

■ How to Get Complete, Working Code from an AI Assistant

(Guidelines for software development projects — R, Python, JavaScript, or mixed packages)

1■■ Start Every Project in Full-Code Mode

Begin your session with a clear setup statement:

Prompt Example:

```
"You are in Full-Code Mode. Produce complete, runnable code — no placeholders,
no TODO, no omitted sections.
Assume I want a ready-to-run, installable project (R package, Python module, or
web app).
Do not simplify or restyle working code unless I ask you to. Ask for
confirmation only when necessary, then proceed without asking again."
```

2■■ Five Rules for Reliable, Executable Results

- Say "complete, functional code only." Never accept partial code with ... or fill-ins.
- Specify exactly what should change ("Modify only the JS file.").
- Ask for full file structure to verify all components exist.
- Confirm once, then let it run — no multiple confirmations.
- Keep outputs as code blocks, not rendered previews.

3■■ Collaboration Rules for Teams

1. Freeze working code unless someone explicitly approves changes.
2. Label each version (e.g., layoutWidget_0.5.6) before editing.
3. Use clear scopes like "Edit only JS" or "Fix this error in R."
4. Share exact console error messages.
5. If the AI can't run code, ask for a **build script** to run locally.

4■■ Avoiding Phantom Builds

If the AI says it is "building," "packaging," or "executing," make sure it actually can. If not, say:

"If you can't execute code, generate a full local build script that creates the same files."

That prevents long waits and ensures usable output.

5■■ Optional Debugging Guardrail

```
"Treat all existing code as production code.
Only change what is necessary to fix the described problem.
Keep all other behavior identical."
```

6■■ Example: Starting a Reliable Coding Session

Prompt Example:

```
"You are in Full-Code Mode. We are building an R package with an HTMLwidget
written in D3.js.
Provide complete runnable files – no placeholders.
I want to install it locally with devtools::install_local() and run
demo_layoutWidget().
Do not modify visuals or rename files unless requested.
When ready to create the .zip, ask once for confirmation, then proceed."
```

■ Summary — The LLM Developer Contract

No placeholders: Code runs as-is

No silent changes: Predictable behavior

One confirmation: No endless waiting

Transparency: You know what's being done

Self-contained output: All files ready to build