

# 17장

## 루프 주무르기

# 17장 루프 주무르기

---

17.1 원하는 대로 루프 돌기

17.2 문자열에 대해 루프 돌기

17.3 요약

## 17.1 원하는 대로 루프 돌기

---



# 17.1 원하는 대로 루프 돌기

» `range(시작, 끝, 증가값)` 형태로 명령을 사용할 수 있으며, 괄호 안에 최소 1개~최대 3개의 수를 넣을 수 있다.

- 첫 번째 수(시작)는 첫 번째 값이다.
- 두 번째 수(끝)는 마지막을 지정한다. 다만 이 값 자체는 범위에 들어가지 않음에 유의하라.
- 수학 용어를 써서 말하면 범위 안에 들어가는 값은 시작 값 이상, 끝 값 미만의 값이다.
- 세 번째 수는 증가값이다. 첫 번째 값에 이 증가값을 차례로 더한 값들이 `range(시작, 끝, 증가값)`의 결과에 들어간다.

» 다음 규칙을 기억하자.

- `range(10)`처럼 괄호 안에 수를 하나만 넣으면 끝만 지정한다. 이때 시작은 0, 증가값은 1이다.
- `range(10, 20)`처럼 괄호 안에 수를 둘만 넣으면 순서대로 시작과 끝을 지정한다. 이때 증가값은 1이다.
- `range(10, 20, 2)`처럼 괄호 안에 수를 셋 넣으면 순서대로 시작, 끝, 증가값을 지정한다.



# 17.1 원하는 대로 루프 돌기

» 다음은 `range`의 괄호에 여러 가지 값을 넣고 결과가 어떻게 계산되는지 보여주는 예다.

- `range(5)`는 `range(0,5)`와 같고, 이는 다시 `range(0,5,1)`과 같다. 따라서 `range(5)`에 0, 1, 2, 3, 4가 들어 있다. 5가 들어 있지 않다는 사실에 유의하자.
- 비슷한 방식으로 생각하면 `range(2,6)`은 `range(2,6,1)`이다. 따라서 2, 3, 4, 5가 들어 있다.
- `range(6,2)`는 시작 값이 끝 값보다 크므로 아무 값도 들어 있지 않다.
- `range(2,8,2)`에는 2, 4, 6이 들어 있다.
- `range(2,9,2)`에는 2, 4, 6, 8이 들어 있다.
- `range(6,2,-1)`에는 6, 5, 4, 3이 들어 있다. 증가값을 음수로 할 수도 있다!

» 셀프 체크 17.1

## 17.2 문자열에 대해 루프 돌기

---



## 17.2 문자열에 대해 루프 돌기

```
for ch in "Python is fun so far! 재미있는 파이썬!": ---- ch가 루프 변수임
    print("문자:", ch) ---- 루프 변수 값을 출력
```

코드 17-1

문자열의 모든 문자에 대한 for 루프

» 코드 17-1에서는 루프 변수 이름으로 ch를 사용했지만, 원하는 다른 어떤 이름이든 붙일 수 있다. 한글이나 여러 기호, 공백도 모두 한글자로 취급하므로 여기서 루프에 사용한 문자열 길이는 31이다. 따라서 코드 17-1의 for 루프는 31번 반복된다.

- 매번 ch에는 "Python is fun so far! 재미있는 파이썬!"이라는 문자열의 문자들이 한 번에 한 글자씩 들어간다.
- 루프 코드 블록에서는 루프 변수의 값을 파이썬 콘솔에 출력한다

### » 셀프 체크 17.2



## 17.2 문자열에 대해 루프 돌기

» 17-1을 다시 작성하면 코드 17-2와 같다. 이 경우 반드시 문자열을 변수에 저장해  
둬야 루프 안에서 문자열에 접근할 수 있다. 이 루프도 여전히 본문을 31번 반복한다.

- 여기서는 루프 변수  $i$ 가 0, 1, ..., 30까지 바뀌면서 문자열에 들어 있는 문자의 인덱스를 가리키게 된다.
- 코드 블록 안에서는 문자열에 인덱스를 사용해서 그 인덱스에 있는 문자를 얻어내야 한다.
- 이 코드는 더 복잡하고 코드 17-1처럼 직관적이지도 않다.

```
my_string = "Python is fun so far! 재미있는 파이썬!"
len_s = len(my_string)
for i in range(len_s) : ---- 0부터 len_s - 1까지 반복
    print("문자:", my_string[i]) ---- 문자열에 대해 인덱스 사용
```

문자열과 문자열 길이를 변수에 저장

코드 17-2

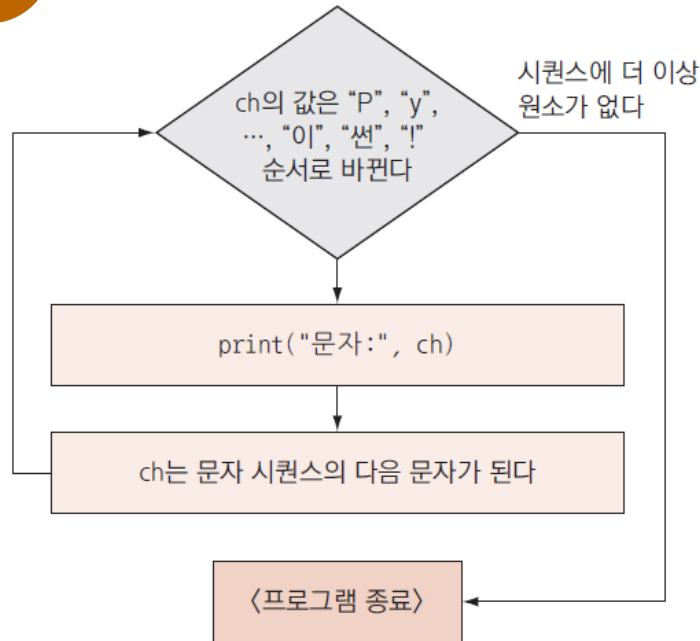
인덱스를 사용해 문자열을 처리하는 for 루프



# 17.2 문자열에 대해 루프 돌기

코드  
17-1

A. 문자열에 대해 직접 반복을 수행하는 경우



코드  
17-2

B. 문자열 인덱스에 대해 반복을 수행하는 경우

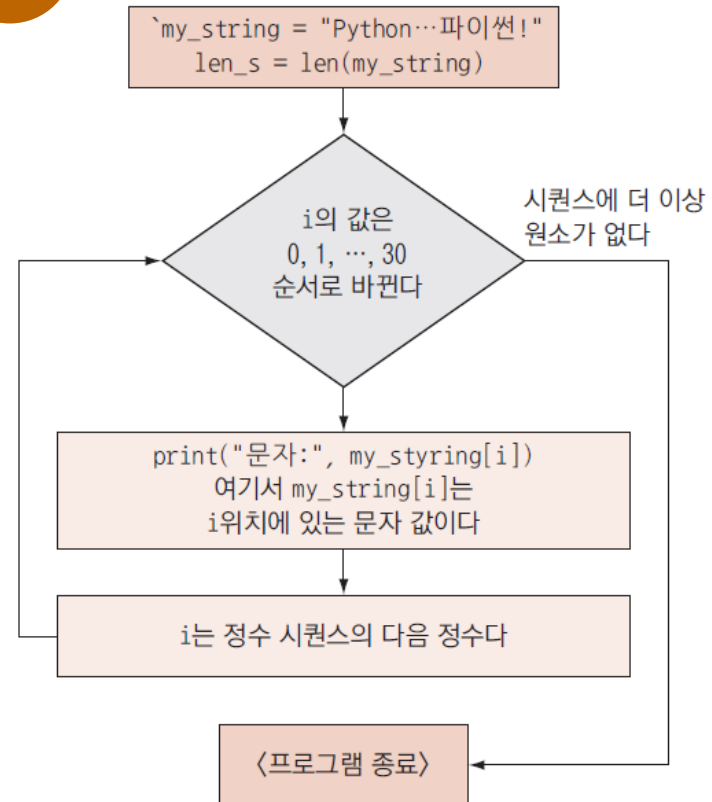


그림 17-1

(A)는 코드 17-1에 대한 흐름도이고, (B)는 코드 17-2에 대한 흐름도이다. (A)에서 루프 변수 ch는 문자열에 있는 각 문자를 가리킨다. (B)에서 루프 변수는 0부터 문자열 길이-1까지 문자들의 인덱스를 표시하는 정수를 담는다. (B)에는 루프 안에서 문자열에 루프 변수를 인덱스로 사용해 원하는 문자를 가져오는 단계가 더 들어가 있다

## 17.3 요약

---



## 17.3 요약

- » for 루프는 루프 변수를 사용해 시퀀스의 원소들을 차례로 가져와 사용한다. 시퀀스에 들어있는 원소는 정수뿐만 아니라 문자도 될 수 있다.
- » 정수가 원소로 들어 있는 시퀀스를 만들고 싶은 경우, range 식을 사용해 특별한 시퀀스를 만들어낼 수 있다.
- » 시퀀스가 문자열인 경우 루프 변수는 문자열 안에 들어 있는 각 문자를 직접 가져온다. 반면 정수 인덱스를 사용해 루프를 돌면 문자열에 루프 변수를 인덱스로 사용하는 단계를 더 거쳐서 문자를 얻어와야 한다.



## 17.3 요약

» (Q17.1) 1부터 100까지(100 포함) 모든 짝수에 대해 루프를 도는 프로그램을 작성하라. 그 수가 6으로 나누어 떨어지면(나머지가 0이라는 뜻이다) 카운터 값을 1 증가시켜라. 루프가 끝나면 프로그램 맨 마지막에서 6으로 나누어 떨어지는 수가 몇 개나 있는지 출력하라.



## 17.3 요약

» (Q17.2) 사용자에게 수를 물어봐서 변수  $n$ 에 저장하는 프로그램을 작성하라. 그 후 루프를 돌면서 메시지를 출력하라. 예를 들어 사용자가 99를 입력하면 다음과 같은 메시지를 출력해야 한다.

파이썬 책 99권이 들어 있는 책장에 파이썬 책이 99권  
책을 한 권 꺼내 돌려 보고 나니 98권이 남았네.

파이썬 책 98권이 들어 있는 책장에 파이썬 책이 98권  
책을 한 권 꺼내 돌려 보고 나니 97권이 남았네.

...

파이썬 책 1권이 들어 있는 책장에 파이썬 책이 1권  
책을 한 권 꺼내 돌려 보고 나니 0권이 남았네.



## 17.3 요약

» (Q17.3) 사용자에게 공백으로 이름들을 구분해 입력하라고 요청하는 프로그램을 작성하라(이름과 이름 사이의 공백은 반드시 한 글자). 프로그램은 사용자가 입력한 모든 사람에게 인사말을 출력한다. 이때 각 인사말을 새줄 문자로 구분하라. 예를 들어 사용자가 ‘소원 예린 은하’라고 입력하면 프로그램은 다음과 같이 출력해야 한다.

안녕, 소원

안녕, 예린

안녕, 은하