5.3 Python Environment

Go To Top

Return to Development Environment

Phase 0 may use Python as a developer support and automation language. The team may use Python for Repository setup, prerequisite Validation, generation orchestration, Configuration checks, log inspection, test harnesses, and other lightweight developer utilities.

Python should not automatically become the normative runtime implementation language for Phase 0 Nodes. If the team uses Python for runtime Nodes, the relevant Node profile should explicitly identify Python as the implementation technology for that Node. Otherwise, the team should treat Python as support tooling.

The Python environment should remain isolated from the system Python installation. Developers should use a project-local virtual environment, typically named `.venv/`, and the Repository should ignore that directory in Git.

The Repository should explicitly declare Python dependencies. For a simple Phase 0 support environment, the team may use `requirements.txt`. If the project later adopts a package-based structure, the team may use `pyproject.toml`. The team should not rely on globally installed Python packages, undocumented local tools, or developer-specific shell profiles.

Python scripts should provide clear command-line behaviour. They should validate inputs, avoid hard-coded absolute paths, return non-zero exit codes on failure, write useful diagnostic output, and document functions with docstrings. Python utilities should support Reproducibility, not hide implementation decisions inside informal scripts.
