

16장

루프로 같은 작업 반복하기

16장 루프로 같은 작업 반복하기

16.1 작업 반복하기

16.2 정해진 횟수 동안 반복하기

16.3 N번 반복하기

16.4 요약

16.1 작업 반복하기



16.1.1 프로그램에 비선형성 추가하기

- » 컴퓨터를 사용하는 경우의 대부분이 짧은 시간에 많은 계산을 수행할 수 있는 컴퓨터의 능력을 활용한다.
- » 컴퓨터는 반복 작업을 아주 잘 수행하므로, 노래 목록에 있는 모든 노래를 차례로 재생하는 작업 등을 아주 쉽게 할 수 있다.
- » 모든 프로그래밍 언어는 컴퓨터가 일련의 명령을 반복 실행하는 방법을 지정해준다.

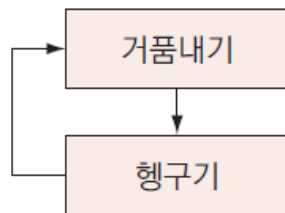


16.1.2 무한 반복

- » 컴퓨터는 명령한 일만 수행하기 때문에 일련의 명령을 몇 번이나 반복해야 할지 스스로 결정할 수 없다. 그래서 컴퓨터에 명령을 반복 수행하라고 명령할 때는 조심해야 한다.
- » 원하는 반복 횟수가 있거나, 다시 반복해야 하는 명확한 조건이 있는가? 거품을 내고 행구기 예제 중에서 ‘필요한 만큼’ 부분이 바로 여러분이 거품 내기와 행구기를 반복할지 결정하는 조건 부분이다.
- » 그런 조건 대신 거품 내기와 행구기를 3번만 반복한 후 그만하라고 할 수도 있다.

16.1.2 무한 반복

A. 거품을 내고 행구는 걸 반복
(반복되는 루프에서 나오는 경로가 없다)



B. 거품을 내고 행구는 걸 필요한 만큼 반복
(반복해야 할지 물어본다)

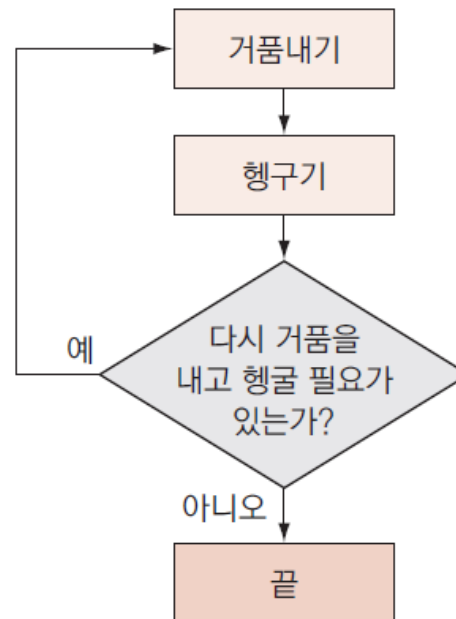


그림 16-1

(A)는 결코 끝나는 일이 없는 ‘거품을 내고 행구는 걸 반복’하는 명령이고, (B)는 매번 반복하기 전에 한 번 더 반복할지 물어보는 ‘거품을 내고 행구는 걸 필요한 만큼 반복한 후 끝내기’ 방식의 명령이다

16.2 정해준 횟수 동안 반복하기



16.2.1 for 루프

- » for는 파이썬에서 명령을 몇 번 반복하라고 지정할 때 사용하는 키워드다.
- » 이 키워드로 명령을 여러 번 반복시키는 방법을 살펴보자.
- » 루프를 사용하지 않는 경우

```
print("야호")
print("야호")
print("야호")
print("야호")
```

- » 루프를 사용하는 경우

```
for i in range(4):
    print("야호")
```



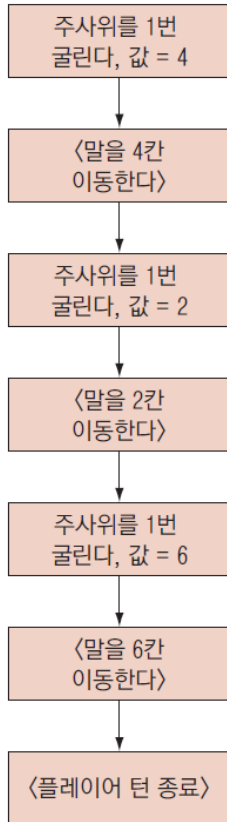

16.2.1 for 루프

» 보드 게임: 자신의 턴마다 주사위를 3번 굴릴 수 있다. 주사위를 굴려 나온 눈 수에 따라 게임판의 말을 이동시킨다. 플레이어가 차례로 4, 2, 6을 굴렸다.

- 플레이어가 주사위를 3번 굴리면 세 값으로 이뤄진 시퀀스를 얻는다. 주사위 눈을 n 으로 표현하자.
- 루프는 주사위를 굴리는 시퀀스를 첫 번째 수부터 하나씩 처리한다. 이 예제에서 첫 번째 수는 $n = 4$ 이다. 그 값에 따라 말을 n 칸 움직인다. 시퀀스의 다음 수는 $n = 2$ 이므로 말을 2칸 이동한다.
- 시퀀스에 있는 마지막 수는 $n = 6$ 이며 그에 맞춰 말을 6칸 움직인다. 시퀀스에 더 이상 수가 남아있지 않으므로 말을 그만 움직이고 플레이어의 턴을 끝낸다.

16.2.1 for 루프

A. 루프를 사용하지 않는 경우



B. for 루프를 사용하는 경우

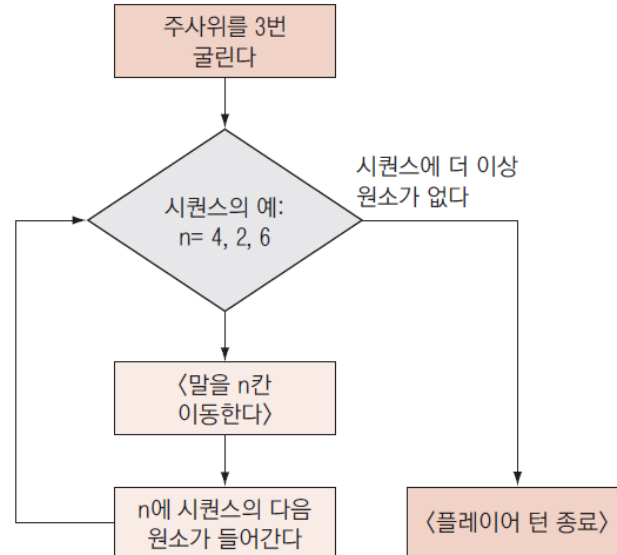


그림 16-2

(A)와 (B) 모두 주사위를 3번 굴려서 4, 2, 6이라는 수를 얻는다. 이런 과정을 (A)와 (B) 모두 똑같이 실행할 수 있다. 하지만 (A)는 명령을 한 단계씩 일일이 작성하여 실행하지만, (B)는 주사위를 던질 때 나오는 눈으로 이뤄진 시퀀스에 루프를 적용해 과정을 반복하고, 시퀀스에 있는 값을 n 이라는 변수에 넣어서 반복 시 바뀌는 값을 일반화함으로써 더 편하게 실행한다



16.2.1 for 루프

for <루프변수> in <시퀀스>: ---- 루프 시작을 나타냄. <루프변수>는 <시퀀스>에 있는 값을 체계적으로 하나씩 가져옴
 <어떤 일을 한다> ---- <시퀀스>에 값이 있는 동안 실행할 코드 블록

코드 16-1

for 루프를 작성하는 일반적인 방법

» for는 정해진 횟수 동안 반복할 블록을 작성하겠다고 파이썬에게 알려주는 키워드다.

- for 키워드다음에는 루프 변수의 이름을 적는다. 루프 변수 이름은 변수 이름이 될 수 있는 올바른 이름이지만 하면, 어떤 이름이라도 상관 없다.
- 루프 본문이 반복 실행될 때마다 루프 변수의 값은 자동으로 시퀀스에 있는 다음 값으로 바뀐다. 이때 루프 변수가 값을 가져올 시퀀스를 지정하기 위해 in 키워드를 적고 그 뒤에 시퀀스나 시퀀스가 들어 있는 변수 이름을 적는다

» 조건문과 마찬가지로 루프에서도 들여쓰기가 중요하다.

» 파이썬은 for보다 더 들여 쓴 블록은 모두 for 루프가 반복해야 할 루프 본문으로 취급한다.

16.2.1 for 루프

» 여기서 핵심은 값으로 이뤄진 시퀀스를 사용한다는 점이다. 루프 본문은 시퀀스에 값이 있는 동안 반복 실행된다. 시퀀스에 있는 값을 모두 처리하면 루프도 반복을 끝낸다.

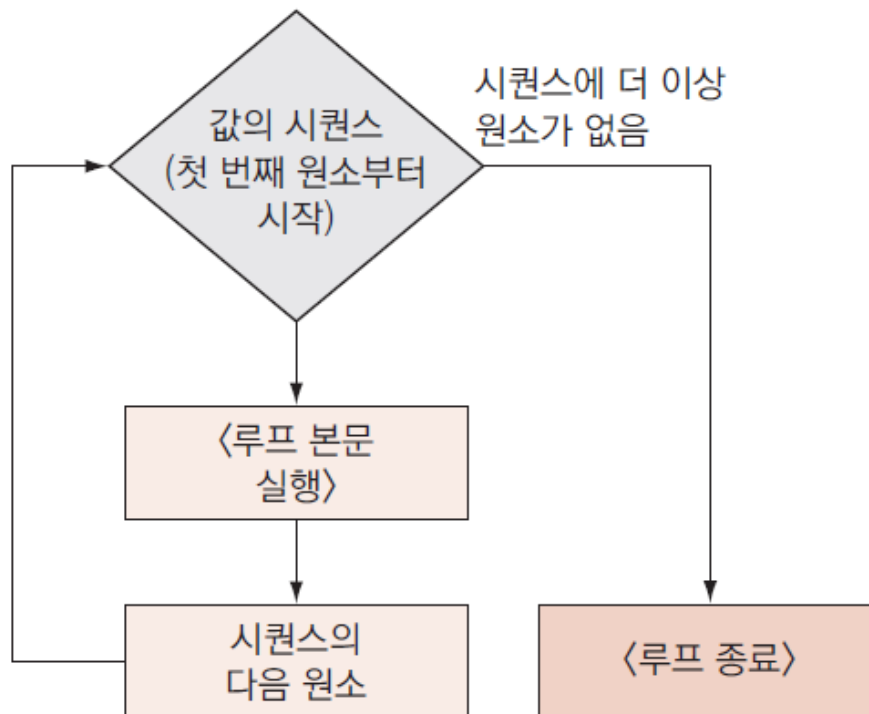


그림 16-3
for 루프를 작성하는 일반적인 방법. 값 시퀀스의 첫 번째 원소부터 시작해 루프 본문을 실행한다. 그 후 시퀀스에서 차례대로 다음 원소를 가져와서 루프 본문을 실행한다. 시퀀스에 있는 모든 원소를 사용해 본문을 실행하면 루프가 끝난다

16.3 N번 반복하기



16.3.1 0부터 N-1까지 수로 이뤄진 시퀀스에 대해 루프 돌기

- » 루프를 N번 돌고 싶다면 코드 16-1의 <시퀀스> 부분을 `range(N)`이라는 식으로 바꾸면 된다(N은임의의 정수).
- » `range`는 파이썬이 제공하는 특별한 명령이다. `range(N)`은 0, 1, 2, ..., N-1이라는 시퀀스를 만들어낸다

```
for v in range(3): ---- v는 루프 변수
    print("var v is ", v) ---- 루프 변수의 값을 출력
```

코드 16-2

루프 변수의 값을 출력하는 for 루프

- » 코드 16-2에서 루프 변수는 `v`, 루프 변수가 값을 가져올 시퀀스는 `range(3)`, 루프 본문은 `print` 문이다.

- 코드 16-2에 있는 프로그램을 실행하면 첫 줄에 있는 for 루프가 `range(3)`으로 시작된다.
- for 루프는 0을 `v`에 저장하고(0은 `range(3)`의 첫 번째 원소다), `print` 문을 실행한다.
- 그 후 시퀀스의 두번째 원소인 1을 루프 변수 `v`에 넣고 다시 `print` 문을 실행한다.
- 그 후 2를 `v`에 넣고 또 `print` 문을 실행한다. 이 루프는 3번 반복하면서 `range(3)`에 있는 0, 1, 2를 루프 변수에 설정한다.



16.3.1 0부터 N-1까지 수로 이뤄진 시퀀스에 대해 루프 돌기

» 3이라는 고정된 수 대신 `n_times` 같은 변수를 써서 시퀀스의 길이를 `range(n_times)`로 지정하면 코드 16-2의 동작을 일반화할 수 있다. 그 경우 루프는 `n_times`번 반복된다.

- 루프가 한 번 돌 때마다, 루프 변수에 다른 값이 들어간 상태로 코드 블록 안에 있는 문장이 실행된다

» 셀프 체크 16.2



16.3.2 루프 펼치기

- » 루프를 반복이 아닌 다른 관점으로 볼 수도 있다. 코드 16-3은 루프를 펼쳐서(unrolling) 코드 16-2에 있는 파이썬 코드가 정확히 어떤 일을 하는지 보여준다.
- » 코드 16-3을 보면 변수 `v`에 매번 다른 값이 대입되는 걸 알 수 있다. 하지만 `v`값이 바뀔 때마다 실행되는 `print`문은 모두 같다.
 - 이 코드는 비효율적이며 지겹고, `v`를 출력하는 코드를 여러 번 반복해서 입력해야 하기 때문에 실수하기도 쉽다.
 - 이런 코드 대신 루프를 사용하면 코드를 훨씬 빠르게 작성할 수 있고, 읽기도 쉽다.

```

v = 0 ---- 변수 v(여기서는 0이 대입됨)는 코드 16-2에서 루프 변수임
print("var v is", v)
v = 1 ---- 변수 v의 값을 직접 1로 바꿈
print("var v is", v)
v = 2 ---- 변수 v의 값을 직접 2로 바꿈
print("var v is", v)
    
```

코드 16-3
코드 16-2에 있는 루프를 펼친 모습

16.4 요약



16.4 요약

- » 루프는 간결하고 읽기 좋은 코드를 만들 때 유용하다.
- » for 루프는 원소가 들어 있는 시퀀스에서 값을 하나씩 가져오는 루프 변수를 사용한다.
물론 정수가 원소인 시퀀스를 사용할 수도 있다.
- » 0부터 시작하는 연속적인 정수로 이뤄진 시퀀스를 range라는 특별한 식을 통해 표현할 수 있다.



16.4 요약

» (Q16.1) 사용자에게 숫자를 하나 물어보고, 사용자가 입력한 수만큼 ‘안녕’이라고 출력하는 루프를 작성하라. 그리고 for 루프 없이 이런 코드를 작성할 수 있을지 생각해 보라