GitHub CLI Login Fallback Guide

If `gh auth login` (the normal browser-based login) fails or hangs, use this guaranteed method.

1■■ Create a Personal Access Token (PAT)

1   Go to https://github.com/settings/tokens
2   Click "Tokens (classic)" → "Generate new token → classic"
3   Name it "GitHub CLI Local", choose an expiration (90 days or longer)
4   Check these scopes: repo, workflow, read:org, gist
5   Click "Generate token" and copy the `ghp_...` value.

2■■ Tell GitHub CLI to use the token (no prompts, no browser)

Open PowerShell 7 (`pwsh`) and run:
```
$env:GH_TOKEN = 'ghp_yourTokenHere'
```
Replace `'ghp_yourTokenHere`` with your real token.

3■■ Verify the login works

```
Run:
gh auth status
```
Expected output:
```
github.com
✓ Logged in to github.com as (token)
```

4■■ Test the GitHub API

```
Run:
gh api user
```
You should see your GitHub username in the JSON output.

5■■ Run your sync

Once authenticated, the CLI will automatically use your token:
```
cd C:\PythonClass
.\sync.ps1
```

6■■ (Optional) Make the login persistent

If you want to remember the token automatically:
```
notepad $PROFILE
```
Add this line at the bottom of the file:
```
$env:GH_TOKEN = 'ghp_yourTokenHere'
```
Save and restart PowerShell or VS Code.

■ Summary

- Create token on GitHub → gives you a login key
- Set $env:GH_TOKEN in PowerShell → supplies token to CLI
- Run gh auth status → confirms login
- Run .\sync.ps1 → performs your sync
- Optional: Add to $PROFILE → keeps token permanently