FROM NON-CODER TO BUILDER

Quick Reference Cheat Sheet

YOU + AI = BUILDER | You: Direction & Judgment | AI: Code & Speed

THE LOOP

DESCRIBE → GET → RUN → EVALUATE

1. Describe what you want in plain English
2. Get code from AI (don't worry if you don't understand)

3. Run it → 4. Evaluate results → Repeat until it works

DECOMPOSITION

BIG TASK → SMALL TASKS

Rule: If it takes more than one sentence, split it

Each piece should be testable independently

Build one piece at a time, verify, then continue

EFFECTIVE PROMPTS

Bad: "Make it work"

Good: Include these 5 elements:

- WHAT: "Create a function that calculates..."
- WHERE: "In Python, add to the existing file..."
- HOW: "Use the formula: stress = force / area"
- WHY: "This will be used for safety checks"

GIVING FEEDBACK

GOT X, EXPECTED Y, CHANGE Z

- "I got [actual result]"
- "I expected [what you wanted]"
- "Please change [specific thing]"

Include error messages, screenshots, examples

DEBUGGING ERRORS

Syntax Error: Typo - check spelling, brackets, colons

Name Error: Variable doesn't exist - check spelling

Type Error: Wrong data type - string vs number

Index Error: List position doesn't exist

Copy full error message → Paste to AI → Ask for fix

THE 5-ATTEMPT RULE

STUCK AFTER 5 TRIES? RESTART FRESH

1. Start new conversation (clear context)
2. Rephrase the problem differently
3. Break into smaller pieces

Sunk cost fallacy is real - cut losses early

READING CODE

Variables: Named boxes storing values (x = 5)

Functions: Reusable recipes (def calculate():)

Conditionals: Decisions (if/else)

Loops: Repeat actions (for, while)

Lists: Collections ([1, 2, 3]) - Dicts: key-value pairs

SHARING CONTEXT

AI ONLY KNOWS WHAT YOU TELL IT

Always include:

- Your existing code (paste it)
- Full error messages (copy exactly)
- What you've already tried

PROJECT STRUCTURE

```
my_project/
  ├── main.py
  ├── requirements.txt
  ├── data/
  └── tests/
```

AI TRUST LEVELS

High trust: Syntax, boilerplate

Medium: Logic, algorithms

Verify: Business logic, edge cases

Always test: Security, data

WHEN STUCK

1. Ask AI with full context
2. Search the error online
3. Rephrase the question
4. Break into smaller parts

Clear Goal + Small Steps + Iteration + Persistence = Working Code

QUICK PYTHON COMMANDS

```
python script.py        pip install package        pip freeze > requirements.txt        python -m venv venv
```

MAGIC PHRASES FOR AI

"Explain like I'm new" "What does this error mean?" "Show me step by step" "What am I missing?"

You don't need to know everything to build something useful. Start building!