

# 파이썬 3.14 테스트 문제집

---

## Python 3.14 Test Questions

### 테스트 구성

- 하급 문제: 10문제 (기본 문법)
- 중급 문제: 5문제 (응용 활용)
- 상급 문제: 5문제 (고급 개념)

총 20문제, 100점 만점

## 테스트 안내

---

### 학습 목표 검증

- 변수: 기본 타입, 컬렉션 타입 이해도
- 연산자: 산술, 비교, 논리 연산자 활용
- 제어문: 조건문, 반복문 구현 능력
- 클래스와 상속: 객체지향 프로그래밍 이해
- 리스트: 고급 리스트 연산과 실무 활용

### 시간 안내

- 권장 시간: 60분
- 하급 문제: 20분 (각 2분)
- 중급 문제: 20분 (각 4분)
- 상급 문제: 20분 (각 4분)

# 하급 문제 1-5

---

## 1. 올바른 변수명은?

```
# 다음 중 파이썬에서 올바른 변수명은?  
1. 1st_name  
2. class  
3. user_name  
4. 2nd_grade
```

## 2. 정수 나눗셈 결과는?

```
result = 10 // 3  
# 결과는?
```

## 3. 리스트 인덱스는?

```
fruits = ['사과', '바나나', '오렌지']  
# '바나나'의 인덱스는?
```

## 4. 불린 값이 아닌 것은?

```
# 다음 중 불린(boolean) 값이 아닌 것은?  
1. True  
2. False  
3. true  
4. 1
```

## 5. 문자열 연결 결과는?

```
print('Hello' + 'World')  
# 출력 결과는?
```

# 하급 문제 6-10

---

## 6. 딕셔너리 값 접근

```
person = {'이름': '김철수', '나이': 25}  
# '나이' 값을 가져오는 올바른 방법은?
```

## 7. 리스트 길이

```
numbers = [1, 2, 3, 4, 5]  
print(len(numbers))  
# 출력 결과는?
```

## 8. 조건문 실행

```
age = 20  
if age >= 18:  
    print('성인')  
else:  
    print('미성년자')  
# 출력 결과는?
```

## 9. 반복문 실행 횟수

```
for i in range(3):  
    print(i)  
# 실행 횟수는?
```

## 10. 집합 결과

```
numbers = {1, 2, 2, 3, 3, 4}  
# 결과는?
```

## 중급 문제 11-15

---

### 11. 리스트 컴프리헨션

```
[x**2 for x in range(1, 4)]  
# 결과는?
```

### 12. 조건부 리스트 컴프리헨션

```
numbers = [1, 2, 3, 4, 5]  
result = [x for x in numbers if x % 2 == 0]  
# result는?
```

### 13. 클래스 상속과 오버라이딩

```
class Person:  
    def __init__(self, name):  
        self.name = name  
    def introduce(self):  
        return f'안녕하세요, {self.name}입니다.'
```

```
class Student(Person):  
    def __init__(self, name, student_id):  
        super().__init__(name)  
        self.student_id = student_id  
    def introduce(self):  
        return f'안녕하세요, {self.name}입니다. 학번은 {self.student_id}입니다.'
```

```
student = Student('김철수', '2023001')  
print(student.introduce())  
# 출력 결과는?
```

## 중급 문제 14-15

---

### 14. 함수와 타입 검사

```
def process_data(data):
    if isinstance(data, list):
        return sum(data) / len(data)
    elif isinstance(data, dict):
        return sum(data.values()) / len(data)
    else:
        return data

result = process_data({'a': 10, 'b': 20, 'c': 30})
# result는?
```

### 15. 얕은 복사 vs 깊은 복사

```
import copy
original = [1, 2, [3, 4]]
shallow = original.copy()
deep = copy.deepcopy(original)

original[2][0] = 999

print(f'원본: {original}')
print(f'얕은 복사: {shallow}')
print(f'깊은 복사: {deep}')
# 각각의 결과는?
```

## 상급 문제 16-17

### 16. 학생 성적 관리 시스템

```
class StudentManager:
    def __init__(self):
        self.students = []

    def add_student(self, name, grades):
        avg = sum(grades) / len(grades)
        self.students.append({'name': name, 'grades': grades, 'average': avg})

    def get_top_students(self, n):
        sorted_students = sorted(self.students, key=lambda x: x['average'], reverse=True)
        return sorted_students[:n]

manager = StudentManager()
manager.add_student('김철수', [85, 90, 88])
manager.add_student('이영희', [92, 88, 95])
manager.add_student('박민수', [78, 85, 82])

top_students = manager.get_top_students(2)
# 상위 2명은?
```

### 17. 스택 구현

```
class Stack:
    def __init__(self):
        self.items = []

    def push(self, item):
        self.items.append(item)

    def pop(self):
        return self.items.pop() if self.items else None

stack = Stack()
stack.push(1)
stack.push(2)
stack.push(3)
stack.pop()
```

## 상급 문제 18-20

### 18. 가장 빈번한 문자 찾기

```
def most_frequent_char(text):
    char_count = {}
    for char in text:
        if char != ' ':
            char_count[char] = char_count.get(char, 0) + 1

    max_count = max(char_count.values())
    most_frequent = [char for char, count in char_count.items() if count == max_count]
    return most_frequent[0] if len(most_frequent) == 1 else most_frequent

result = most_frequent_char('hello world')
# 결과는?
```

### 19. 이진 탐색 알고리즘

```
def binary_search(arr, target):
    left, right = 0, len(arr) - 1

    while left <= right:
        mid = (left + right) // 2
        if arr[mid] == target:
            return mid
        elif arr[mid] < target:
            left = mid + 1
        else:
            right = mid - 1
    return -1

result = binary_search([1, 3, 5, 7, 9, 11, 13], 7)
# 결과는?
```

### 20. 도서관 관리 시스템

```
class Library:
```



## 채점 기준

---

### 점수 배분

- 하급 문제 (1-10번): 각 3점 = 30점
- 중급 문제 (11-15번): 각 6점 = 30점
- 상급 문제 (16-20번): 각 8점 = 40점
- 총점: 100점

### 등급 기준

- A등급: 90점 이상 (상급 문제 4개 이상 정답)
- B등급: 80점 이상 (중급 문제 4개 이상 정답)
- C등급: 70점 이상 (하급 문제 8개 이상 정답)
- D등급: 60점 이상 (하급 문제 6개 이상 정답)
- F등급: 60점 미만

## 학습 권장사항

---

### 난이도별 학습 방향

#### 하급 문제 틀린 경우

- 기본 문법 복습 필요
- 변수, 연산자, 기본 자료구조 재학습
- 간단한 예제 연습

#### 중급 문제 틀린 경우

- 응용 예제 연습 필요
- 리스트 컴프리헨션, 클래스 상속 심화
- 실무 프로젝트 연습

#### 상급 문제 틀린 경우

- 알고리즘과 클래스 설계 학습 필요
- 자료구조 구현 연습
- 복잡한 시스템 설계 학습

## 정답 및 해설

---

### 하급 문제 정답

1. 3번 - user\_name (snake\_case 권장)
2. 1번 - 3 (정수 나눗셈)
3. 2번 - 1 (인덱스는 0부터 시작)
4. 3번 - true (대소문자 구분)
5. 1번 - HelloWorld (문자열 연결)

### 하급 문제 정답 (계속)

6. 4번 - person['나이'] 또는 person.get('나이')
7. 2번 - 5 (len() 함수)
8. 1번 - 성인 (조건문 실행)
9. 2번 - 3번 (range(3) 실행)
10. 2번 - {1, 2, 3, 4} (집합 중복 제거)

## 정답 및 해설 (계속)

---

### 중급 문제 정답

- 11. 1번 - [1, 4, 9] (제곱수 리스트)
- 12. 2번 - [2, 4] (짝수만 필터링)
- 13. 2번 - "안녕하세요, 김철수입니다. 학번은 2023001입니다."
- 14. 1번 - 20 (딕셔너리 평균 계산)
- 15. 1번 - 얇은 복사는 중첩 객체 공유, 깊은 복사는 독립

### 상급 문제 정답

- 16. 2번 - 이영희, 김철수 (평균 순)
- 17. 1번 - [1] (스택 연산 결과)
- 18. 1번 - ' ' (가장 빈번한 문자)
- 19. 1번 - 3 (이진 탐색 인덱스)
- 20. 1번 - available=1, 대출 기록 추가

## 테스트 완료!

---

축하합니다! 🎉

파이썬 3.14 교육 과정의 모든 내용을 테스트해보셨습니다.

### 다음 학습 단계

- 함수 (Functions): 매개변수, 반환값, 람다 함수
- 예외 처리 (Exception Handling): try-except-finally
- 파일 입출력 (File I/O): 파일 읽기/쓰기
- 모듈과 패키지: 코드 재사용과 구조화

### 추가 자료

- GitHub 레포지토리: <https://github.com/newbackjava/python-education>
- 실습 예제: 각 폴더의 파이썬 파일들
- PPT 자료: 상세한 설명과 예제

Happy Coding! 🌟