ATOC: Windows Setup Sheet (VS Code + Conda/Mamba + Git Bash)

What you are installing

- **Visual Studio Code (VS Code)**: the editor you will write code in.
- **Git for Windows**: provides **Git Bash** (a Linux-like terminal).
- **Conda + Mamba**: installs Python and scientific packages in an isolated environment.

Before you start (2 minute checklist)

1. Make sure you are on Windows 10 or 11.
2. Close VS Code if it is currently open.
3. If you have a school-managed laptop and installs fail, you may need to run installers as admin or contact IT.

Step 1: Install VS Code on Windows

1. Open a web browser and go to the official download page:
   `https://code.visualstudio.com/download`
2. Download the **Windows User Installer** (recommended).
3. Run the installer you downloaded (it will be named like `VSCodeUserSetup-x64-...exe`).
4. Click through the installer:
   - Accept the license agreement.
   - Keep the default install location.
   - **Important option:** If you see an option like **"Add to PATH"**, check it.
   - **Important option:** If you see **"Open with Code"** (adds a right-click menu), you can check it.
5. Finish the installation, then **Launch** VS Code.

Step 2: Install Git for Windows (this gives you Git Bash)

1. Go to the official Git for Windows page:
   `https://git-scm.com/install/windows`
2. Download the installer and run it.
3. Click through the installer:
   - Keep defaults unless your instructor says otherwise.
   - If asked about a default editor, choosing **VS Code** is fine (if it appears).
   - If asked about PATH, the common safe choice is **"Git from the command line and also from 3rd-party software"**.
4. Verify Git works:
   - Open **Git Bash** from the Start Menu.

- Type this and press Enter:

  ```
  git --version
  ```
- You should see a version number (example: `git version 2.xx.x`).

Step 3: Install Conda + Mamba (recommended: Miniforge)

Recommended for this course: Miniforge because it includes `conda` and `mamba` out of the box.

1. Go to the conda-forge installer page:
   https://conda-forge.org/download/
2. Download **Miniforge** for Windows (x86_64 for most laptops).
3. Run the Miniforge installer:
   - Keep defaults.
   - Make sure it creates Start Menu shortcuts (so you get "Miniforge Prompt").
4. Open **Miniforge Prompt** from the Start Menu.
5. Verify installs:

   ```
   conda --version
   mamba --version
   ```

If you already installed Anaconda instead (acceptable)

If you used Anaconda, you may need to install mamba once.

1. Open **Anaconda Prompt**.
2. Install mamba into the base environment:

   ```
   conda install -n base -c conda-forge mamba
   ```
3. Verify:

   ```
   mamba --version
   ```

Step 4: Make Conda work inside Git Bash (one-time setup)

We want Git Bash to recognize `conda` and `mamba`.

1. Open **Miniforge Prompt** (or Anaconda Prompt).
2. Run:

   ```
   conda init bash
   ```
3. Close **all** Git Bash windows.
4. Re-open **Git Bash**.
5. Test:

   ```
   conda --version
   mamba --version
   ```

If that test fails: Use the **Miniforge Prompt / Anaconda Prompt** for environment commands, and use Git Bash mainly inside VS Code for basic navigation and git.

Step 5: Configure VS Code to use Git Bash

1. Open VS Code.
2. Open the command palette: **Ctrl+Shift+P**.
3. Type `Terminal: Select Default Profile` and press Enter.
4. Choose **Git Bash**.
5. Open a new terminal: **Terminal** → **New Terminal**.
6. You should see a bash prompt (often with a `$`).

If Git Bash does not appear as an option

1. Make sure **Git for Windows** is installed (Step 2).
2. Restart VS Code.
3. If still missing, you can manually add a profile:
   - In VS Code press **Ctrl+,** to open Settings.
   - Search: `terminal.integrated.profiles.windows`
   - Add a profile pointing to Git Bash. Common locations are:

     ```
     C:\Program Files\Git\bin\bash.exe
     C:\Program Files\Git\usr\bin\bash.exe
     ```

Step 6: Install required VS Code extensions

You will install these extensions:

- Python (`ms-python.python`)
- Pylance (`ms-python.vscode-pylance`)
- Jupyter (`ms-toolsai.jupyter`)
- Ruff (`charliermarsh.ruff`)
- Markdown All in One (`yzhang.markdown-all-in-one`)

Option A (recommended): Install via the Extensions tab

1. In VS Code, click the Extensions icon on the left (or press **Ctrl+Shift+X**).
2. For each item below:
   - Paste the extension ID into the search bar.
   - Click **Install**.
3. Install these IDs (one at a time):

   ```
   ms-python.python
   ms-python.vscode-pylance
   ms-toolsai.jupyter
   charliermarsh.ruff
   yzhang.markdown-all-in-one
   ```

Option B: Install via command line (works if the `code` command is available)

1. Open VS Code, then open a terminal.
2. Run:

   ```
   code --install-extension ms-python.python
   code --install-extension ms-python.vscode-pylance
   code --install-extension ms-toolsai.jupyter
   code --install-extension charliermarsh.ruff
   code --install-extension yzhang.markdown-all-in-one
   ```

Step 7: Create the class Python environment using Mamba

Goal command:

```
mamba env create -f atoc-2025-lite.yml
```

1. Get the file `atoc-2025-lite.yml` from your course site or class repo.
2. Put `atoc-2025-lite.yml` in a folder you can find (example: `C:\Users\YourName\Desktop\atoc`).
3. Open **Miniforge Prompt** (or Anaconda Prompt).
4. Change directory to the folder containing the file (example):

   ```
   cd C:\Users\YourName\Desktop\atoc
   ```

5. Create the environment:

   ```
   mamba env create -f atoc-2025-lite.yml
   ```

6. Wait for it to finish. Then list environments:

   ```
   conda env list
   ```

7. Activate the new environment. (Use the name shown in `conda env list`. Example name below.)

   ```
   conda activate atoc-2025-lite
   ```

8. Quick test:

   ```
   python --version
   python -c "import numpy as np; print(np.arange(5))"
   ```

Step 8: Tell VS Code to use your new environment

1. Open VS Code.
2. Open the command palette: **Ctrl+Shift+P**.
3. Run: `Python: Select Interpreter`
4. Choose the interpreter that matches your new environment (it often includes the environment name).
5. **Jupyter:** If you open a notebook, choose the same environment as the kernel.

Common problems (fast fixes)

- **“mamba not found”**:
  - If using Miniforge, open **Miniforge Prompt** and try again.
  - If using Anaconda, run `conda install -n base -c conda-forge mamba`.
- **“conda not found” in Git Bash**:
  - Run `conda init bash` in Miniforge/Anaconda Prompt, then restart Git Bash.
- **VS Code terminal is not Git Bash**:
  - **Terminal** → **Select Default Profile** → **Git Bash**.
- **Jupyter cannot find a kernel**:
  - Activate your environment, then install the kernel tools if needed:

    ```
    mamba install ipykernel jupyter
    ```

Tip: When asking for help, include a screenshot of your terminal command + the full error message.