



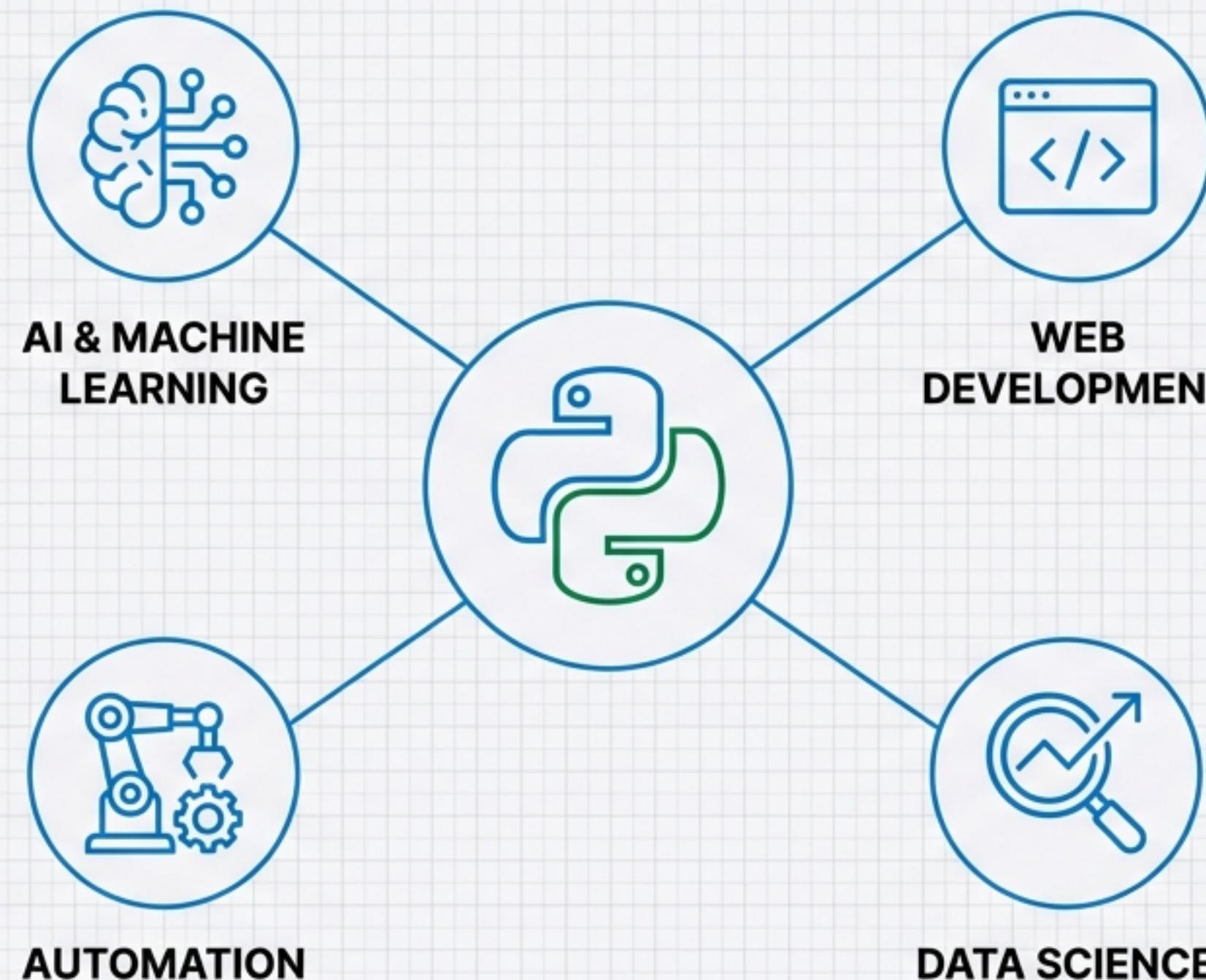
PYTHON MASTERY

From Zero to Hero.

A comprehensive blueprint for building logic, automation, and AI.

Curriculum based on the expertise of Mosh Hamedani

THE PYTHON ADVANTAGE



INDUSTRY ADOPTION
Google, Spotify, Dropbox, Facebook

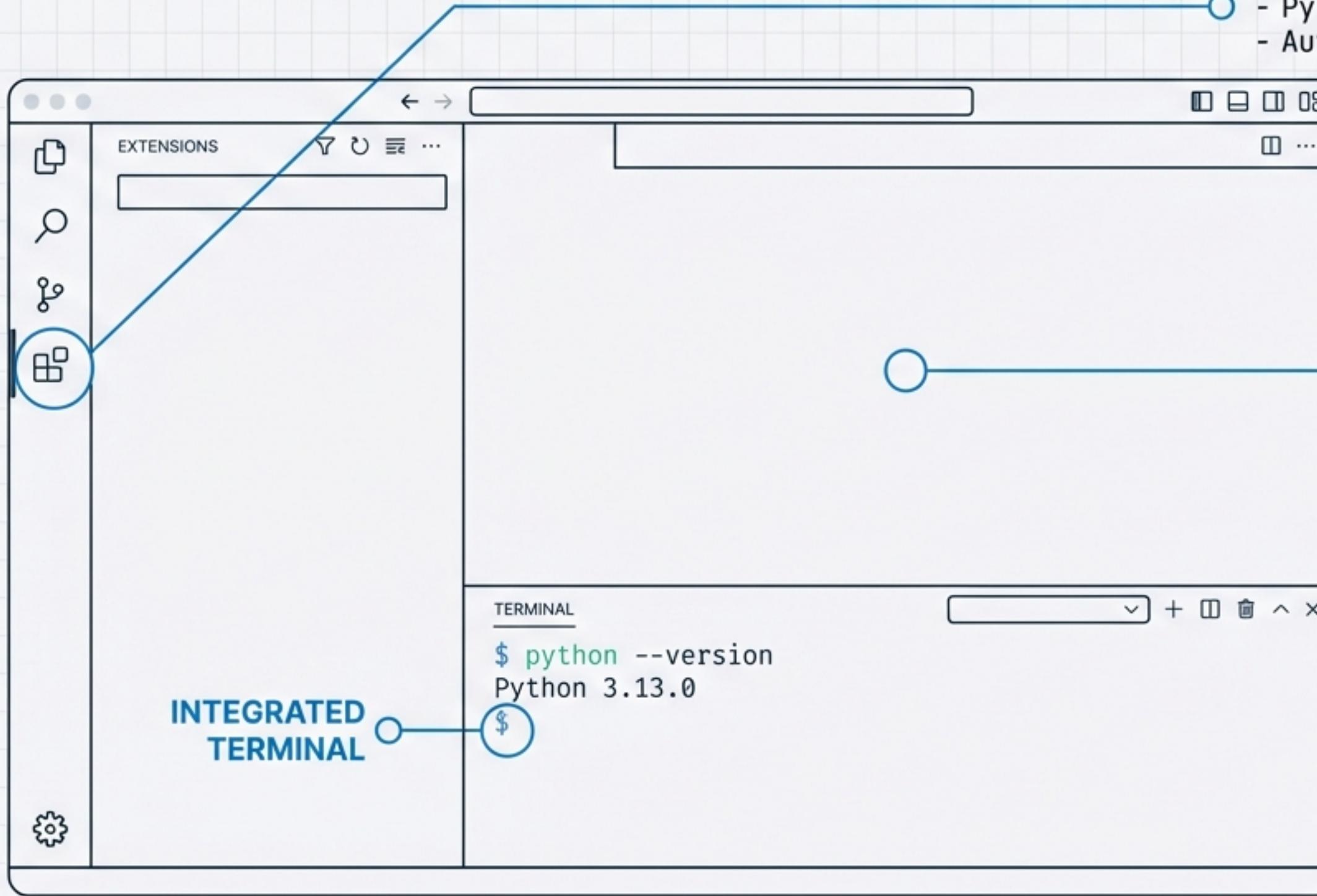
ROI
Average US Salary: \$115,000+

EFFICIENCY



Solves complex problems with fewer lines of code

THE WORKSHOP: ENVIRONMENT & TOOLS



EXTENSIONS PANEL: THE SECRET WEAPON

- Python (Microsoft)
- Pylint (Error Checking)
- Autopep8 (Auto Formatting)

THE EDITOR

Optimized for speed and debugging.

CAUTION

Windows Users: Ensure "Add Python to Path" is checked during installation.

Storing Data: Variables & Naming

Code is read more often than it is written

Bad Practice (Mystical)

```
1 cn = "Python"
2 x = 1000
```

Ambiguous names

No context

Clean Code (PEP 8)

```
1 course_name = "Python Mastery"
2 student_count = 1000
```

Descriptive

Lowercase

Underscores

Spaces around =

Variables are case-sensitive labels for memory locations.
Always prioritize readability.

The Raw Materials: Primitive Types

Integer

```
1 count = 1000
```

JetBrains Mono

Whole numbers.

Float

```
1 rating = 4.99
```

JetBrains Mono

Numbers with decimals.

Dynamic Typing

JetBrains Mono

Python determines
the type automatically
at runtime.

Boolean

```
1 is_published = True
```

JetBrains Mono

Logic states. Must be Capitalized
(True/False).

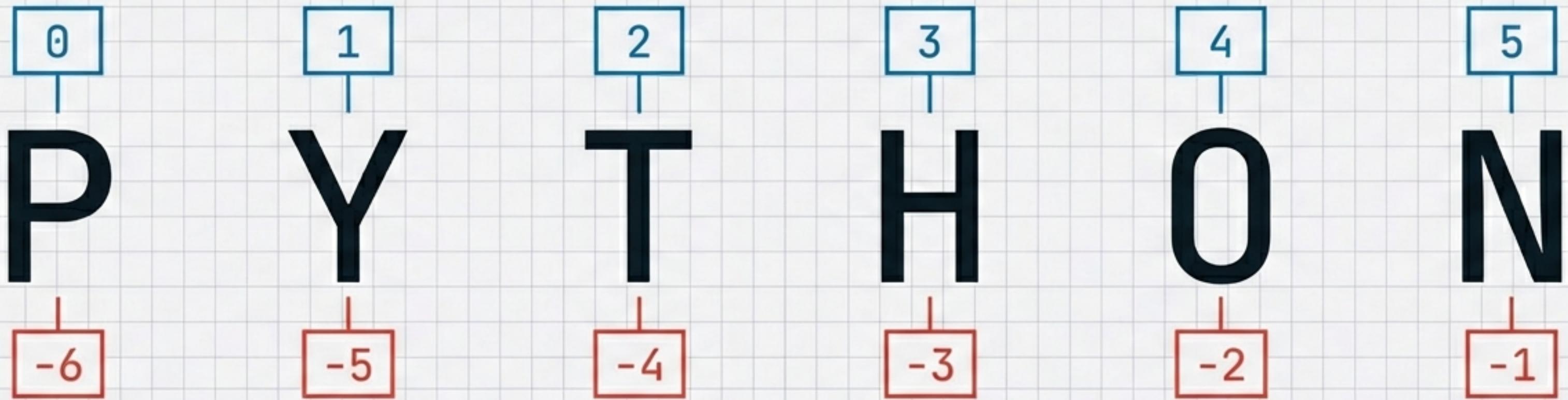
String

```
1 course = "Python"
```

JetBrains Mono

Text sequences in quotes.

Text Manipulation: Strings & Slicing



```
1 course[0] -> Returns "P"
```



```
1 course[-1] -> Returns "n"
```



```
1 course[-1] -> Returns "n"
```



```
1 course[0:3] -> Returns "Pyt"
```

(End index 3 is excluded)

Note: Escape Sequences: `\"` for quotes, `\\n` for new lines.

Formatting & Methods

f-Strings (The Modern Way)

```
1 first = "Mosh"  
2 message = f"Hello {first}"
```

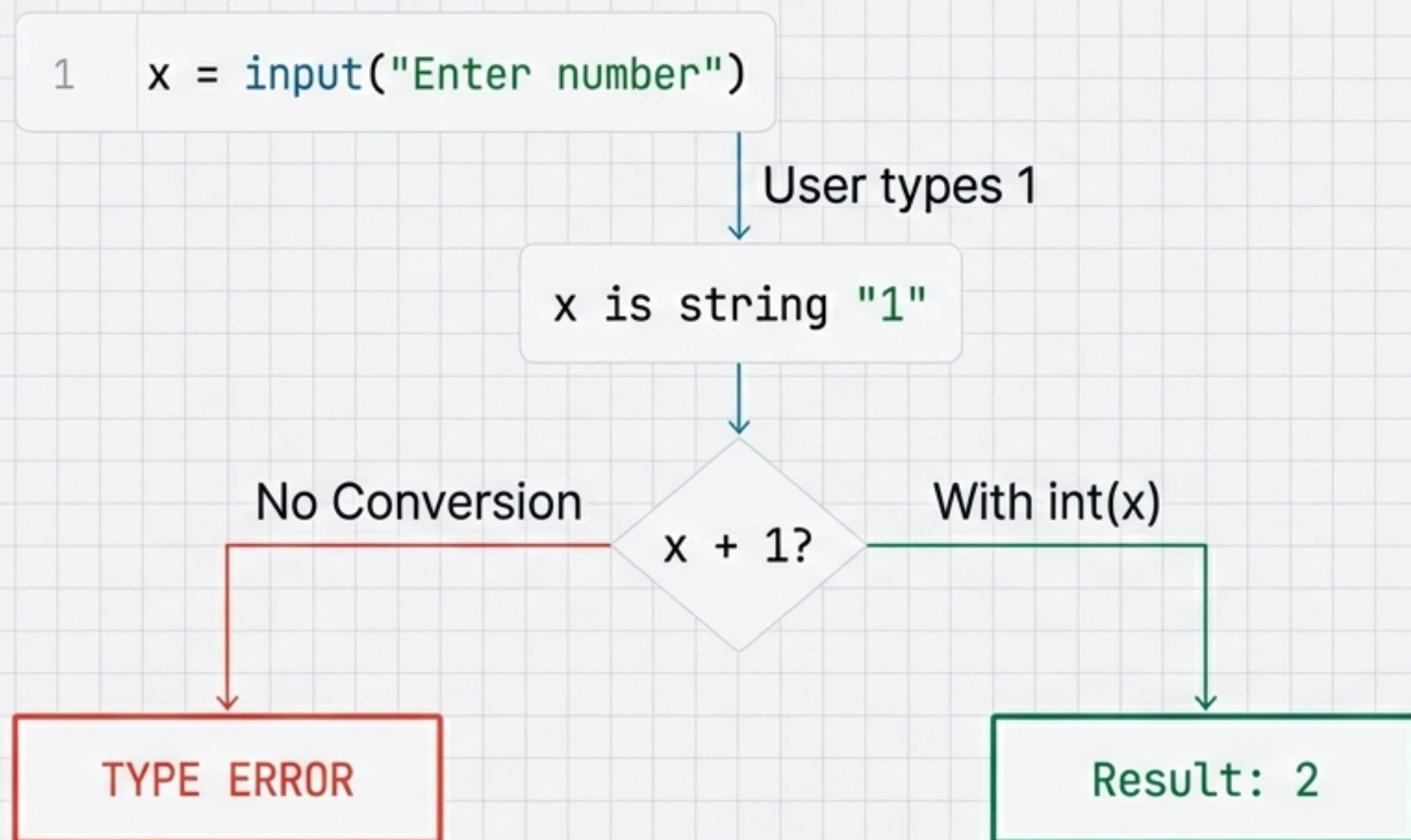
Dynamic
Expression

Essential Methods

- .upper() → "PYTHON" (Case conversion)
- .strip() → Removes whitespace (Critical for input)
- .find("pro") → Returns Index or -1
- .replace("p", "j") → Swaps characters
- "Pro" in course → Returns Boolean (True/False)

Input & Type Conversion

The Input Trap



Falsy Values

Python treats these as False:

- 0
- Empty Strings
- None

Control Flow: Conditional Logic

Standard Structure



```
1 if temperature > 30:  
2     print("It's warm")  
3 elif temperature > 20:  
4     print("It's nice")  
5 else:  
6     print("It's cold")
```



→ Crucial: Semantic Indentation

The Ternary Operator

Cleaner logic for simple assignments.



```
1 message = "Eligible" if age >= 18 else "Not Eligible"
```

Boolean Logic & Comparisons

Operators

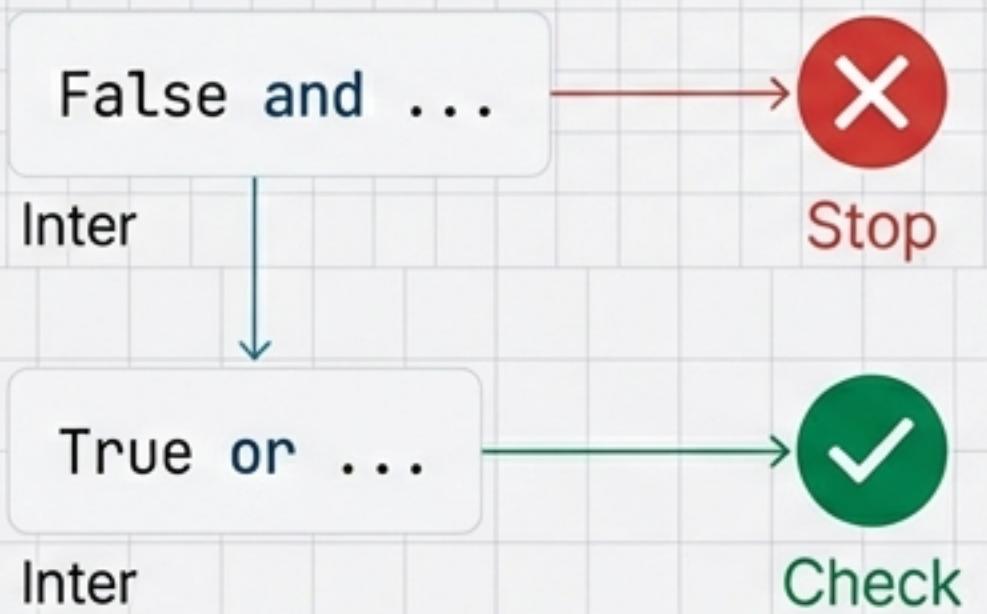
and Both True

or At least one True

not Inverse

JetBrains Mono

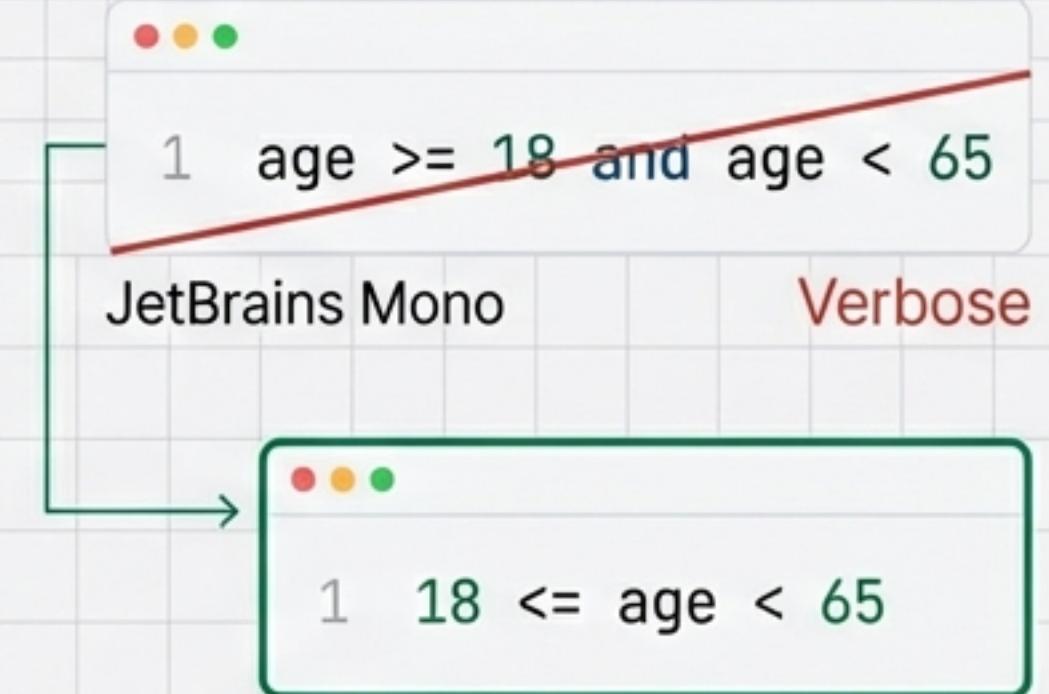
Short Circuiting



Stops evaluation as soon as result is determined.

JetBrains Mono

Chaining Comparison



JetBrains Mono Pythonic

The Engine of Repetition: Loops

For Loops (Sequences)

```
1 for number in range(1, 10, 2):
```

range(start, end, step) → [1 | 3 | 5 | 7 | 9]

Iterates over Strings, Lists, or Ranges.

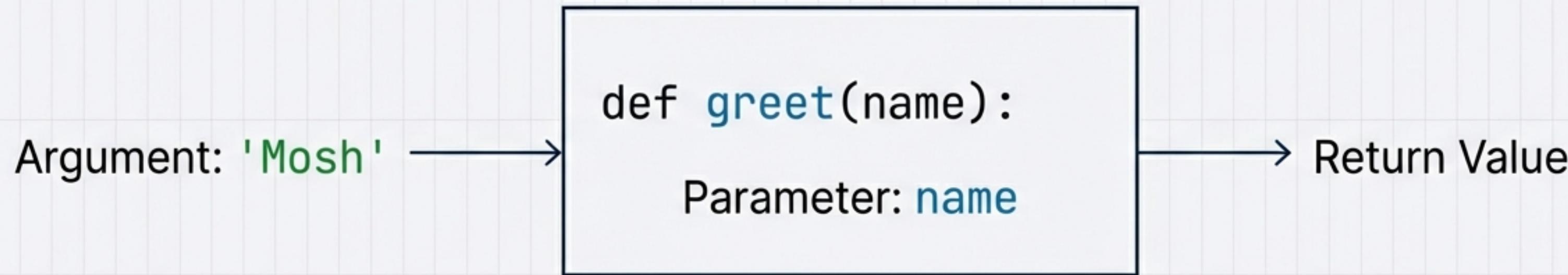
While Loops (Conditions)

```
1 while x < 5:  
2     print(x)  
3     x += 1
```

Repeats as long as condition is True.

Note: `break` terminates a loop. `else` runs if loop finishes without break.

The Blueprint: Functions



Parameters vs. Arguments

Parameter is the variable definition.
Argument is the actual value passed.

Return vs. Print

Printing displays to console.
Returning passes value back to the system.

Note: By default, functions return `None`.

Advanced Function Flexibility

Keyword Arguments

```
1 increment(2, by=1)
```

Improves readability for unclear arguments.

Default Arguments

```
1 def increment(number, by=1):
```

Makes 'by' optional.
Required parameters come first.

Variable Arguments (*args)

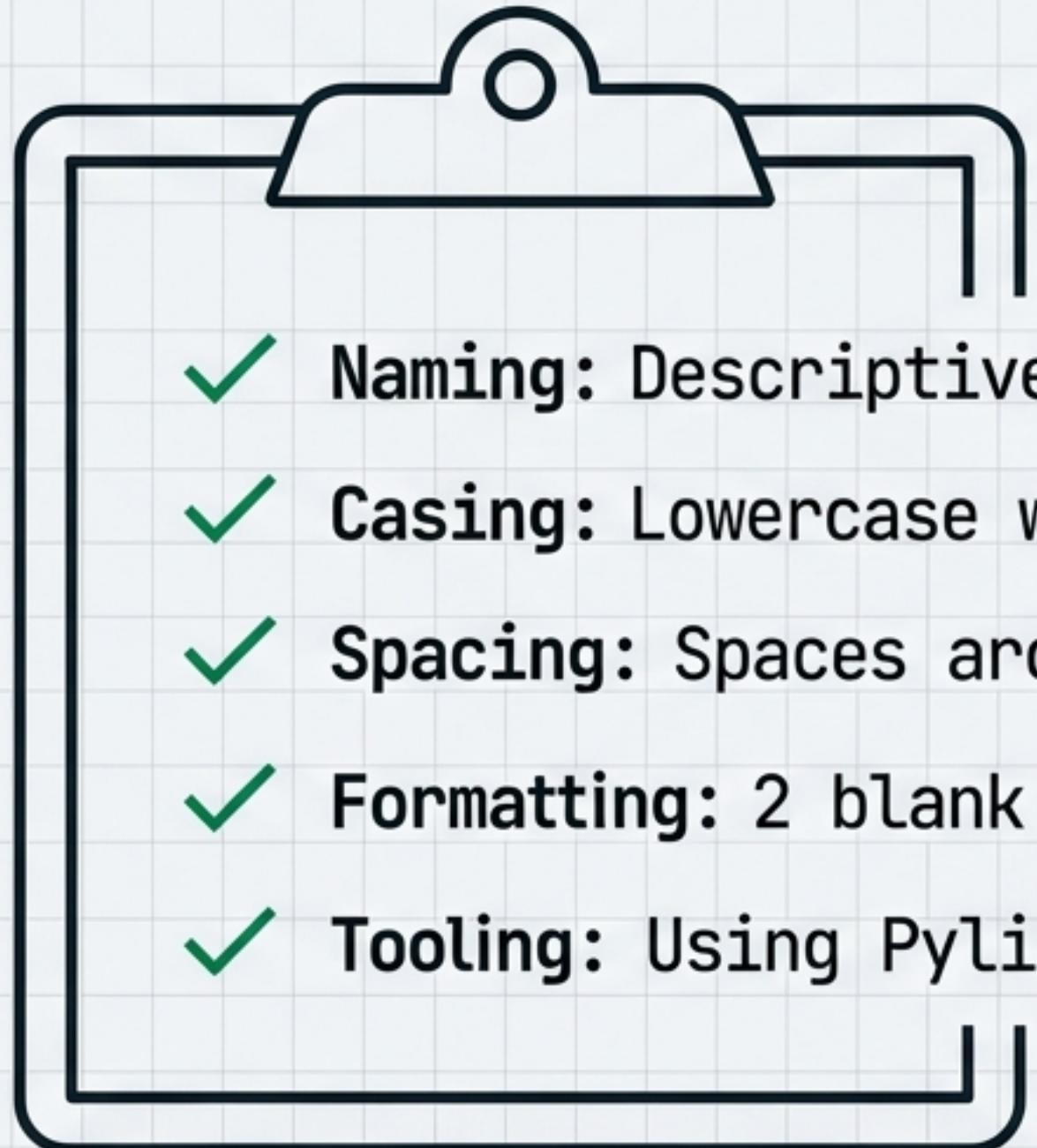
```
1 def multiply(*numbers):
```



Packs arguments into a Tuple (2, 3, 4, 5)

Allows iterating over any number of inputs.

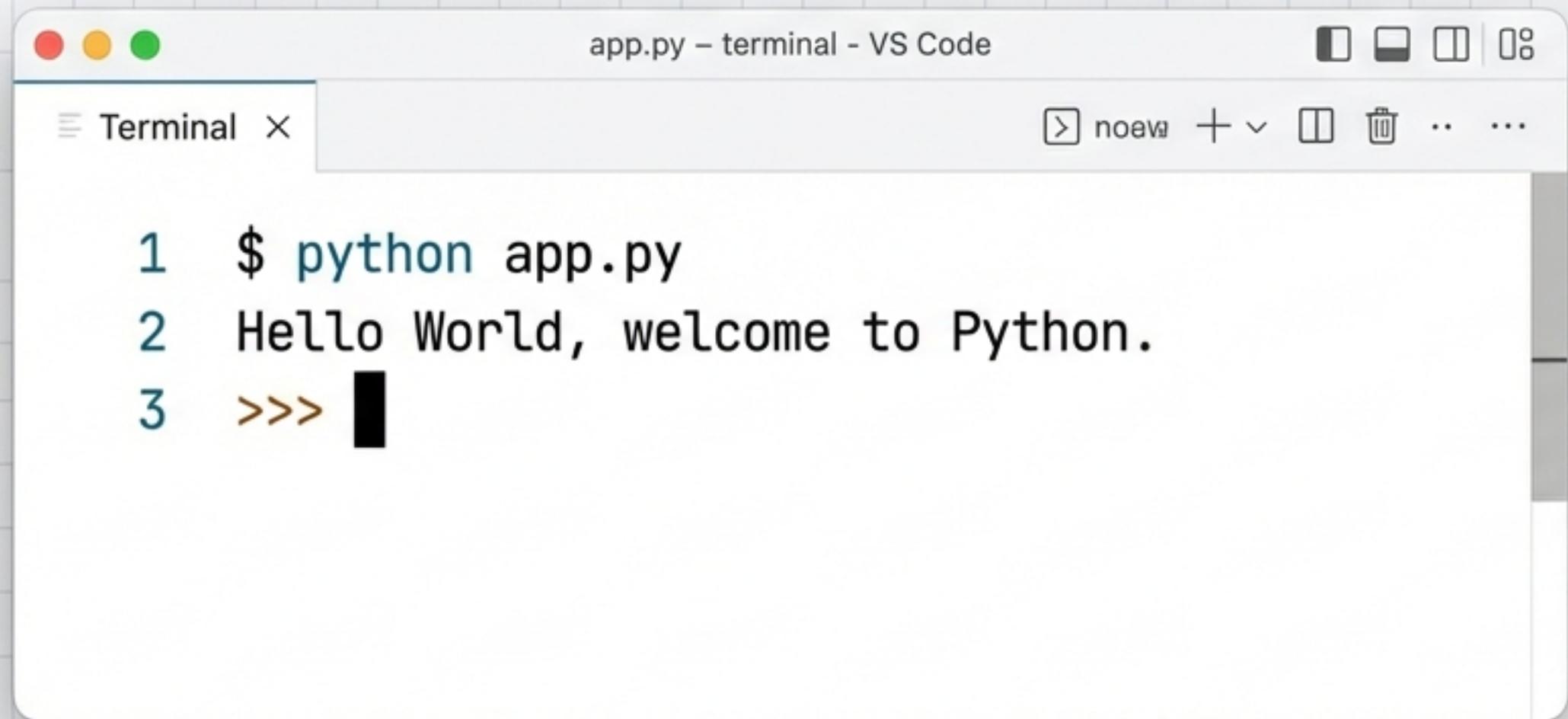
The “Clean Code” Philosophy (PEP 8)



- ✓ **Naming:** Descriptive and meaningful (`student_count`, not `sc`).
- ✓ **Casing:** Lowercase with underscores (`my_function`).
- ✓ **Spacing:** Spaces around operators (`x = 1`).
- ✓ **Formatting:** 2 blank lines after function definitions.
- ✓ **Tooling:** Using Pylint and Autopep8.

“Format code like a newspaper article—scannable and structured.”

Your Python Journey Begins



A screenshot of the VS Code interface showing a terminal window. The title bar says "app.py - terminal - VS Code". The terminal tab is active. The content of the terminal shows the following text:

```
1 $ python app.py
2 Hello World, welcome to Python.
3 >>> █
```

Next Steps

1. Move from Shell to `.py` applications.
2. Practice building reusable Functions.
3. Explore Modules (`import math`).

You don't need prior knowledge to start,
but you need persistence to become a Hero.