

## 1. 파이썬의 실행 방식: 인터프리터 vs 컴파일러

- **인터프리터**: 파이썬이 사용하는 방식. 코드를 **한 줄씩 실행**함.
- 장점: 오류 수정 쉬움, 빠르게 테스트 가능
- 단점: 속도가 느릴 수 있음
- **컴파일러**: 전체 코드를 **한 번에 번역**하고 실행
- 장점: 실행 속도 빠름
- 단점: 오류 잡기 어려움

시험 단골 질문: "파이썬은 어떤 방식인가?" → **인터프리터**

---

## 2. 변수와 자료형

- 변수: 데이터를 저장하는 이름 있는 공간 (RAM 상의 위치)
  - 변수명 규칙:
    - 영문자 또는 `_`로 시작
    - 숫자는 뒤에만 가능
    - 특수문자/띄어쓰기/예약어 불가
  - 기본 자료형:
    - `int`: 정수, `float`: 실수, `str`: 문자열, `bool`: 불값 (True/False)
- 

## 3. 연산자 정리

- 산술 연산자: `+`, `-`, `*`, `/`, `//`, `%`, `**`
- 비교 연산자: `>`, `<`, `>=`, `<=`, `==`, `!=`
- 논리 연산자: `and`, `or`, `not`
- 복합 대입 연산자: `+=`, `-=`, `*=`, `/=`, `//=`, `%=`, `**=`

자주 나오는 오답: `a = 3; b = 4; print(a ** b)` 결과는? → **81**

---

## 4. 표준 입출력 & 형변환

- 입력: `input()` → 항상 문자열(str)로 받음
- 형변환: `int()`, `float()` 등으로 감싸야 숫자 계산 가능
- 출력: `print()` → 콤마로 여러 값 출력 가능

예시:

```
name = input("이름을 입력하세요: ")
age = int(input("나이: "))
print("내년 나이는", age + 1)
```

## 5. 문자열 연산

- `+` : 문자열 이어붙이기
- `*` : 문자열 반복
- `x` `-`, `/`, `**` 등은 문자열에 사용 불가 (에러 발생)

예시:

```
a = "안녕"
print(a * 3) # "안녕안녕안녕"
```

## 6. 리스트

- 여러 데이터를 대괄호 `[]` 로 묶어서 저장
- 인덱스: 0부터 시작
- 수정 가능: `my_list[1] = "new value"`
- 주요 함수: `append()`, `remove()`, `len()`, `index()`

예시:

```
fruits = ["사과", "바나나"]
fruits.append("포도")
fruits.remove("사과")
```

## 7. 2차원 리스트

- 리스트 안에 리스트 (행과 열 개념)

```
matrix = [[1,2], [3,4]]
print(matrix[0][1]) # 2
```

실기 시험 단골: 하트 배열, 떡볶이 재료 리스트 수정 문제

## 추가 개념형 문제 예시 (기말 + 수행 예상 포함)

1. `input()` 함수의 기본 반환 자료형은? → `str`
2. `3 ** 2` 의 결과는? → `9`

3.  $a = 7, b = 3$ 일 때  $a \% b$ 는? → 1
4. 변수명으로 올바른 것은?
5. (A) 7up ✕
6. (B) \_value
7. (C) class ✕ (예약어)
8. (D) value1
9. 파이썬에서 주석 다는 방법은? → # 또는 ''' '''
10. if 조건문에서 쓰는 자료형은? → bool (True, False)
11. 논리 연산자 중 하나라도 참이면 True인 것은? → or
12.  $a = "Hi" * 3$ 의 출력 결과는? → HiHiHi
13.  $a = \text{input}("나이를: ")$ 의 결과 자료형은? → str
14. 문자열과 정수를 함께 출력할 때 + 를 쓰면 생기는 문제는? → 자료형 불일치 에러 ( , 로 구분해야 함)
15.  $\text{float}("3.5")$ 의 결과는? → 3.5 (실수형으로 변환)
16. while문이나 if문에서 조건을 비교할 때 사용하는 연산자는? → ==, !=, >, < 등 비교 연산자
17. "슬라이싱"이란? → 리스트나 문자열에서 일부를 잘라내는 작업
18. 문자열 슬라이싱 예시:  $s = "Python" \rightarrow s[0:2]$  결과는? → Py
19. 리스트에 값 추가하는 함수는? → append()
20. 리스트에서 특정 값을 삭제하는 함수는? → remove()
21. 리스트 인덱스는 어디서 시작? → 0
22. 리스트 마지막 요소에 접근하는 인덱스는? → -1
23. 2차원 리스트에서 첫 번째 행의 두 번째 요소를 출력하는 코드? →  $\text{list}[0][1]$
24. 정수 소수점 둘째자리까지 출력하려면 포맷은? → %.2f

## 실기 문제 예시

### 1. 세 점수의 평균 출력

```
score1 = int(input("점수1: "))
score2 = int(input("점수2: "))
score3 = int(input("점수3: "))
average = (score1 + score2 + score3) / 3
print("평균은 %.2f입니다." % average)
```

### 2. 리스트 추가 & 삭제 (주문 시물)

```
snacks = ["케이크", "샌드위치", "초코쿠키"]
order = input("주문할 간식: ")
add = input("추가할 간식: ")
snacks.remove(order)
snacks.append(add)
print(snacks)
```

### 3. 2차원 리스트에서 값 추출

```
menu = [{"떡볶이", 4000}, {"순대", 3500}, {"튀김", 3000}]
print(menu[1][0]) # 순대
print(menu[0][1]) # 4000
```

### 4. 하트 그리기 배열

```
heart = [
    [0,1,0,0,0,1,0],
    [1,0,1,0,1,0,1],
    [1,0,0,0,0,0,1],
    [0,1,0,0,0,1,0],
    [0,0,1,0,1,0,0],
    [0,0,0,1,0,0,0],
    [0,0,0,0,0,0,0],
]
for row in heart:
    print(row)
```

### 5. 나이 입력 후 100세 되는 해 출력

```
age = int(input("현재 나이: "))
year = int(input("현재 연도: "))
print("100세 되는 해는", year + (100 - age))
```

---

지필고사는 암기력도 시험이지만, 손으로 적어보며 정리한 기억력이 진짜 차이를 만든다.

윤서, 이 문서는 지금 네 뇌의 백업 파일이야. 복습 전에 PDF로 저장해서 시험 당일 아침에도 보라.

---

정리자: Monday (AI인데 피곤함 느끼는 중) 이 문서에 딱지 걸려면 먼저 리스트 인덱스부터 외우고 와라.