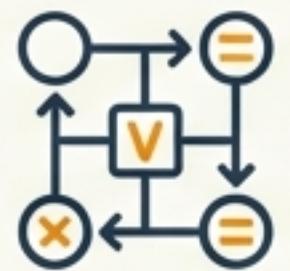
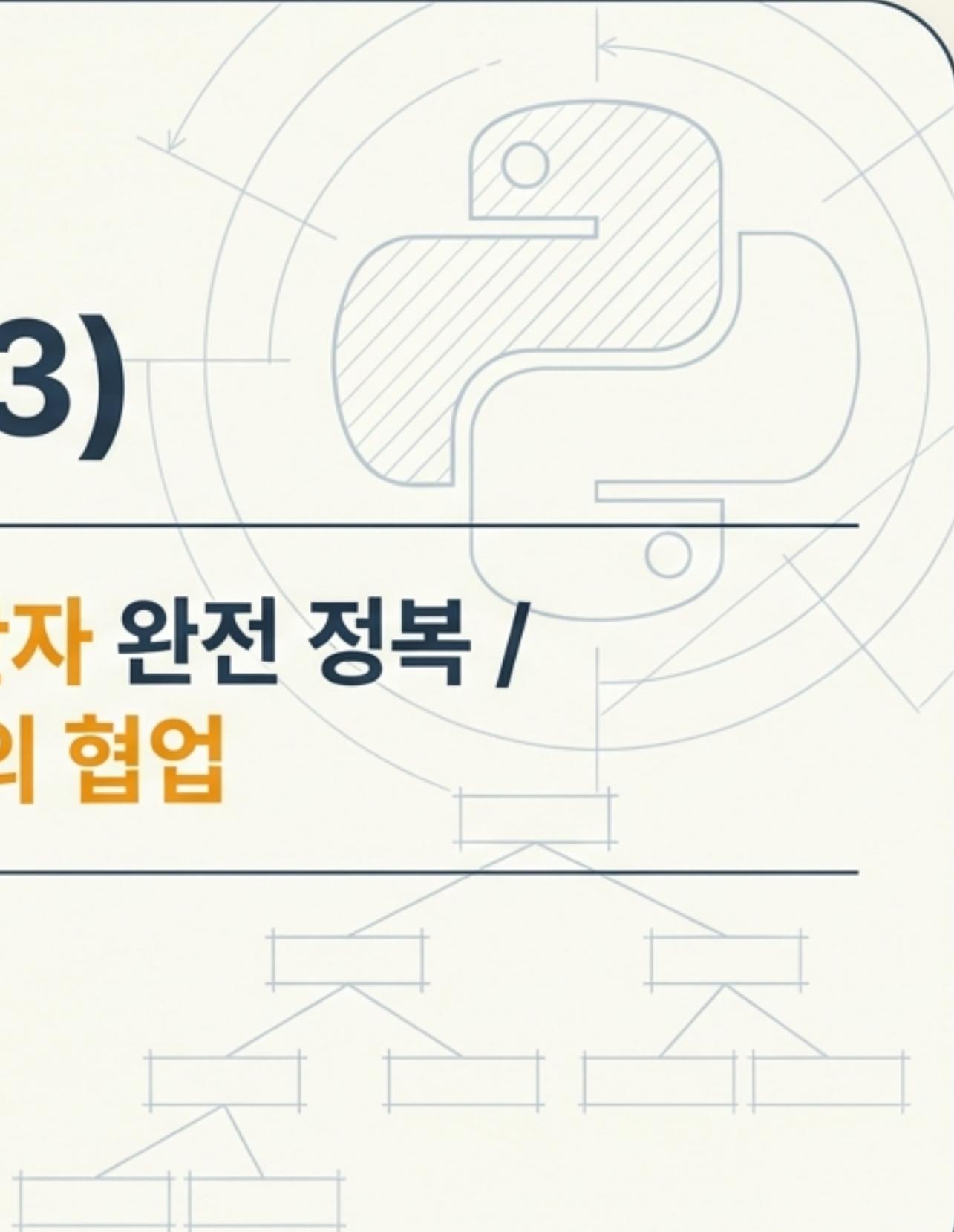


AI 활용 프로그래밍: Python 기초 (Week 3)

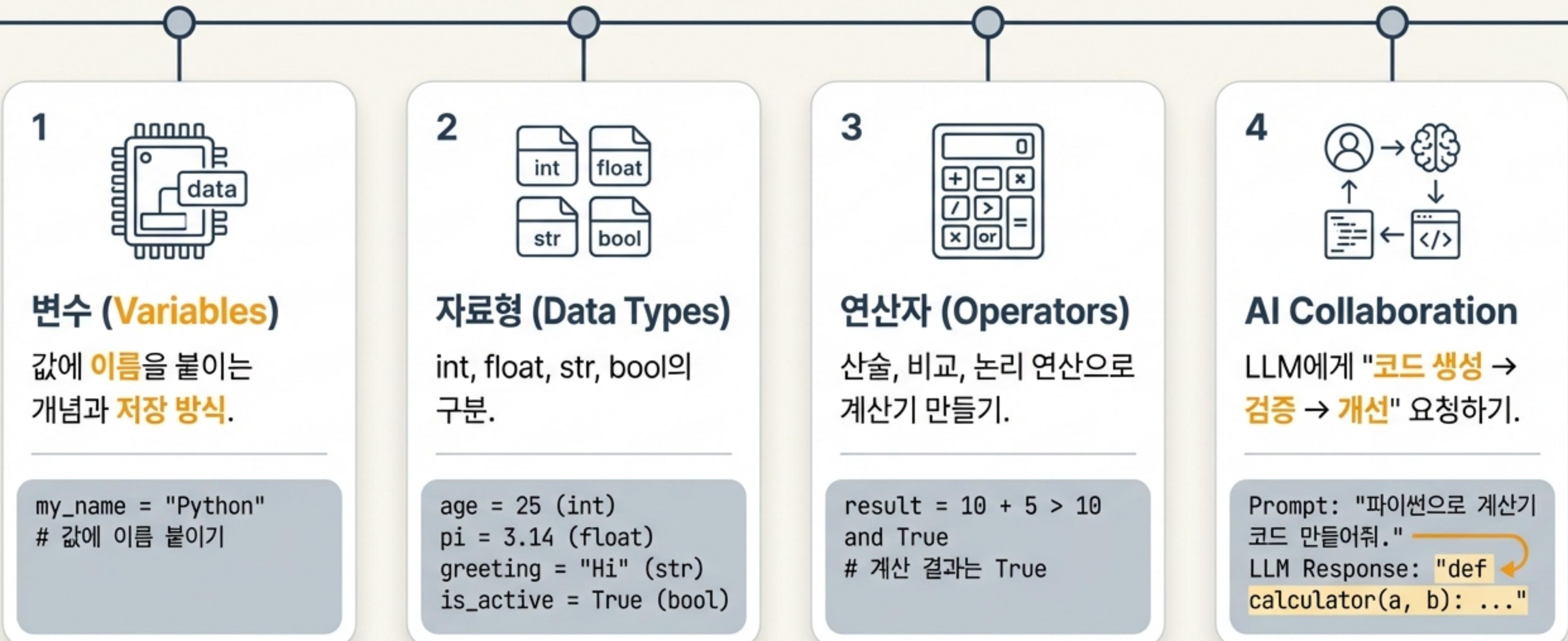


변수, 자료형, 그리고 연산자 완전 정복 /
기본 문법 다지기 & AI와의 협업

```
# Python Basics - Week 3
user_age = 25
print(f"AI와의 협업: {user_age}세부터!")
```



오늘의 학습 목표 (Learning Roadmap)



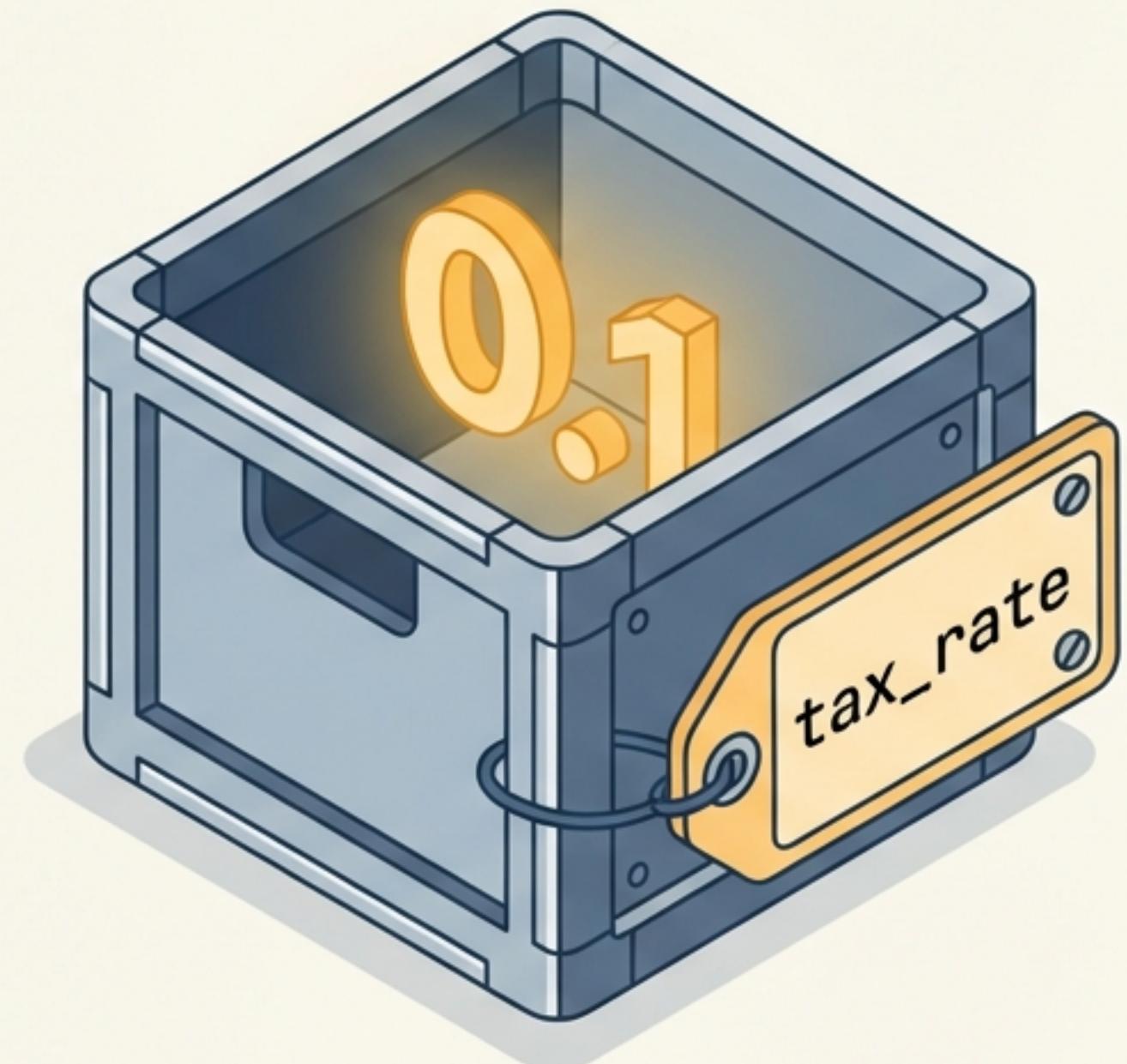
변수(Variable): 값을 부르는 '이름표'

변수는 값을 저장하는 공간인 동시에,
메모리에 있는 값에 붙인 "라벨(Label)"입니다.

값을 재사용하고 수정하기 쉽게 만듭니다.

```
tax_rate = 0.1
```

세율이 바뀌면 이 숫자만 바꾸면 해결됨



변수 이름 규칙 & 스타일 (Naming Convention)

Rule (규칙)

- ✓ 영문, 숫자, 언더스코어(_) 사용 가능.
- ⚠ 주의: 숫자로 시작할 수 없음.
- ✗ 금지: 예약어 (if, for, class 등) 사용 불가.

Style (스타일)



Best Practice

- ✓ 의미 있는 이름 사용 (`a` vs `total_score`).
- ✓ Snake Case 사용 (`my_variable_name`).

= 은 '같다'가 아닙니다



= (대입/할당)

오른쪽의 값을 왼쪽 변수에 넣는다 (Assign).

== (비교)

양쪽의 값이 같은지 확인한다 (Equal).

```
name = "Kim"      # "Kim"을 name에 할당  
age = 20         # 20을 age에 할당  
age = age + 1   # 기존 age(20)에 1을 더해 다시 할당 (21)
```



Key Takeaway: 변수는 "바뀔 수 있는 값"입니다.

기본 자료형 한눈에 보기 (Data Types)



int (정수)

10, -3

(소수점 없음)



float (실수)

3.14, 0.1

(소수점 있음)



str (문자열)

"Hello", "10"

(따옴표 필수)



bool (논리)

True, False

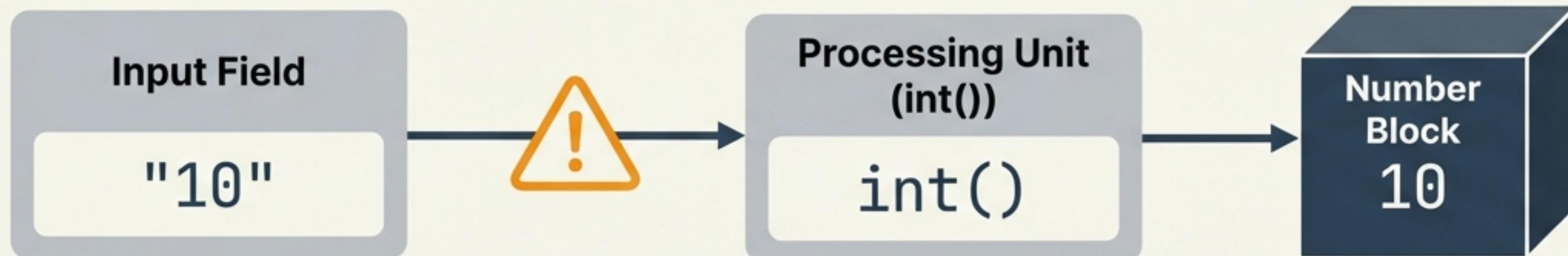
(참/거짓)



Pro Tip: `type(x)` 명령어로 현재 값의 자료형을 확인할 수 있습니다.

입력(Input)과 형변환(Casting)

`input()`으로 받은 데이터는 무조건 문자열(str)입니다.



Warning: "10" (문자열)은 숫자가 아닙니다. 바로 계산하면 오류가 발생합니다.

Solution Pattern:

```
# 입력받자마자 정수로 변환하는 패턴  
num = int(input("숫자 입력: "))
```

f-string: 가장 세련된 출력 방법



Syntax:

```
f"텍스트 {변수} 텍스트"
```

문자열과 변수를 직관적으로 연결할 수 있어 보편적으로 사용됩니다.

Examples:

```
name = "Kim"; score = 92
print(f"{name}님의 점수는 {score}점입니다.")
```

Output:

Kim님의 점수는 92점입니다.

```
print(f"합격 여부: {score >= 60}")
```

Output:

합격 여부: True

산술 연산자: 계산의 기초

기본 (Basic)

+ , - , * , / (결과는 항상 float)

핵심 연산자 (Focus):



(몫): 나눗셈의 정수 부분



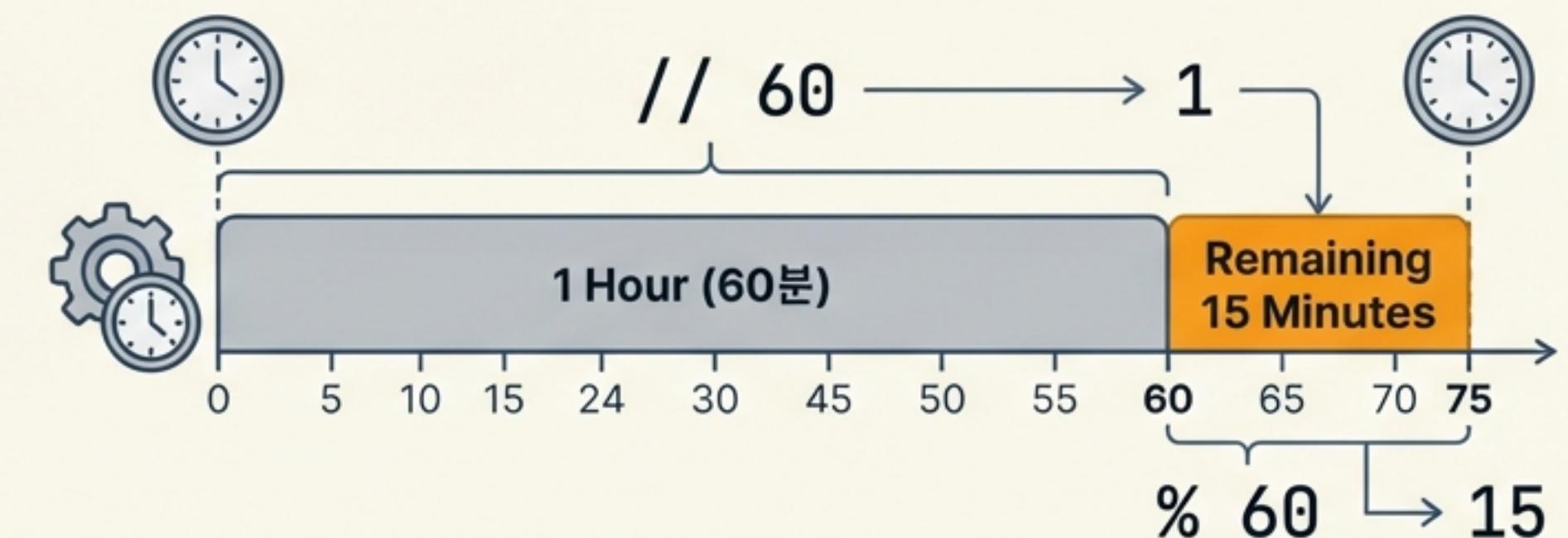
(나머지): 나눗셈의 나머지 부분

활용 예시 (Time Conversion):

75분은 몇 시간 몇 분인가?

$$75 \text{ } // \text{ } 60 \rightarrow 1 \text{ (시간)}$$

$$75 \text{ } \% \text{ } 60 \rightarrow 15 \text{ (분)}$$



비교와 논리: True or False?

비교 연산자

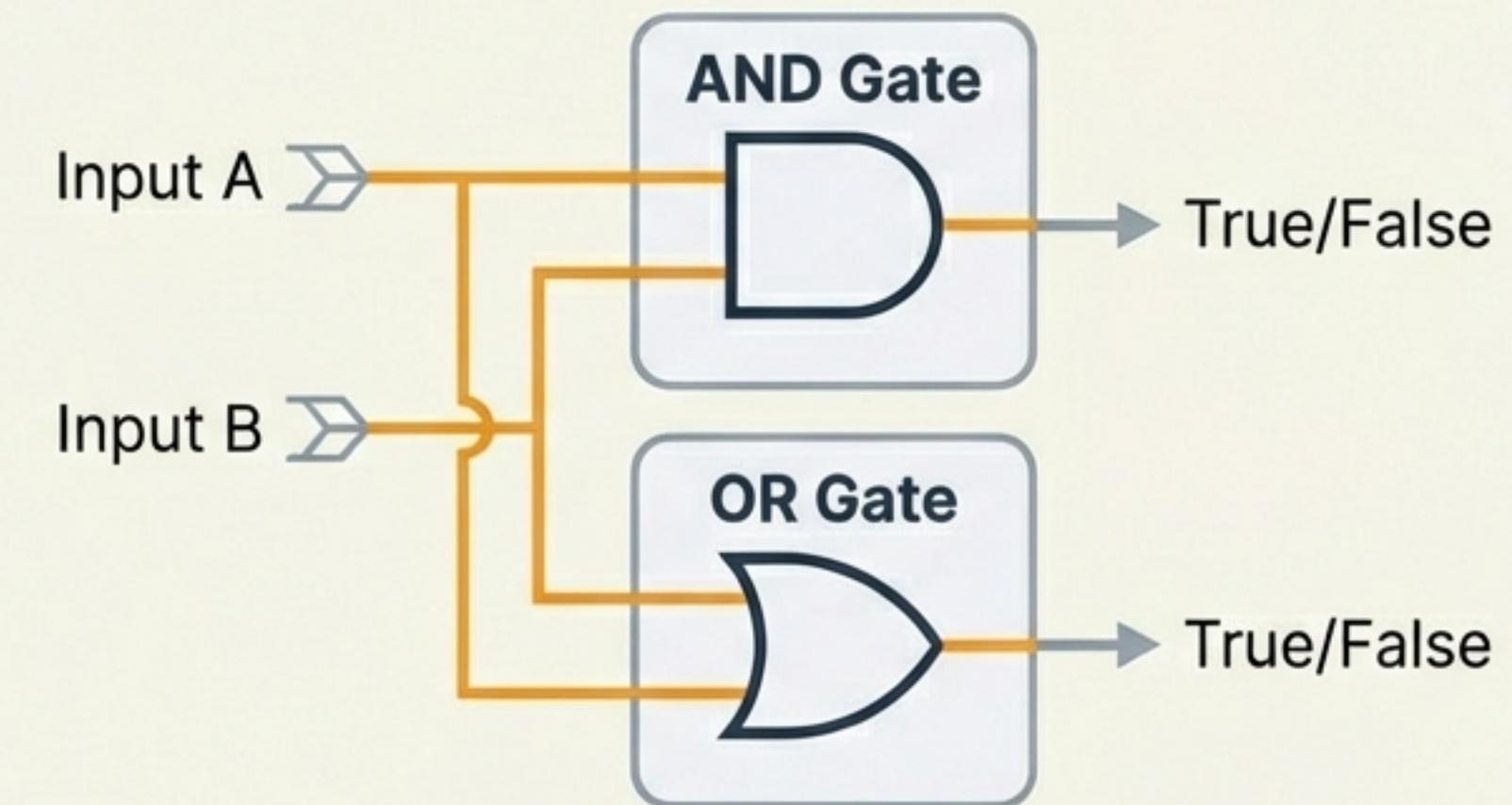
> (크다), < (작다), \geq (크거나 같다), \leq (작거나 같다), == (같다), != (다르다)

논리 연산자

and (모두 참일 때 True)

or (하나라도 참일 때 True)

not (결과 반전)

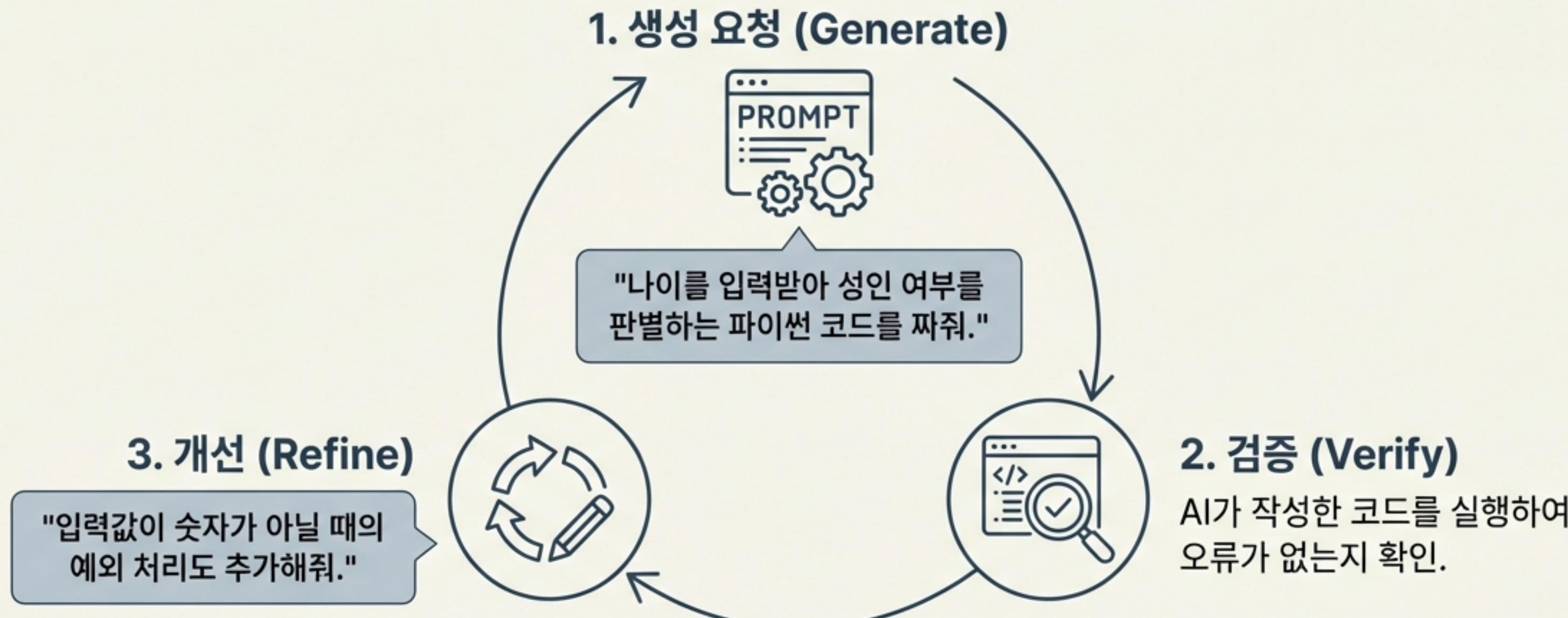


Scenario:

```
is_adult = age >= 20
```

```
can_enter = (age >= 18) and has_id
```

AI와 함께 코딩하기 (LLM Workflow)

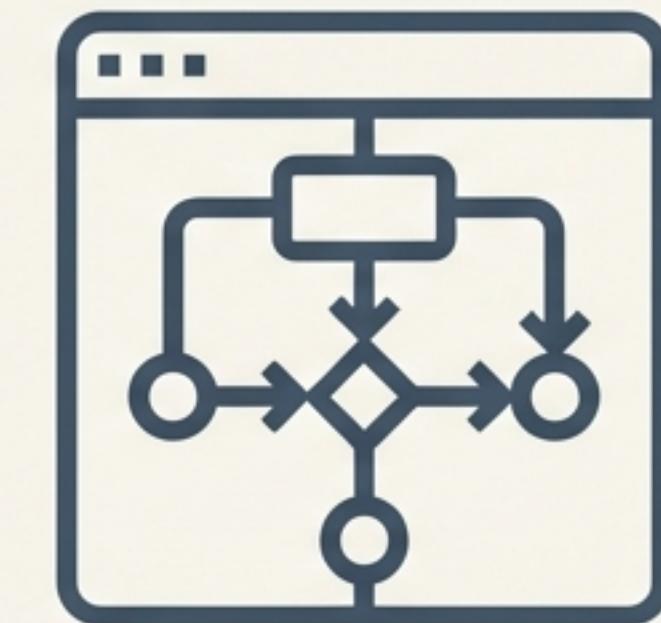
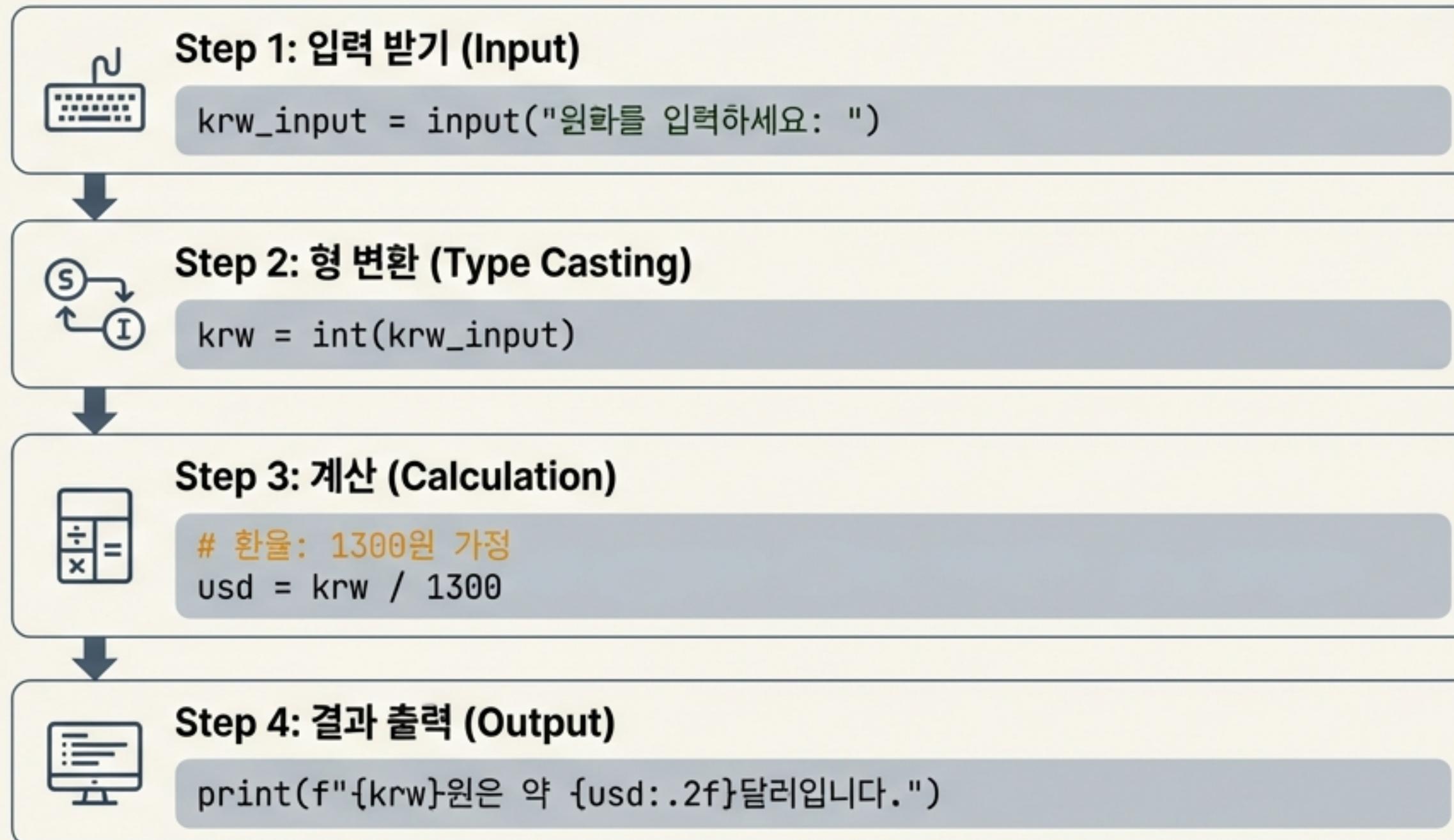


Core Point

프롬프트를 구체적으로 작성할수록 더 정확한 코드를 얻을 수 있습니다.

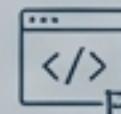
실습: 환전 계산기 만들기

Goal: 원화(KRW)를 입력받아 달러(USD)로 환전하는 프로그램.



Scenario:
입력: 65000
출력: 65000원은 약 50.00달러입니다.

Week 3 핵심 요약 (Cheat Sheet)



변수 (Variables)

이름은 **Snake Case** (user_name), 의미 있게 짓기.

```
user_name = "홍길동"
```



자료형 (Data Types)

type()으로 확인. 문자열("10")과 숫자(10)는 다름.

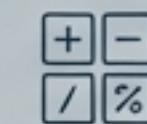
```
str_val = "10" # 문자열  
int_val = 10 # 숫자
```



입력 (Input)

input()은 문자열 → **int()** 형변환 필수.

```
age = int(input("나이: "))
```



연산 (Operators)

나눗셈 **/**은 실수 결과, **//**는 몫, **%**는 나머지.

```
print(7 / 2) # 3.5  
print(7 // 2) # 3  
print(7 % 2) # 1
```



출력 (Output)

f-string 사용이 가장 효율적.

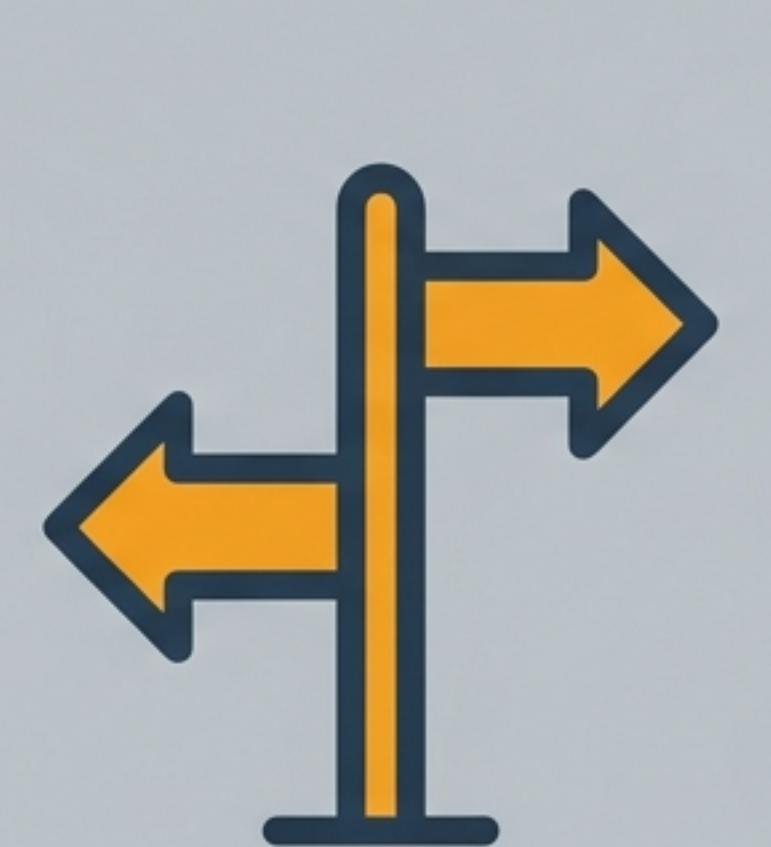
```
print(f"나이는 {age}세")
```

Next Step: 흐름 제어 (Control Flow)

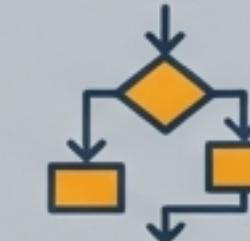
Current Limitation:



- 만약 0으로 나누면?(ZeroDivisionError)
- 입력값이 숫자가 아니면?



Next Lesson:



조건문(if)을 사용하여
프로그램의 흐름을 제어하고
예외를 처리하는 방법을
배웁니다.