

35강. 함수를 사용하는 이유

혼자 공부하는 파이썬 - 윤인성 -

<https://www.youtube.com/playlist?list=PLBXuLglnP-5kr0PclHz1ubNZgESmliuB7>



TABLE OF CONTENTS



1. 재귀함수 복습문제 (리스트 평탄화)
2. 코드에 이름 붙이기
3. 코드 유지 보수

1. 재귀함수 복습문제(리스트 평탄화)

1. 다음 빈칸을 재귀 함수로 만들어서 리스트를 평탄화하는 함수를 만들어 보세요. **리스트 평탄화** flatten는 중첩된 리스트가 있을 때 중첩을 모두 제거하고 풀어서 1차원 리스트로 만드는 것을 의미합니다. 다음 실행결과를 참고해 주세요.

```
def flatten(data):
```

```
example = [[1, 2, 3], [4, [5, 6]], 7, [8, 9]]
```

```
print("원본:", example)
```

```
print("변환:", flatten(example))
```

실행결과



원본: `[1, 2, 3], [4, [5, 6]], 7, [8, 9]`

변환: `[1, 2, 3, 4, 5, 6, 7, 8, 9]`

이 문제를 풀 때는 리스트의 데이터가 리스트인지 아닌지 구분할 수 있어야 합니다. `type()` 함수를 사용해서 자료형을 판별할 때는 다음과 같은 코드를 사용합니다.

```
>>> type(10) == int
```

```
True
```

```
>>> type("10") == str
```

```
True
```

```
>>> type([]) == list
```

```
True
```

```
def flatten(data):  
    output = []  
    for item in data:  
        if type(item) == list:  
            output += flatten(item)  
        else:  
            output += [item]  
    return output  
  
example = [[1,2,3],[4,[5,6]],7,[8,9]]  
print(flatten(example))
```

함수의 기본구조부터 ...

1. 초기값(output) 만들어주고 return
 output = []
 return output
2. 반복문 돌리고 나온 값들을 output에
 넣어주기

플래팅은 재귀함수 이외에도 여러가지 방법이
있으니 구글링해보기

<좀 더 알아보기> 코드에 이름 붙이기

가독성이란 코드를 쉽게 읽을 수 있는 성질입니다.

<가독성을 높이는 방법>

1. 주석 사용

2. 함수 사용

즉, 가독성을 높이기 위해 함수를 사용합니다.

왜 함수를 사용하면 가독성이 높아지는지 다음 장부터 살펴보겠습니다.

```
number_input_a = input("숫자 입력>")
radius = float(number_input_a)
print(2 * 3.14 * radius)
print(3.14 * radius * radius)
```

왼쪽을 보면 뭐하는 코드인지 한 눈에 들어오지 않습니다.

원 둘레와 넓이를 구하는 코드인데 이해하는데 시간이 조금 걸립니다.

하지만 함수를 활용하면 바로 이해할 수 있습니다.

```
# 함수 정의 |
def number_input():
    output = input("숫자 입력>")
    return float(output)
def get_circumference(radius):
    return 2 * 3.14 * radius
def get_circle_area(radius):
    return 3.14 * radius * radius
# 코드 본문
radius = number_input()
print(get_circumference(radius))
print(get_circle_area(radius))
```

함수를 만들면 코드에 이름을 붙일 수 있습니다.

코드에 이름을 붙이면 가독성이 증가합니다.

왼쪽에 **#코드 본문** 쪽을 보면 주석이 없어도 뭐하는 코드인지 쉽게 알 수 있습니다.

(area와 circumference라는 단어를 모르면 힘들겠지만요.)

한줄의 코드라도 의미를 가지고 있다면 함수로 만드는 것이 좋습니다.

이전 슬라이드에 함수 부분은 곧 배울 **모듈(module)**이라는 기능으로 더 간단하게 만들 수 있습니다.

<좀 더 알아보기> 코드 유지보수

```
def get_circumference(radius):  
    return 2 * 3.14 * radius
```

여기서 3.14를 변수로 만들어 가독성을 높여 줄 수 있습니다.

```
PI = 3.14
```

```
def get_circumference(radius):  
    return 2 * PI * radius
```

이렇게 변수로 만들어 저장하면 누군가 “파이의 정밀도를 올리고 싶어요 3.141592로 변경해주세요”라고 요청하면 변수값만 수정하면 됩니다.

기존형태로 하면 2017.03.14라는 날짜가 코드에 포함되어 있다고 했을 때 3.14를 파이값으로 알고 수정하면 문제가 발생합니다.

하지만 변수로 선언하면 이런 실수가 발생할 경우가 없습니다.

함수도 마찬가지입니다.

```
print("<p>{}</p>".format("안녕하세요"))  
print("<p>{}</p>".format("간단한 HTML 태그를 만드는 예입니다."))
```

```
<p>안녕하세요</p>  
<p>간단한 HTML 태그를 만드는 예입니다.</p>
```

갑자기 '단순한 p태그로 감싸지 말고 <p class='content-line'></p>로 감싸주세요'라는 요청을 받는다고 가정해봅시다.

함수를 사용하지 않으면 일일이 다 바꿔줘야 하지만 함수를 사용하면 함수만 변경해주면 됩니다.

```
def p(content):  
    return "<p class='content-line'>{}</p>".format(content)
```

```
print(p("안녕하세요"))  
print(p("간단한 HTML 태그를 만드는 예입니다."))
```

```
<p class='content-line'>안녕하세요</p>  
<p class='content-line'>간단한 HTML 태그를 만드는 예입니다.</p>
```