

```
# 삽입 정렬
def insertion_sort(my_list):
    # 코드를 입력하세요.

    some_list = [11, 3, 6, 4, 12, 1, 2]
    insertion_sort(some_list)
    print(some_list)
```

'선택 정렬'에서는 0 번 인덱스에 들어갈 값을 찾고, 1 번 인덱스에 들어갈 값을 찾고, 2 번 인덱스에 들어갈 값을 찾고... 이런 식으로 각 자리에 들어갈 값을 찾아서 정렬을 했습니다.

'삽입 정렬'에서는 0 번 인덱스에 있는 값이 들어갈 자리를 찾고, 1 번 인덱스에 있는 값이 들어갈 자리를 찾고, 2 번 인덱스에 있는 값이 들어갈 자리를 찾고... 이런 식으로 각 값이 들어갈 자리를 찾아서 정렬을 해야 합니다.

현재 자리를 찾아야 하는 값을 **key** 라고 부릅니다. 일단 반복문을 쓰면:

```
for i in range(len(my_list)):
    key = my_list[i]
```

매번 반복문의 수행 부분에 들어갈 때 이 두 가지를 생각하고 있어야 합니다:

1. **my_list[:i]** (0 번 인덱스부터 **i - 1** 번 인덱스까지)는 정렬되어 있다.
2. **key** 를 **my_list[:i]** 에서 들어갈 수 있는 자리에 삽입해야 한다.

자리 찾기

my_list[:i] 는 이미 정렬되어 있기 때문에 **my_list[:i]** 에서 **i - 1** 번 인덱스의 값이 가장 큰 값입니다. **i - 1** 번 인덱스에 있는 값이 **key** 보다 작거나 같으면 **key** 는 이미 올바른 자리에 있기 때문에 그냥 두면 됩니다.

만약 **i - 1** 번 인덱스에 있는 값이 **key** 보다 크면 **key** 보다 오른쪽으로 가야하기 때문에 **key** 가 있던 자리에 넣어줍니다. 그리고 이제 똑같이 **key** 를 **i - 2** 번에 있는 값과 비교해서 **key** 가 더 크면 그냥 두고, 아니면 **i - 2** 번 값을 **key** 의 자리에 넣어줍니다.

이런식으로 **key** 가 큰 경우가 생기면 바로 끝낼 수 있고, 아니면 0 번에 있는 값을 확인할 때까지 반복해야 합니다.

```

# i - 1부터 시작해서 왼쪽으로 하나씩 확인
# 왼쪽 끝까지(0번 인덱스) 다 봤거나
# key가 들어갈 자리를 찾으면 끝냄
j = i - 1
while j >= 0 and my_list[j] > key:
    my_list[j + 1] = my_list[j]
    j = j - 1

# key가 들어갈 자리에 삽입
# 왼쪽 끝까지 가서 j가 -1이면 0번 인덱스에 key를 삽입
my_list[j + 1] = key

```

모범 답안

테스트 해봅시다!

```

# 삽입 정렬
def insertion_sort(my_list):
    for i in range(len(my_list)):
        key = my_list[i]

        # i - 1부터 시작해서 왼쪽으로 하나씩 확인
        # 왼쪽 끝까지(0번 인덱스) 다 봤거나
        # key가 들어갈 자리를 찾으면 끝냄
        j = i - 1
        while j >= 0 and my_list[j] > key:
            my_list[j + 1] = my_list[j]
            j = j - 1

        # key가 들어갈 자리에 삽입
        # 왼쪽 끝까지 가서 j가 -1이면 0번 인덱스에 key를 삽입
        my_list[j + 1] = key

some_list = [11, 3, 6, 4, 12, 1, 2]
insertion_sort(some_list)
print(some_list)

```

[1, 2, 3, 4, 6, 11, 12]



수업을 완료하셨으면 체크해주세요.



수강생 Q&A 보기

(/questions?

assignment_id=211&sort_by=popular)

(/questions/new?

assignment_id=211&op1=%ED%94%84%EB%A1%9C%EA%B7%

< 이전 강의
삽입 정렬 코드 짜기 (/assignments/199)

다음 강의

합병 정렬 (Merge Sort)

> (/assignments/200)