FULL TEXT FILE: COMPLETE EXECUTION GUIDE

Run Brownie Simple Storage Project on Any Windows Machine (Step-by-Step)

This is the **FINAL MASTER TEXT FILE** containing **every installation**, **every fix**, and **every step** required to run the Brownie Simple Storage project successfully on ANY Windows computer.

It includes: - Required tools & download links - Installing Python, Git, C++ Build Tools - Installing pipx, Brownie - Installing correct solc version (0.6.0) - Fixing solc install errors (403, blocked downloads) - Cloning project - Running compile, test, deploy - Creating .env and setting up Sepolia - Deploying to testnet - Troubleshooting table

---

1. REQUIRED TOOLS

You must install the following tools. Without them, Brownie will NOT run.

Tool	Why Needed	Download
Python 3.10+	Required to run Brownie & solcx	https://www.python.org/downloads/
pipx	Safely installs Brownie globally	Installed through Python
Git	Clone GitHub repositories	https://git-scm.com
Microsoft Visual C++ Build Tools	Required for solcx compiler	https://visualstudio.microsoft.com/visual-cpp-build-tools/
solc 0.6.0	Solidity compiler for this project	Installed using solcx
Brownie	Ethereum testing + deployment framework	Installed using pipx
MetaMask Wallet	For Sepolia deployments	https://metamask.io
Infura Project ID	RPC endpoint for Sepolia	https://infura.io

---

2. INSTALL PYTHON

Download and install Python from: https://www.python.org/downloads/

IMPORTANT: ✔️Check **Add Python to PATH** during installation.

Verify installation:

```
python --version
```

---

3. INSTALL VISUAL C++ BUILD TOOLS

Required for solcx compiler to work.

Download: https://visualstudio.microsoft.com/visual-cpp-build-tools/

Open installer → Select: ✔️Desktop Development with C++

Install → Restart your computer.

---

4. INSTALL GIT

Download Git: https://git-scm.com/download/win

Verify:

```
git --version
```

---

5. INSTALL pipx AND BROWNIE

Run:

```
python -m pip install --user pipx
python -m pipx ensurepath
```

Restart PowerShell.

Install Brownie:

```
pipx install eth-brownie
```

If pipx fails:

```
python -m pip install --user eth-brownie
```

---

6. CLONE THE PROJECT

Open PowerShell:

```
cd %USERPROFILE%\Desktop
```

```
git clone https://github.com/PatrickAlphaC/brownie_simple_storage
cd brownie_simple_storage
```

---

7. INSTALL SOLIDITY COMPILER (solc 0.6.0)

Install solcx:

```
python -m pip install py-solc-x
```

Install solc version 0.6.0:

```
python -m solcx.install 0.6.0
```

FIX: solc INSTALLATION FAILING (403 ERROR)

If solc cannot download:

Option A – Clear cache and reinstall

```
python -m solcx.clear_cache
python -m solcx.install 0.6.0
```

Option B – Install GitHub tag

```
python -m solcx.install v0.6.0
```

Option C – Manual installation (100% works)

1. Visit GitHub solidity release v0.6.0
2. Download **solc-windows.exe**
3. Rename to **solc.exe**
4. Create folder:

   ```
   C:\Users\YourName\.solcx\solc-0.6.0
   ```

5. Move solc.exe into this folder.

---

8. COMPILE SMART CONTRACT

Run:

```
brownie compile
```

Expected:

```
Solc version: 0.6.0
Project has been compiled.
```

---

9. RUN TESTS

```
brownie test
```

Expected:

```
2 passed
```

---

10. DEPLOY LOCALLY (GANACHE)

```
brownie run scripts/deploy.py
```

This will: - launch Ganache - deploy SimpleStorage contract - print contract address

---

11. INTERACT USING BROWNIE CONSOLE

Run:

```
brownie console
```

Inside:

```
from brownie import SimpleStorage, accounts
contract = SimpleStorage.deploy({'from': accounts[0]})
contract.retrieve()
contract.store(25, {'from': accounts[0]})
contract.retrieve()
```

---

12. CREATE `.env` FILE

Inside project folder, create `.env`:

```
PRIVATE_KEY=0xYOUR_PRIVATE_KEY
WEB3_INFURA_PROJECT_ID=YOUR_INFURA_PROJECT_ID
```

RULES: ✔️Must start with **0x**✔️ No quotes✔️ No spaces✔️ Use a TEST ACCOUNT

---

13. UPDATE `brownie-config.yaml`

```
compiler:
  solc:
    version: "0.6.0"

dotenv: .env

wallets:
  from_key: ${PRIVATE_KEY}

networks:
  sepolia:
    host: https://sepolia.infura.io/v3/${WEB3_INFURA_PROJECT_ID}
    chainid: 11155111
```

---

14. UPDATE deploy.py

```
from brownie import SimpleStorage, accounts, config

def main():
    account = accounts.add(config["wallets"]["from_key"])
    print("Using account:", account.address)
    contract = SimpleStorage.deploy({"from": account})
    print("Contract deployed at:", contract.address)
```

---

15. RESTART TERMINAL

Then go back to the project:

```
cd path\to\brownie_simple_storage
```

---

16. GET SEPOLIA TEST ETH

Use Chainlink faucet: https://faucets.chain.link/sepolia

17. DEPLOY TO SEPOLIA

Run:

```
brownie run scripts/deploy.py --network sepolia
```

Expected output: - account address - transaction hash - deployed contract address

18. INTERACT ON SEPOLIA

```
brownie console --network sepolia
```

Inside:

```
from brownie import SimpleStorage, accounts, config
acct = accounts.add(config["wallets"]["from_key"])
contract = SimpleStorage[-1]
contract.retrieve()
tx = contract.store(42, {"from": acct})
tx.wait(1)
contract.retrieve()
```

19. TROUBLESHOOTING TABLE

Problem	Cause	Fix
Private key error	`.env` not loaded	Restart terminal / check names
solc 403	Download blocked	Clear cache or manual install
SimpleStorage[-1] fails	No contract deployed	Deploy again
Brownie not found	PATH issue	Restart terminal
MetaMask wrong key	Missing 0x prefix	Add 0x
Sepolia tx fails	No test ETH	Use faucet

20. EXPECTED OUTPUT SUMMARY

You should see: - Successful compile - 2 tests passed - Local deployment address - Sepolia deployment address - retrieve() function returns updated value

---

✔️END OF FILE — COMPLETE WORKING BROWNIE EXECUTION GUIDE

If you want: - PDF version - Word (DOCX) version - MCA practical file format

Just tell me!