



GlobalBoost Coding Hacks

8. Leverage Python Type Hints for Better Code Quality

Why: Dynamic typing in Python can lead to runtime errors. Type hints provide static type checking, improving IDE autocompletion, bug detection, and team collaboration, especially in large-scale projects using frameworks like FastAPI or Django.

How to Implement: Use annotations like `: type` for parameters and `-> type` for returns. Check with tools like mypy via `mypy script.py`. Integrate in new codebases or gradually add to existing ones for refactoring.

```
python
# Without hints
def add(a, b):
    return a + b

# Hack: With type hints
def add(a: int, b: int) -> int:
    return a + b

# Usage
result = add(5, 3) # IDE flags if wrong types passed
```

Analysis: Type hints catch 15-20% more bugs pre-runtime per studies, without performance overhead (they're ignored at runtime). Tools like PyCharm or VS Code enhance productivity; in open-source repos, hinted code sees 30% fewer issues, fostering scalable, maintainable software.