty members who want to streamline their coding workflows and embrace best practices in open research software. No prior experience with Git, GitHub, or uv is required, making it an excelle starting point for beginners.

### Learning Outcomes

By the end of the workshop, participants will be able to:

- Understand the significance of open code in research.
- Utilize Git for version control and code management.
- Navigate GitHub for collaboration and sharing.
- Integrate Git and GitHub seamlessly within VS Code.
- Manage Python projects and dependencies efficiently using uv.
- Implement best practices for open and reproducible research coding.

### Format

This is an interactive, hands-on workshop conducted virtually via Zoom. Participants will follow along with demonstrations and practice key concepts through guided exercises.

### Prerequisites

To ensure a smooth learning experience, participants are encouraged to install the following tools before the workshop:

- **Git**: Download Git
- **VS Code**: Download VS Code
- **uv**: Installation Guide

We look forward to seeing you at the workshop and helping you unlock the power of open code in your research!

## Date and Time

- **Date**: Monday, March 10, 2025
- **Time**: 10:00 AM - 11:00 AM ET
- **Location**: Virtual (Zoom link will be provided upon registration)

## Instructor

- Name: Qiusheng Wu
- Affiliation: Department of Geography & Sustainability, University of Tennessee, Knoxville
- Email: qwu18@utk.edu
- Website: https://gishub.org

## Who Should Attend?

This workshop is ideal for:

- Researchers looking to improve their coding workflow and version control.
- Students and faculty interested in collaborative coding practices.
- Anyone new to Git, GitHub, VS Code, and uv who wants a hands-on introduction.

No prior experience with Git or GitHub is required, making this session suitable for beginners.

## Registration

To attend, please complete the registration form at this link. Once registered, you will receive a confirmation email with the Zoom link and preparation instructions.

### Workshop Recording

A recording of the workshop will be available on the Open Geospatial Solutions YouTube channel after the event.

---

## 1. Introduction to Open Code

### Why Open Code?

- **Reproducibility**: Enables others (and your future self) to verify research results.
- **Collaboration**: Facilitates teamwork and contributions from the wider community.
- **Transparency**: Builds trust and allows scrutiny of methodologies.
- **Credit & Recognition**: Properly shared code can be cited and acknowledged.

### Tools for Open Code

- **Git**: A version control system to track changes in code.
- **GitHub**: A cloud-based platform for hosting and sharing Git repositories.
- **VS Code**: A popular code editor with built-in Git and GitHub integration.
- **GitHub Alternatives**: GitLab, Bitbucket, Codeberg, etc.
- **uv**: An extremely fast Python package and project manager, written in Rust.

---

## 2. Introduction to Git

### What is Git?

- A distributed version control system.
- Allows tracking changes, branching, and collaboration.

### Installing Git

- Windows: Download Git
- macOS: `brew install git`
- Linux: `sudo apt install git` (Debian/Ubuntu) or `sudo pacman -S git` (Arch)

### Basic Git Commands

```
# Configure Git with your name and email
$ git config --global user.name "Your Name"
$ git config --global user.email "your.email@example.com"

# Initialize a Git repository
$ git init
```

```
# Check repository status
$ git status

# Add files to staging area
$ git add filename

# Commit changes
$ git commit -m "Your commit message"

# View commit history
$ git log
```

---

## 3. Introduction to GitHub

### Setting Up GitHub

- Create an account at GitHub

### Creating a Repository on GitHub

1. Go to GitHub, click on *New Repository*.
2. Name the repository and select visibility (public/private).
3. Initialize with a README (optional).

### Connecting Local Repository to GitHub

```
# Clone a repository from GitHub
$ git clone https://github.com/your-username/repository-name.git
```

---

## 4. Introduction to VS Code

Visual Studio Code is a free, cross-platform source-code editor developed by Microsoft for Windows, Linux, and macOS. It's lightweight, highly customizable, and packed with features to make coding efficient and enjoyable. Key features include:

- **Debugging**: Identify and fix issues in your code directly within the editor.
- **Syntax Highlighting**: Make your code easier to read and understand.
- **Intelligent Code Completion**: Get smart suggestions based on your code.
- **Snippets**: Quickly insert reusable pieces of code.
- **Code Refactoring**: Simplify and restructure your code efficiently.
- **Built-in Git Integration**: Manage version control seamlessly within the editor.

VS Code also supports an extensive library of **extensions** to enhance functionality, allowing you to tailor the editor to meet your specific needs.

### Installation

1. **Download and Install**: Visit the **Visual Studio Code website** and download the installer for your operating system (Windows, macOS, or Linux). Follow the installation instructions provided on the website.
2. **Launch VS Code**: After installation, open Visual Studio Code.
3. **Verify Installation**: Ensure that VS Code is installed correctly by checking the version. Open the terminal or command prompt and type `code --version`. You should see the installed version number.

### Important Extensions for Software Development

These extensions are critical for Python and Jupyter-based workflows:

- Python: Essential for Python development, offering linting, debugging, and IntelliSense.
- Jupyter: Enables seamless integration with Jupyter notebooks, letting you write, run, and debug notebooks within VS Code.
- GitHub Copilot: AI-powered code completion for productivity boosts.

---

### VS Code Extension Packs

An **extension pack** is a curated collection of related extensions bundled together to enhance a specific workflow or domain. These packs save time by providing all the essential tools in a single installation.

The Microsoft Python Data Science Extension Pack is a must-have for data scientists and geospatial analysts. It includes the following powerful tools:

- **Python**: Core Python support with IntelliSense, debugging, and more.
- **Pylance**: A fast, feature-rich language server for Python.
- **Jupyter**: Seamless Jupyter notebook support within VS Code.
- **Data Wrangler**: Explore, visualize, and clean tabular data
- **GitHub Copilot**: - AI pair programmer

## 5. Git Integration in VS Code

### Setting Up VS Code for Git and GitHub

- Authenticate with GitHub in VS Code:
    - Click on the **Accounts** icon in lower left corner.
    - Sign in to GitHub.

### Using Git in VS Code

- Open a project folder in VS Code
- Open the **Source Control** panel (Ctrl+Shift+G)
- Click “Initialize Repository” if necessary
- Use the UI to **stage, commit, and push changes**
- Use the built-in terminal for advanced Git commands

### Cloning Repositories in VS Code

1. Open **Command Palette** (`Ctrl+Shift+P` / `Cmd+Shift+P` on
2. Search for **“Git: Clone”**
3. Enter the repository URL from GitHub
4. Choose a local directory to store the repository
5. Open the repository in VS Code

---

## 6. Introduction to uv

uv is an extremely fast Python package and project manager, written in Rust. For packages that can be installed from PyPI, `uv` is a great alternative to `pip`. To install `uv`, follow the instructions in the official documentation to install uv on your system.

Once installed, you can use `uv` to create a new environment and install packages as follows:

```
cd /path/to/your/project
uv venv
uv venv --python 3.12
uv pip install jupyterlab leafmap
uv run jupyter lab
```

## 7. Collaborative Workflows

### Forking and Pull Requests

- **Forking**: Create your own copy of a repository to work on.
- **Pull Request (PR)**: Propose changes to the original repository.

### Steps for Contributing

1. **Fork a repository**
2. **Clone your fork using VS Code**
3. **Create a new branch**

```
$ git checkout -b feature-branch
```

4. **Make changes and commit**

```
$ git add .
$ git commit -m "Added a new feature"
```

5. **Push to your fork**

```
$ git push origin feature-branch
```

6. **Create a Pull Request on GitHub**

---

## 8. Best Practices for Open Code

- Use **meaningful commit messages**.
- Write **clear documentation** (README, comments, inline explanations).
- License your code (e.g., MIT, GPL) to clarify reuse terms.
- Use `.gitignore` to exclude unnecessary files.
- Regularly push and back up your work.

---

## 9. Hands-on Exercises

### Exercise 1: Version Control Basics

1. Create a new local Git repository.
2. Add and commit a sample file.
3. View commit history.

### Exercise 2: Using VS Code with Git

1. Clone a repository using VS Code.
2. Make changes and use the Source Control panel to commit.
3. Push the changes using the UI or terminal.

### Exercise 3: Collaboration on GitHub

1. Fork a sample repository.
2. Make changes in a new branch.
3. Submit a pull request.

---

## Wrap-up & Q&A

- Recap key concepts.
- Address participant questions.
- Encourage applying Git, GitHub, VS Code, and uv in research workflows.

---

Happy coding! 🎉

← Open Science
Open Science

Open Science Workshop
Open Data →