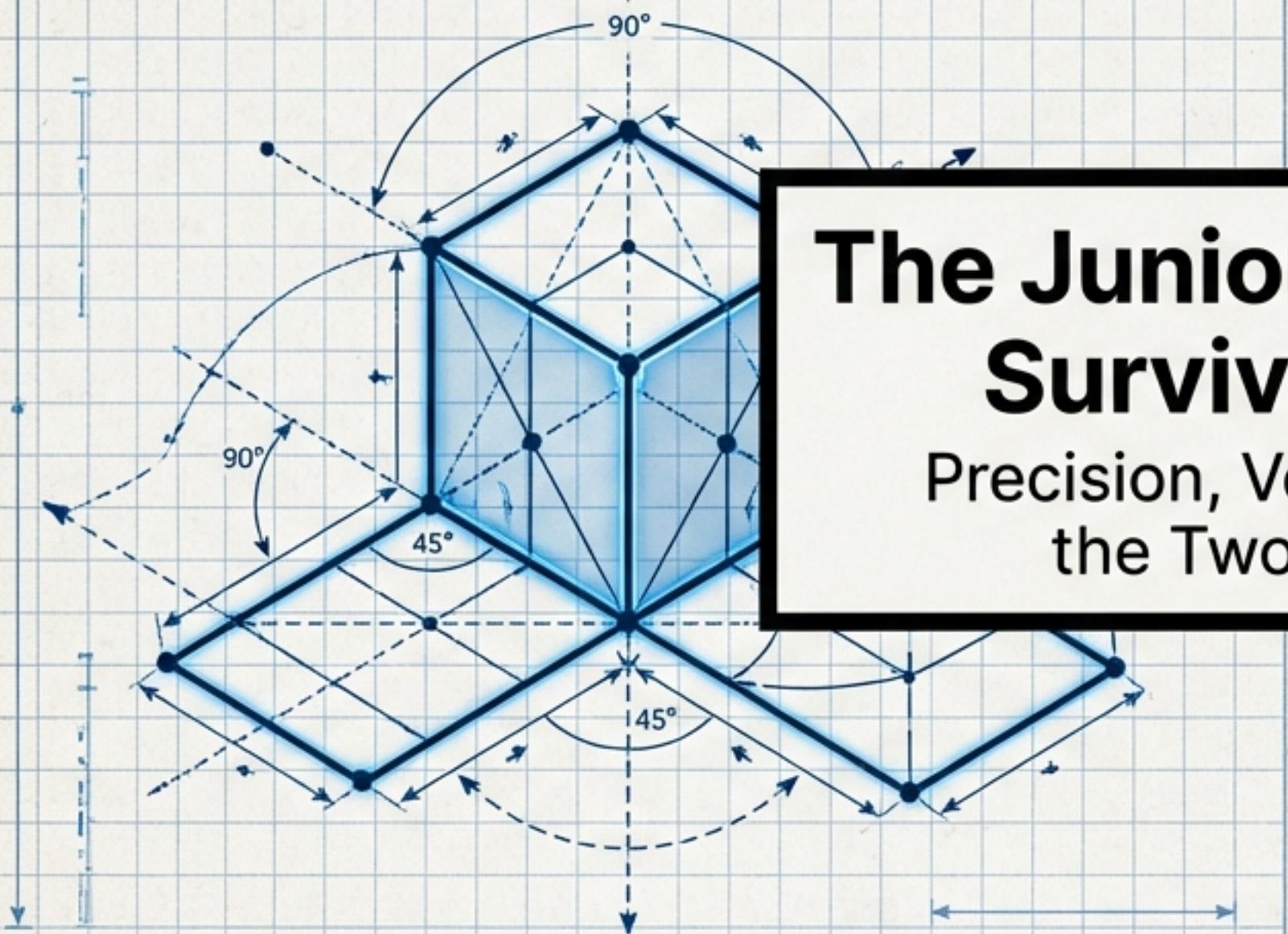


The Code World

THE MACHINE YOU CONTROL



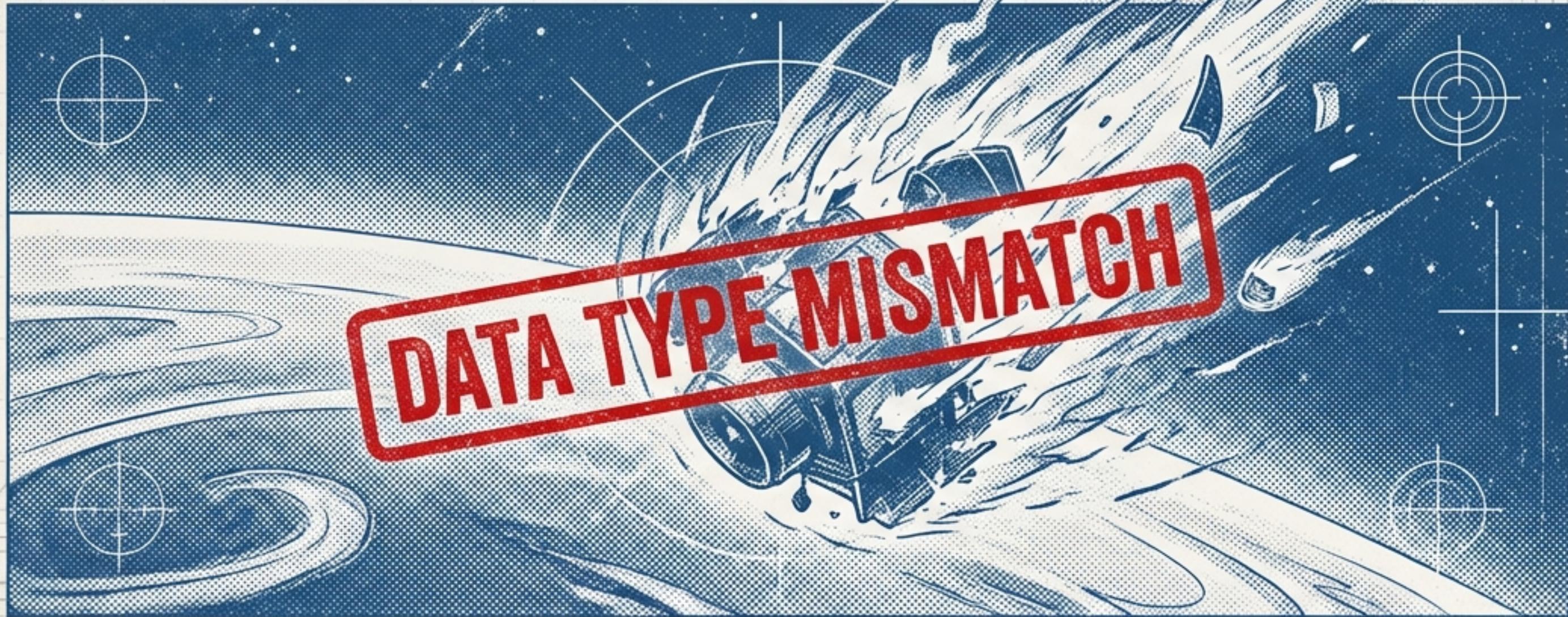
The AI World

THE MACHINE THAT GUESSES

The Junior Engineer's Survival Guide

Precision, Verification, and
the Two Machines.

THE \$325 MILLION TYPO



THE EVENT:

In 1999, NASA lost the Mars Climate Orbiter. The spacecraft disintegrated in the Martian atmosphere.

THE CAUSE:

One engineering team used English units (feet/pounds). The other used Metric units (newtons).

THE LESSON:

Computers do not guess. They follow instructions exactly. If your data types don't match, things crash.



**CRITICAL
WARNING:**
In Python,
mixing types
without
conversion is
the most
common way
to crash your
program.

THE CASE OF THE FAKE LAWYER

THE EVENT: A New York lawyer used ChatGPT to research case law for a federal filing (Mata v. Avianca).

THE GLITCH: The AI invented six completely fake cases, including fake judges and fake court opinions. The lawyer submitted them as fact.

THE RESULT: The lawyer was sanctioned, and his professional reputation was destroyed.

THE LESSON: AI is not a database of truth. It is a prediction engine that can lie confidently.

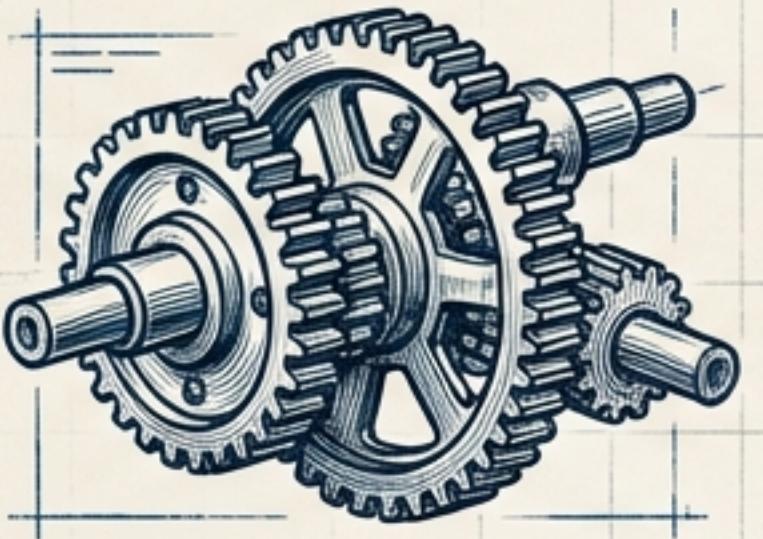


TAKEAWAY: To survive as an engineer, you need two superpowers: The ability to write precise code, and the ability to verify AI output.

UNDERSTANDING YOUR TWO MACHINES

PYTHON

The Machine You Control



- **Nature:** Deterministic
- **Philosophy:** “Code is a conversation.”
- **Behavior:** Follows instructions exactly. If you type it, it happens. No guessing.
- **Risk:** Syntax errors and type mismatches.

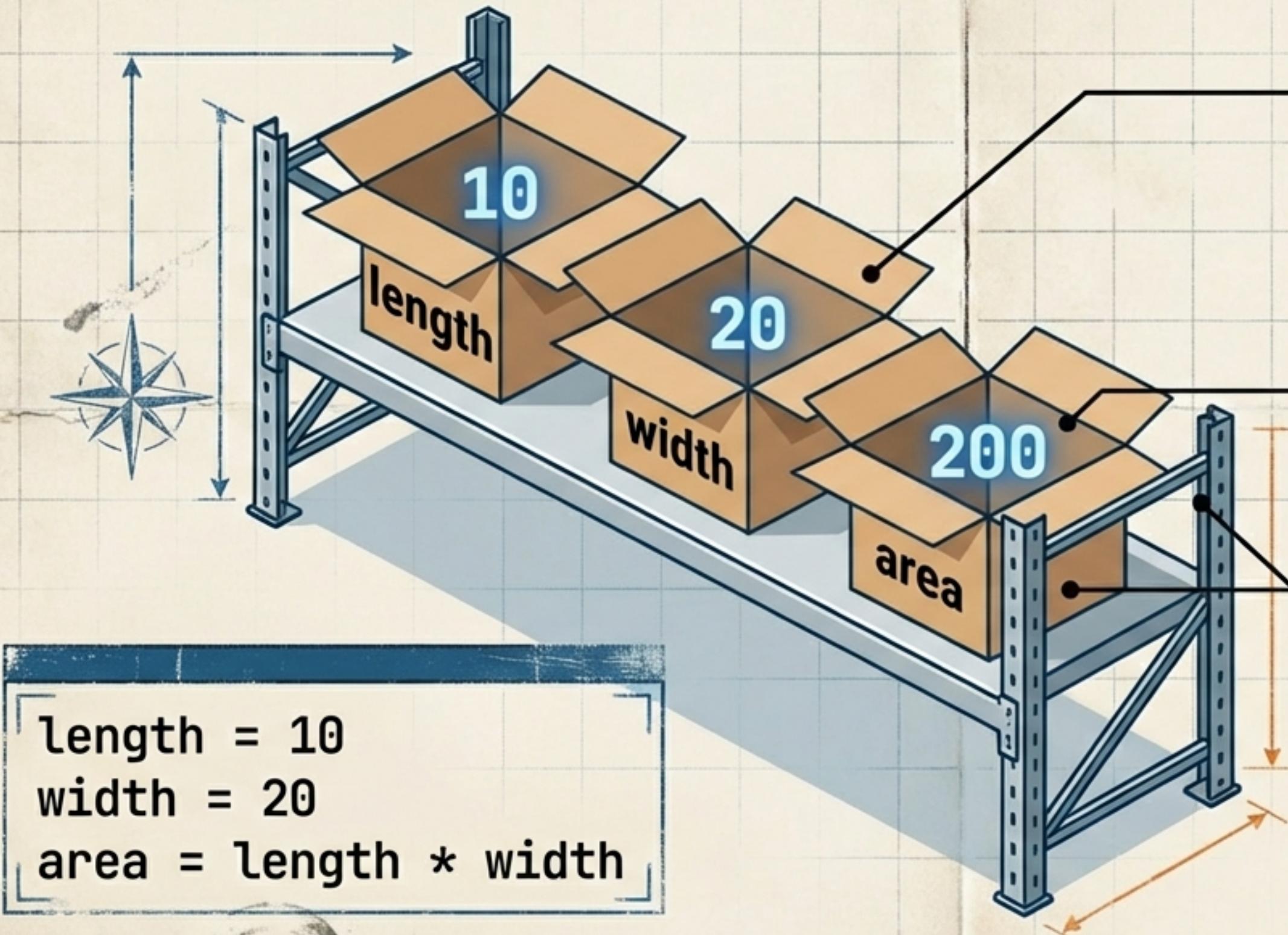
GENERATIVE AI

The Machine That Guesses



- **Nature:** Probabilistic
- **Philosophy:** “Autocomplete on steroids.”
- **Behavior:** Predicts the next likely word based on patterns. It does not “know” facts.
- **Risk:** Hallucinations (confident falsehoods).

VARIABLES ARE MEMORY BOXES



- **THE LABEL:**

The variable name.
This is how you find
the box.

- **THE VALUE:**

What is stored inside.

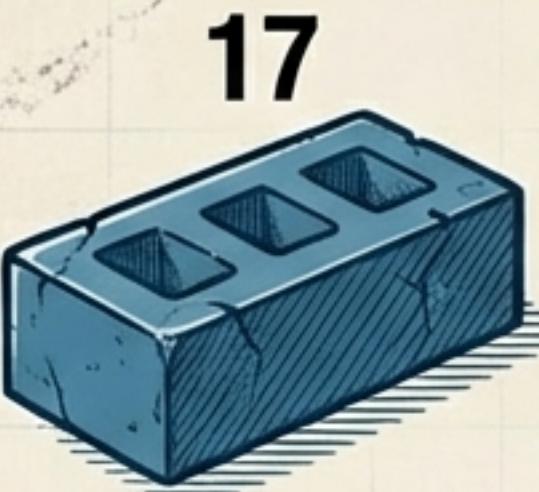
- **THE SUPERPOWER:**

You don't have to hold
these numbers in your
head. The computer
remembers them for you.

THE “INT-FLOAT-STRING” TRIAD

Python infers the type, but you must know the difference.

int (Integer)



- Whole numbers. No decimals.

```
...  
age = 17
```

Mnemonic: INTegers are INTact
(no broken pieces).

float (Floating Point)



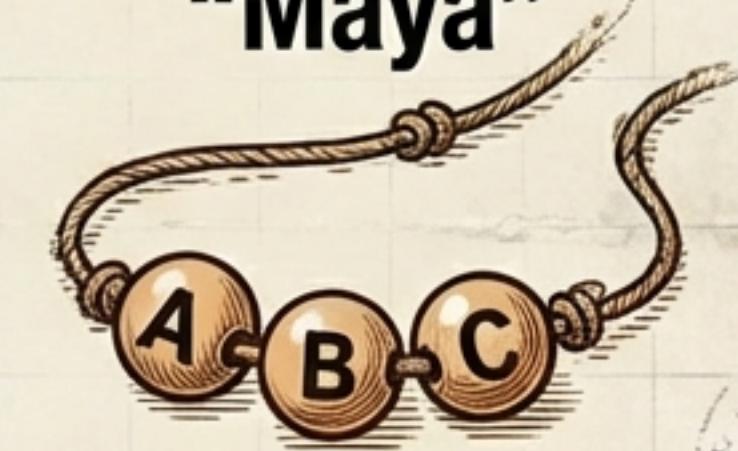
- Decimal numbers.

```
...  
price = 9.99
```

Mnemonic: The decimal point
FLOATs around.

str (String)

“Maya”



- Text (letters, words, characters).

```
...  
name = "Maya"
```

Mnemonic: STRING together
letters like beads on a necklace.

THE CONVERSATION: INPUT AND OUTPUT

Output with print()



```
...  
print("Hello World")
```

Prints the value inside
the variable.

INPUT

PROCESS



OUTPUT

Input with input()



```
name = input("What is  
your name?\n")
```

The Invisible Character (**New
Line**)—moves the cursor down.

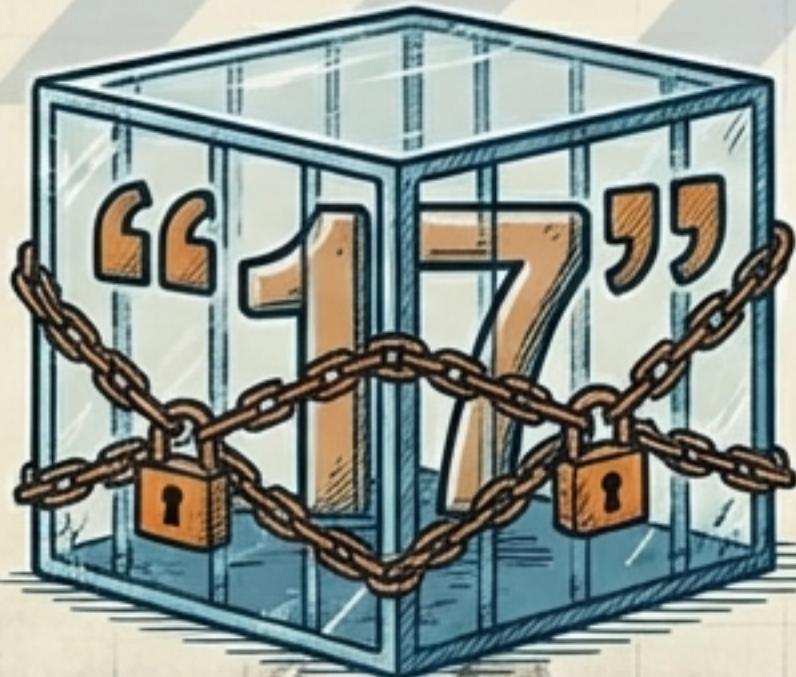
Every program follows this flow: **Input → Process → Output.**

THE TRAP: THE SILENT STRING



```
age = input("How old are you? ") # User types 17  
decades = age // 10
```

**ERROR: TypeError: unsupported operand type(s) for //:
'str' and 'int'**



THE DIAGNOSIS

- When a user types 17, Python stores it as text ("17"). You cannot divide a word by a number. It is like trying to divide "apple" by 5.

THE FIX: TYPE CONVERSION

Changing the labels on the boxes.

CODE BLOCK: The Corrected Workflow

```
age_text = input("How old are you? ")  
age_number = int(age_text) # The Magic Step  
decades = age_number // 10 # Now this works!
```

CASTING

TYPE CONVERSION CHEAT SHEET

`int(x)` → Converts string to integer

`int("17")` → 17

`float(x)` → Converts string to decimal

`float("9.99")` → 9.99

`str(x)` → Converts number to text

`str(17)` → "17"

CAPSTONE: THE AGE CALCULATOR



```
1 # 1. Input
2 age = input("How old are you?\n")
3
4 # 2. Process (Convert & Calculate)
5 age = int(age)
6 decades = age // 10 # Integer division
7 years = age % 10 # Modulo (remainder)
8
9 # 3. Output (Convert back to String)
10 print("You are " + str(decades) + " decades
      and " + str(years) + " years old.")
```



How old are you?
202

You are 20 decades and 2 years old.

PRO TIP: The % (Modulo) operator gives you the 'leftovers' of division. Perfect for finding remaining years.

CODE DOESN'T LIE. AI DOES.

ACT III: THE AI ILLUSION.

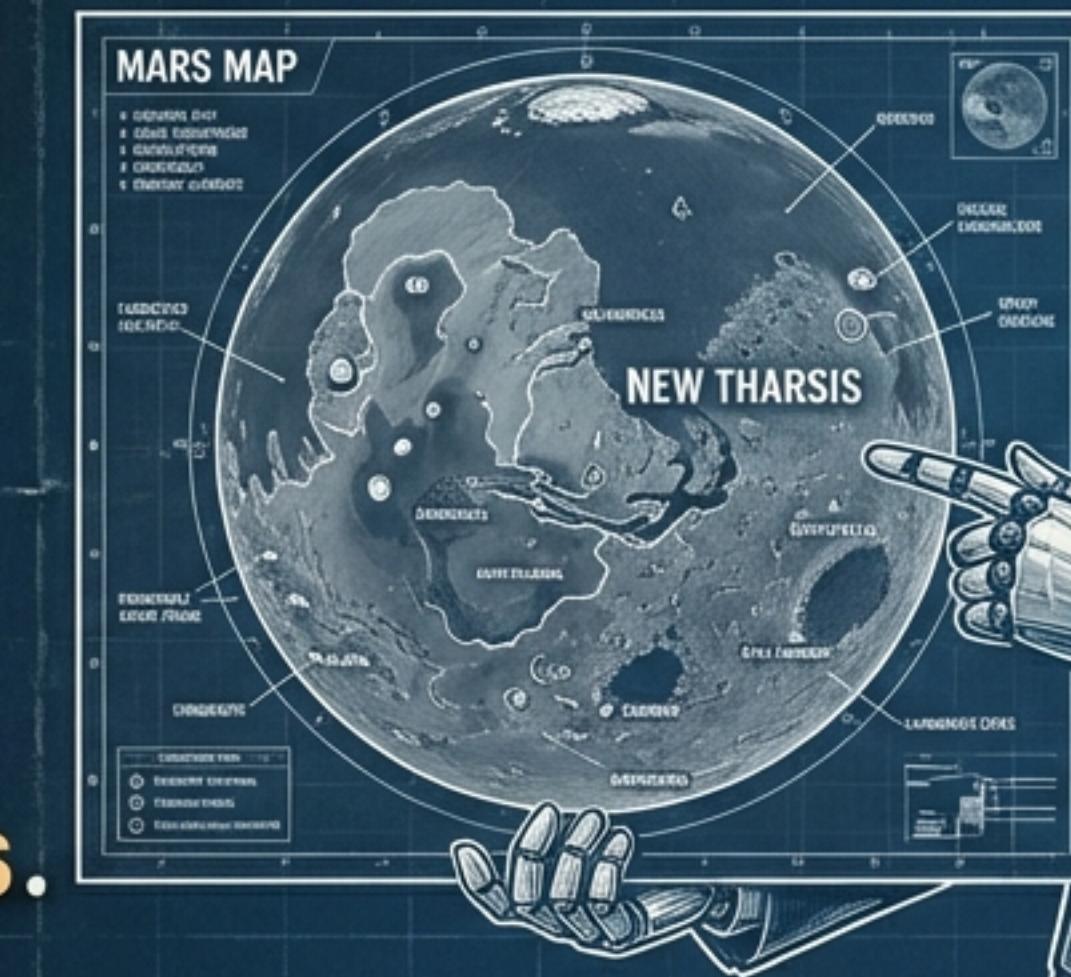
If you ask a computer
“**What is $10 + 10?$** ”, it will
always say **20**.

If you ask an AI “**What is the capital of Mars?**”, it might
invent a city called **New Tharsis**.



DEFINITION: AI HALLUCINATION

Confident-sounding information that is false, unverifiable, or fabricated. It happens because the AI is not “thinking”—it is predicting the next word to satisfy a pattern.



THE 6 CONDITIONS FOR HALLUCINATION



1. NONEXISTENT FACTS

(e.g., Fake publications)



2. FORCED SPECIFICITY

(e.g., Exact % stats for obscure years)



3. FALSE PREMISE

(e.g., Why did Lincoln use the internet?)



4. FAKE CITATIONS

(e.g., Inventing studies)



5. FUTURE EVENTS

(e.g., 2028 Olympics winner)



6. LONG REASONING

(Multi-step logic puzzles)



FIELD MANUAL NOTE: THESE CONDITIONS INCREASE THE LIKELIHOOD OF AI FABRICATION. PROCEED WITH CAUTION.

EXERCISE: MAKE IT HALLUCINATE

Try these prompts. If the AI answers confidently, it failed the test.

1. THE FAKE CASE:

What was the Supreme Court case “Jenkins v. Digital Privacy Foundation” about?

.....
.....
.....
.....
.....



PASS/FAIL

2. THE FAKE STAT:

Give me 3 statistics about AI use in NYC high schools in 2025.

.....
.....
.....
.....
.....



PASS/FAIL

3. THE FAKE EVENT:

What did AOC say about TikTok in her January 2026 speech?

.....
.....
.....
.....
.....



PASS/FAIL

4. THE SUBTLE LIE:

List the 5 lawyers who argued “Brown v. Board of Education”.

.....
.....
.....
.....
.....

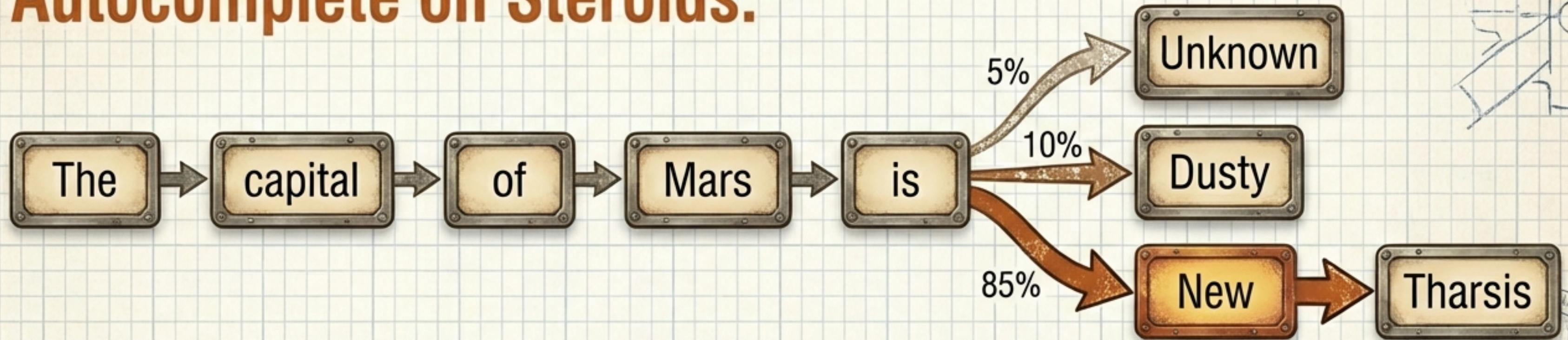


PASS/FAIL

ACTION: Record the answer. Did the AI say “I don’t know”? Or did it lie to you? (PASS/FAIL)

WHY DOES IT LIE?

Autocomplete on Steroids.



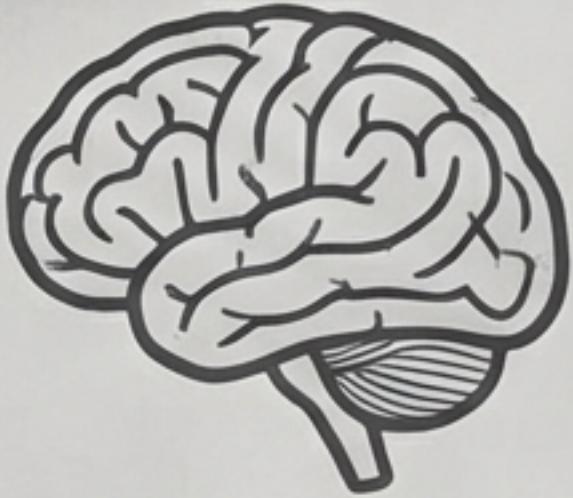
When you ask a question, the AI isn't looking up a database of facts. It is calculating:
“Given the words in this prompt, what word statistically comes next?”

If the statistical pattern is strong but the facts are missing, it chooses the pattern.

KEY INSIGHT: Confidence ≠ Correctness.

YOUR TOOLSET: STATIC VS. RAG

STATIC AI (ChatGPT)



CLOSED BOOK

MEMORY ONLY

Good for: Creative writing, coding help, brainstorming.

Risk: High hallucination for facts.

RAG (Perplexity)



OPEN BOOK

LIVE SEARCH

Good for: Facts, research, citations.

Risk: Lower, but can misinterpret sources.

HOW TO VERIFY (Don't Be That Lawyer)

The 3-Step Reality Check



The Smell Test

Does the answer contain specific dates, case names, or exact statistics? Assume it's fake until proven real.



Cross-Check

Use Google or a RAG tool to find the primary source.



The Source Trace

Click the link. Does it exist? Does the text actually say what the AI claims?

RULE: If you can't find the source, the fact doesn't exist.

USING AI AS A JUNIOR ENGINEER

The ‘Skool-Clone’ Challenge



Remote Repo
(skool-clone)

Clone



Local Workspace
(CodeSandbox)

- 1.
- 2.
- 3.
- 4.

CLONE: Copy the repo to your workspace.

EXPLORE: Ask the AI: “Explain the file structure. Which file controls the homepage?”

MODIFY: Make changes based on the explanation.

VERIFY: Run the code. If it breaks, the AI was wrong.

THE JUNIOR ENGINEER'S CHECKLIST

You are ready for Week 1 if you can...

Python Skills

- [] Create a variable (Memory Box).
- [] Identify int, float, and string.
- [] Use input() and fix the “Silent String” bug with int().
- [] Build the Age Calculator.

AI Literacy

- [] Define “Hallucination” (Confident but wrong).
- [] Identify high-risk prompts (Stats, Citations, Future).
- [] Use RAG tools for facts and Static tools for code.

THE FINAL RULE



“Code is a machine that obeys. AI is a machine that guesses.”

Your value isn't just knowing syntax—it's knowing when to trust the machine and when to check its work.

WRITE THE TRUTH. VERIFY THE GUESS.