

>>>>>>>>>> 제어 구조의 설계 원리를 중심으로 배우는 >>>>>>>>>>

# 프로그래밍의 정석

# 과이션

도경구 지음



CHAPTER 5

재귀와 반복 : 정렬

- 연습

## 5.1 팰린드롬 검사 함수

팰린드롬(Palindrome, 회문, 回文)은 거꾸로 읽어도 제대로 읽는 것과 같은 낱말이다. 문자열을 인수로 받아서 팰린드롬인지 확인하는 함수는 다음과 같이 작성하면 잘 작동한다.

code : 5-33.py

```

1 def ispalindrome(s):
2     if len(s) <= 1:
3         return True
4     elif s[0] != s[-1]:
5         return False
6     else:
7         return ispalindrome(s[1:-1])

```

그런데 이 함수는 선택문을 사용하는 대신에 논리식 하나만으로 작성할 수도 있다. 다음 뼈대코드에 맞추어 논리식 하나만으로 동일하게 작동하도록 함수를 재작성하자.

code : 5-34.py

```

1 def ispalindrome(s):
2     return None # Write your Boolean expression here.

```

s	ispalindrome(s)
""	True
"a"	True
"aa"	True
"aba"	True
"abba"	True
"aaba"	False
"abcba"	True
"여보 안경 안 보여"	False
"여보안경안보여"	True

## 5.2 삼각수 (for 루프 버전)

4장 <연습 문제 4.3>에서 작성한 `trinum` 함수를 for 루프로 재작성하자.

## 5.3 덧셈만 가지고 제곱 계산하기 (for 루프 버전)

4장 <연습 문제 4.4>에서 작성한 `square` 함수를 for 루프로 재작성하자.

## 5.10 재귀 함수를 꼬리재귀, while 루프, for 루프 함수로 변환하기

다음 함수를 똑같이 작동하는 (1) 꼬리재귀 함수, (2) while 루프 함수, (3) for 루프 함수로 차례로 변환하자. 선호하는 경우 리스트 append 메소드를 사용해도 좋다.

code : 5-38.py

```
1 def updown(ns):
2     if ns != []:
3         if ns[0] % 2 == 0:
4             return [ns[0]//2] + updown(ns[1:])
5         else:
6             return [ns[0]*2] + updown(ns[1:])
7     else:
8         return []
9
10 # Test code
11 print(updown([4, 6, 5, 3, 7, 6, 2, 1, 3, 2]))
12 #           [2, 3, 10, 6, 14, 3, 1, 2, 6, 1]
13 print(updown([14, 69, 87, 13, 0, 16, 83, 19, 45, 88]))
14 #           [7, 138, 174, 26, 0, 8, 166, 38, 90, 44]
```

>>>>>>>>>> 제어 구조의 설계 원리를 중심으로 배우는 >>>>>>>>>>

# 프로그래밍의 정석

도경구 지음

# 과이썬



CHAPTER 5

재귀와 반복 : 정렬

- 연습