

AI활용프로그래밍

Week 3. 파이썬 변수·자료형·연산자

기본 문법 다지기 With AI 연습문제

학습 목표

- 변수의 개념(이름 ↔ 값)을 설명할 수 있다.
- 기본 자료형(int/float/str/bool)을 구분하고 사용할 수 있다.
- 연산자(산술·비교·논리)를 사용해 간단한 계산 프로그램을 작성할 수 있다.
- LLM에게 “코드 생성→검증→개선”을 요청하는 프롬프트를 작성할 수 있다.

오늘의 구성

- 1) 변수와 자료형 기초
- 2) 연산자(산술/비교/논리)와 형변환
- 3) 입력(input)과 출력(print) 패턴
- 4) With AI 실습 3종 + 연습문제
- 5) 미니 퀴즈 & 과제 안내

변수(Variable)란?

값(Value)을 저장해두고 “이름”으로 다시 쓰게 해주는 것
변수 이름은 “라벨(label)”이고, 실제 값은 메모리에 저장됨
같은 값을 여러 번 쓸 때, 변수로 만들면 수정과 재사용이 쉬움
예: `tax_rate = 0.1` → 세율이 바뀌면 한 곳만 수정

변수 할당과 출력

```
name = "Kim"  
age = 20  
print(name)  
print(age)  
  
# 값 업데이트  
age = age + 1  
print("next:", age)
```

핵심 포인트

"="는 수학의 같다(=)가 아니라
“대입(할당)”

변수는 “바뀔 수 있는 값”

print()로 중간값을 찍으면 확인하면
디버깅이 쉬움

변수 이름 규칙 & 스타일

영문/숫자/_ (언더스코어) 사용, 숫자로 시작 불가
예약어(if, for, class 등)는 이름으로 사용 불가
의미 있는 이름 추천: a,b 보다 total_score
스타일: snake_case 사용 (예: total_score)

기본 자료형 한눈에 보기

int: 정수 (예: 10, -3)

float: 실수 (예: 3.14, 0.1)

str: 문자열 (예: "hello")

bool: 참/거짓 (True, False)

자료형 확인(type) 예시

```
x = 10  
y = 3.14  
s = "10"  
flag = True  
  
print(type(x))  
print(type(y))  
print(type(s))  
print(type(flag))
```

핵심 포인트

type()은 “현재 값의 자료형”을 알려줌

문자열 "10"은 숫자가 아님
→ 계산하려면 변환 필요

bool은 조건문에서 많이 사용(Week 4)

출력 포맷: f-string

```
name = "Kim"  
score = 92  
  
print(f"{name}님의 점수: {score}")  
print(f"합격 여부: {score >= 60}")
```

핵심 포인트

f"..."{변수}..." 형태

문자열 + 숫자 연결은 f-string이
보편적으로 사용됨

출력 예시를 먼저 정하면 코딩이
쉬워짐

형 변환(Casting) & 입력(input)

input()은 항상 문자열(str)로 입력을 받음
숫자로 계산하려면 int(), float()로 변환
변환 실패(ValueError) 가능 → 입력값 가정/검증이 중요
오늘은 “정상 입력”을 가정하고 연습

입력 2개 받아 계산하기

```
a = int(input("a: "))
b = int(input("b: "))

print("sum:", a + b)
print("diff:", a - b)
print("prod:", a * b)
print("quot:", a / b)
```

핵심 포인트

int(input()) 패턴은 자주 씀
나누셈(/) 결과는 float
b가 0이면 오류(ZeroDivisionError)
→ Week 4에서 처리

연산자 1: 산술 연산자

$+, -, *, /$: 기본 연산

$//$: 몫(정수 나눗셈), $\%$: 나머지

$**$: 거듭제곱

우선순위: $** \rightarrow *//,% \rightarrow +,-$ (필요하면 괄호)

연산자 예시(몫/나머지)

```
n = 17
print(n // 5)    # 3
print(n % 5)    # 2

# 1시간 75분 → 몇 시간 몇 분?
m = 75
print(m // 60, "hour")
print(m % 60, "min")
```

핵심 포인트

몫/나머지는 “단위 변환”에 자주 사용
예: 초 → 분/초, 돈 → 만원/천원
실습에서 자주 등장

연산자 2: 비교/논리

비교: ==, !=, >, >=, <, <=

논리: and, or, not

비교 결과는 bool(True/False)

조건문(if)의 “재료”가 됨 (다음 주!)

비교/논리 연산 예시

```
age = 19  
has_id = True  
  
is_adult = age >= 20  
can_enter = (age >= 18) and has_id  
  
print(is_adult)  
print(can_enter)
```

핵심 포인트

긴 조건은 괄호로 묶어 가독성↑
중간 변수를 두면 디버깅이 쉬움
조건식은 “문장”처럼 읽히게 작성

With AI: 오늘의 협업 방식

사람이 할 일: 요구사항 정리, 입력/출력 예시 만들기, 결과 검증

AI가 할 일: 코드 초안 생성, 설명/주석 추가, 리팩터링 아이디어

주의: AI 코드는 “실행해보고” 수정해야 내 코드가 됨

규칙: (1) 프롬프트에 예시 포함 (2) 테스트로 검증

프롬프트 템플릿(코드 생성)

역할: 너는 파이썬 튜터야.
목표: 아래 문제를 파이썬으로 풀어줘.
입력: (예) a=3, b=5
출력: (예) sum=8, avg=4.0
제약: 변수/자료형/연산자만 사용(조건문 X)
요청: 코드 + 한 줄씩 설명 + 테스트 3개

프롬프트 포인트

같은 요청도 구체적인 프롬프트,
예를들어

역할/목표/입력/출력/제약을 분리
“테스트 3개”
와 같이 요구하면 생성 결과물의
신뢰도 ↑
제약을 걸면 의도 범위 밖 기능을
방지

실습 1: 합계/평균 계산기

문제: 정수 2개를 입력받아 합계(sum)와 평균(avg)을 출력

조건: 평균은 실수로 출력(예: 4.0)

LLM 활용: “입력/출력 예시 + 제약조건” 포함해 코드 생성 요청

완료 기준: 테스트 3개를 직접 실행해 통과시키기

실습 2: 문자열 포매팅 영수증

문제: 상품명, 가격, 수량을 입력받아 영수증 형태로 출력

필수: f-string 사용

예: 사과 1000원 × 3개 → 총액 3000원

LLM 활용: “출력 모양”을 먼저 보여주고 그대로 나오게 요청

실습 3: 단위 변환기(리팩터링)

문제: cm를 입력받아 inch로 변환($1 \text{ inch} = 2.54 \text{ cm}$)

추가: 섭씨→화씨 변환도 선택 문제로 추가해보기

LLM 활용: 1) 코드 초안 2) 변수명 개선 3) 주석/함수화 아이디어

완료 기준: 입력/출력 안내 문구까지 자연스럽게

AI 답변 검증 체크리스트

- 입력값이 문자열인지/숫자인지 확인했나?
- 출력 형식이 예시와 동일한가? (띄어쓰기, 단위)
- 테스트 3개 이상 실행했나?
- 엣지 케이스 1개 이상 생각했나? (0, 음수, 큰 수)
- 코드가 “내 말”로 설명 가능한가?

미니 퀴즈(5문항)

- 1) `input()`의 반환 자료형은?
- 2) `17 // 5` 의 결과는?
- 3) `"10" + "3"` 의 결과는?
- 4) `3 < 5 and 2 > 10` 의 결과는?
- 5) 변수 이름으로 사용할 수 없는 예: ①`my_var` ②`2nd` ③`score`

과제/연습문제

기본: 단위 변환기 2종(길이, 온도) 완성 + 테스트 5개

도전: “거스름돈 계산기” 만들기(몫/나머지 활용)

제출: .py 파일 1개 + 실행 결과 캡처 1장

AI 사용 시: 사용한 프롬프트 2개를 주석으로 첨부

요약 & 다음 주 예고

오늘: 변수/자료형/연산자 + 입력/출력 패턴

핵심: “문제 → 변수 정의 → 계산 → 출력” 순서로 생각

다음 주(Week 4): 조건문/반복문으로 프로그램 흐름 제어

예고 실습: 성적 등급, 구구단, 숫자 맞히기 게임