

# 4

## 반복

- 학/습/목/표**
- 반복의 개념을 이해할 수 있다.
  - 반복 구조를 활용한 프로그램을 작성할 수 있다.
  - 중첩 제어 구조를 활용한 프로그램을 작성할 수 있다.

**핵/심/용/어** 반복, while, for

프린터는 문서 작성 프로그램으로 만든 문서를 지시한 매수만큼 반복적으로 인쇄하고, 엘리베이터는 사용자가 원하는 층까지 한 층씩 올라가거나 내려가는 행동을 반복한다. 이처럼 동일한 방식으로 계속해서 반복되는 것을 프로그래밍에서는 어떻게 처리하는지 알아보자.



❓ 동일한 작업을 여러 번 반복하는 프로그램은 어떻게 만드는 것일까?

### 1 반복

**반복**이란 어떤 일을 계속해서 되풀이하는 것으로, 우리의 일상생활 속에서 반복되는 일들은 쉽게 찾을 수 있다.

#### 해보기

#### 일상생활에서 반복 찾기

분석하기

- 우리의 일상생활에서 반복되는 일을 찾아 적어 보자.

예) 20회가 될 때까지 팔 굽혀 펴기를 계속한다.

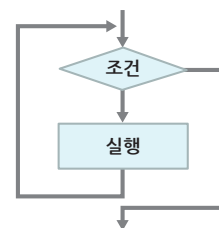
- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_

만약 반복되는 일이 컴퓨터로 처리 가능한 일이라면, 반복 구조를 적용한 프로그램을 개발함으로써 사람을 대신해서 컴퓨터가 처리하도록 할 수 있다. 컴퓨터의 빠른 처리 속도를 이용해서 똑같은 계산을 반복적으로 수행하기 때문에 많은 양의 자료도 원하는 횟수만큼 빠르게 처리할 수 있다.

예를 들어 이동 통신사에서 가입자들의 핸드폰 요금을 계산한다고 할 때, 천만 명이 넘는 가입자들의 요금을 사람이 직접 계산하는 것은 거의 불가능하다.

#### 반복 구조

어떤 처리를 여러 번 반복해서 실행하는 프로그램 구조를 '반복 구조'라고 한다.



하지만 컴퓨터가 반복 계산하도록 프로그램을 개발하면, 사람이 계산하는 것과는 비교할 수 없을 정도로 빠르게 요금을 계산할 수 있다. 일반적으로 반복 구조를 적용한 프로그램을 만들 때는 while 반복문과 for 반복문을 이용한다.

## 2 while을 이용한 반복 구조

while은 조건식을 이용해서 어떤 일을 반복하도록 할 때 사용하는 반복문으로, 파이선 언어에서는 다음과 같은 구조로 되어 있다.

### • 탭(TAB) 들여쓰기

파이선에서는 while 반복문에서 반복 실행하는 영역의 문장을 탭 키 또는 스페이스 키 4개로 들여쓰기한다.

while (조건식) :

    반복 실행 영역

while (조건식) :

반복 실행 영역

while은 조건식의 결과가 참인 동안 ‘반복 실행 영역’을 계속 실행하다가, 조건식의 결과가 거짓이 되는 순간 반복을 멈추게 된다. 우리의 생활 속에서 ‘~가 될 때까지’ 또는 ‘~하는 동안’ 등의 경우가 많이 있는데, 이런 경우를 프로그래밍할 때 while 반복문이 사용된다.

다트 게임의 점수 합이 20점 이상이 될 때까지 다트를 던져 얻은 점수를 계속 출력하다가, 20점을 넘으면 합계 점수를 출력하고 종료하는 프로그램을 while 반복문으로 만들어 보자. 먼저 게임 상황을 분석해서 반복되는 요소를 찾아야 한다.

게임을 반복하는 조건	점수의 합계 < 20
게임에서 반복하는 일	<ul style="list-style-type: none"> <li>• 다트를 던지고 점수를 입력한다.</li> <li>• 입력된 점수를 출력한다.</li> <li>• 입력된 점수를 합계 점수에 더한다.</li> </ul>



분석한 내용을 while 반복문으로 표현하면 다음과 같다.

while (점수의 합계 < 20) :

    다트를 던지고 점수를 입력한다.

    입력된 점수를 출력한다.

    입력된 점수를 합계 점수에 더한다.

이 프로그램은 다트를 던져 얻은 점수와 합계 점수를 저장해야 하므로 두 개의 변수가 필요하다.

point: 다트를 던져 얻은 점수

sum: 얻은 점수들의 합계 점수

분석한 내용과 변수를 이용해서 프로그램을 만들면 다음과 같다.

#### 프로그램 코드

```
1 point=0 # 입력된 점수를 저장하는 변수
2 sum=0 # 합계 점수를 저장하는 변수
3 while (sum < 20) : # 합계 점수가 20점 미만이면
4     point=input("다트 점수를 입력하세요.:") # 점수를 입력
5     point=int(point) # 입력된 점수를 숫자형으로 변환
6     print("이번 점수는", point) # 입력된 점수를 출력
7     sum=sum + point # 입력된 점수를 합계 점수에 누적
8 print("합계 점수는", sum) # 최종 점수를 출력
```

프로그램 코드에서 4~7번 줄이 조건식이 참인 동안 반복 실행되는 영역이고, 탭 들여쓰기가 되어 있다.

while 반복문은 조건식에 의해 반복이 결정되기 때문에 조건식에 오류가 없어야 한다. 조건식에 오류가 있으면, ‘반복 실행 영역’이 한 번도 실행되지 않거나 영원히 반복되는 경우가 발생할 수도 있기 때문이다. 만약 조건식에 오류가 있어서 결과가 계속 참이 된다면, while 반복문을 빠져나오지 못하고 영원히 반복되는 현상이 생기는데, 이것을 ‘무한 루프’라고 한다. 다음 프로그램 코드는 무한 루프의 예를 보여 준다.

#### 프로그램 코드

```
1 num=1
2 while (num <= 100) :
3     print("현재 숫자는", num)
4 print("프로그램 종료")
```

#### ●무한 루프

무한 루프는 개발자가 필요에 의해 만들어 사용하는 경우도 있지만, 일반적으로는 조건식에 논리 오류가 있을 때 발생한다.

이 프로그램 코드는 1부터 100까지의 숫자를 출력한 다음 종료하는 것을 목표로 만들어진 것이다. 하지만 num 변수의 값이 변하지 않고 계속해서 1이기 때문에 조건식은 영원히 참이 되고, 3번 줄의 코드가 무한 반복 실행된다. 그 결과, 4번 줄의 print() 명령어는 실행되지 않는다. 이 프로그램이 원래의 목적대로 실행되기 위해서는 3번 줄 다음에 num 변수의 값을 증가시키는 코드가 필요하다.

#### 프로그램 코드

```
1 num=1
2 while (num <= 100) :
3     print("현재 숫자는", num)
4     num=num + 1
5 print("프로그램 종료")
```

이처럼 while 반복문을 사용할 때에는 조건식에 논리 오류가 없는지를 잘 살펴서 무한 루프에 빠지지 않도록 주의해야 한다.



## 공의 개수 세기

분석하기 + 설계하기 + 프로그래밍하기

- 학교에 있는 공의 개수를 파악하는 데 사용할 프로그램을 만들려고 한다. while 반복문을 이용해서 다음 조건을 만족하는 프로그램을 만들어 보자.

### 조건

- 프로그램이 입력 대기 상태에 있을 때 '농구공, 축구공, 야구공' 중 하나를 입력하면, 입력한 공의 개수가 1개씩 증가하고, '종료'를 입력하면 프로그램이 종료된다.
- '농구공, 축구공, 야구공, 종료' 이외의 값을 입력하면 잘못된 입력임을 알리고, 다시 입력하도록 한다.

농구공	농구공의 개수를 1 증가시킨다.
축구공	축구공의 개수를 1 증가시킨다.
야구공	야구공의 개수를 1 증가시킨다.
종료	프로그램을 종료한다.
그 외의 입력	정확한 입력을 요구한다.

- 1 이 프로그램에는 몇 개의 변수가 필요하고, 각 변수의 용도는 무엇인지 적어 보자.

- 2 빈칸을 채워 프로그램 코드를 완성해 보자.

### 프로그램 코드

```
1 kind=""
2 _____=0
3 _____=0
4 _____=0
5 while ( _____ ) :
6     kind=input("입력하세요. [농구공, 축구공, 야구공, 종료]:")
7     if (kind=="농구공") :
8         _____
9     elif (kind=="축구공") :
10        _____
11    elif (kind=="야구공") :
12        _____
13    elif (kind=="종료") :
14        break
15    else :
16        print("잘못된 입력입니다.")
17    print("농구공의 개수:", _____)
18    print("축구공의 개수:", _____)
19    print("야구공의 개수:", _____)
```

# 반복문이나 조건문을 빠져나갈 때 사용하는 명령어

### 3 for를 이용한 반복 구조

for는 반복 횟수를 이용해서 어떤 일을 반복하도록 할 때 사용하는 반복문으로, 파이선 언어에서는 다음과 같은 구조로 되어 있다.

for 변수 in range(시작값, 끝값, 증감값) :

반복 실행 영역

for 반복문은 변수의 값이 시작값부터 끝값까지 증가 또는 감소하면서 반복하는 횟수가 결정되고, 그 반복 횟수만큼 ‘반복 실행 영역’을 처리한다. 예를 들어 변수 i의 값이 1부터 10까지 바뀌면서 10번 반복 실행해야 하는 경우는 다음과 같이 for 반복문을 구성한다.

프로그램 코드	실행 결과
<pre>1 for i in range(1, 11, 1) : 2     print("i=", i)</pre>	<pre>i=1 i=2 : i=10</pre>

위의 프로그램 코드를 실행하면, 변수 i에 시작값인 1이 저장된 다음 반복 실행 영역인 print() 명령어를 수행한다. 반복 실행 영역이 수행되고 나면, 변수 i를 증감값 1만큼 증가시켜 2가 저장되고 다시 반복 실행 영역을 수행한다. 이런 방식으로 반복해서 변수 i가 10이 될 때까지 10번 반복 실행 영역을 수행하면 프로그램이 종료된다.



더 알아보기

for 반복문에서 range() 함수 사용하기

파이선 언어로 for 반복문을 사용할 때는 range() 함수에 의해 사용할 숫자를 제공받기 때문에, 원하는 범위의 수를 사용하기 위해서는 range() 함수를 이해해야 한다.

사용 예	설명
for i in range(n)	0부터 (n-1)까지의 숫자만큼 반복한다. 예 for i in range(10) ← 변수 i에 0~9까지 10개의 숫자를 사용하면서 반복
for i in range(m,n,1)	m부터 (n-1)까지 1씩 증가하는 숫자만큼 반복한다. 예 for i in range(1, 51, 1) ← 변수 i에 1부터 50까지 50개의 숫자를 사용하면서 반복
for i in range(m,n,-1)	m부터 (n+1)까지 1씩 감소하는 숫자만큼 반복한다. 예 for i in range(50, 0, -1) ← 변수 i에 50부터 1까지 50개의 숫자를 사용하면서 반복

#### • 탭(TAB) 들여쓰기

파이선에서는 for 반복문에서 반복 실행하는 영역의 문장을 탭 키 또는 스페이스 키 4개로 들여쓰기한다.

for i in range(시작값, 끝값, 증감값) :

반복 실행 영역

우리의 일상생활에서는 반복 횟수가 정해져 있어서 for 반복문을 적용할 수 있는 경우를 쉽게 찾아볼 수 있다. 팔 굽혀 펴기를 20회 하는 경우라든가, 1번 학생부터 30번 학생까지 차례대로 수행 평가를 실시하고 점수를 기록하는 경우 등은 모두 for 반복문으로 표현할 수 있다.

이번에는 다트를 20번 던져 얻은 점수를 계속 출력하고, 점수를 모두 더한 최종 점수를 출력한 다음 종료하는 프로그램을 for 반복문으로 만들어 보자. 먼저 게임 상황을 분석해서 반복되는 요소를 찾아야 한다.



시작값	1
끝값	21
증감값	+1
게임에서 반복하는 일	<ul style="list-style-type: none"> <li>• 다트를 던지고 점수를 입력한다.</li> <li>• 입력된 점수를 출력한다.</li> <li>• 입력된 점수를 최종 점수에 더한다.</li> </ul>



분석한 내용을 for 반복문으로 표현하면 오른쪽과 같다.

```
for i in range(1, 21, 1):
    다트를 던지고 점수를 입력한다.
    입력된 점수를 출력한다.
    입력된 점수를 최종 점수에 더한다.
```

### 파이선 명령어

str()

숫자형 자료를 문자형으로 변환할 때 사용하는 파이선 명령어이다.

이 프로그램은 다트를 던져 얻은 점수와 최종 점수를 저장해야 하므로 두 개의 변수가 필요하다.

point: 다트를 던져 얻은 점수  
total: 얻은 점수들을 모두 더한 최종 점수

분석한 내용과 변수를 이용해서 프로그램을 만들면 다음과 같다.

### 프로그램 코드

```
1 point=0 # 입력된 점수를 저장
2 total=0 # 최종 점수를 저장
3 for i in range(1, 21, 1): # 1부터 20까지 20회 반복
4     text=str(i) + "회 다트 점수를 입력하세요.:" # 입력 안내
5     point=input(text) # 점수를 입력
6     point=int(point) # 입력된 점수를 숫자형으로 변환
7     print(i, "회 점수는", point) # 입력된 점수를 출력
8     total=total + point # 입력된 점수를 최종 점수에 누적
9     print("최종 점수는", total) # 최종 점수를 출력
```

오른쪽 프로그램은 변수 i의 값이 1~20까지 변하면서 '반복 실행 영역(4~8번 줄)'을 반복해서 실행하게 된다.

변수 i의 값이 1일 때 4~8번 줄 실행  
변수 i의 값이 2일 때 4~8번 줄 실행  
변수 i의 값이 3일 때 4~8번 줄 실행  
...  
변수 i의 값이 19일 때 4~8번 줄 실행  
변수 i의 값이 20일 때 4~8번 줄 실행

위의 프로그램 코드에서는 1부터 20까지의 숫자가 변수 i에 저장되어 사용되는데, for 반복문에서는 이 변수를 이용해서 여러 가지 처리를 할 수 있다. 위의 for 반복문에서는 변수 i로 '몇 회'라는 정보를 생성해 사용하고 있다.



## 숫자 더하기

분석하기 + 설계하기 + 프로그래밍하기

- for 반복문을 이용해서 다음 조건을 처리하는 프로그램을 만들어 보자.

### 1 1부터 15까지의 숫자를 화면에 출력하는 프로그램을 만들어 보자.

- (1) 반복하는 상황을 분석해 보자.

시작값		끝값		증감값	
반복하는 일					

- (2) 분석한 내용을 프로그램 코드로 만들어 보자.

#### 프로그램 코드

```

1 for i in range(____, _____, _____) :      # 1부터 15까지 1씩 증가
2     print(____)

```

- (3) 위의 프로그램 코드를 200부터 10까지 5씩 감소하는 숫자를 출력하는 프로그램으로 고쳐 보자.

#### 프로그램 코드

```

1 for i in range(____, _____, _____) :      # 200부터 10까지 5씩 감소
2     print(____)

```

### 2 1부터 100까지 2씩 증가하는 숫자를 차례대로 화면에 출력하고, 마지막에는 그 숫자를 모두 더한 값을 출력하는 프로그램을 만들어 보자.

- (1) 반복하는 상황을 분석해 보자.

시작값		끝값		증감값	
반복하는 일					

- (2) 분석한 내용을 프로그램 코드로 만들어 보자.

#### 프로그램 코드

```

1 sum=0
2 for i in range(____, _____, _____) :      # 1부터 100까지 2씩 증가
3     _____                                     # i에 저장되어 있는 숫자를 출력
4     _____                                     # i에 저장된 숫자를 sum에 누적
5 print(____)                                           # sum을 출력

```



## ○○텔레콤 요금 관리 프로그램 만들기

분석하기 + 설계하기 + 프로그래밍하기

- ○○텔레콤에서는 회원들의 통화 시간과 요금을 관리하기 위한 프로그램을 개발하려고 한다. 오른쪽과 같이 월별 통화 시간을 입력하면 월별 사용 요금을 계산해서 출력하고, 최종적으로 1년 동안의 통화 시간과 사용 요금을 출력하는 프로그램을 만들어 보자.

### 프로그램의 실행 과정

- 매월 통화 시간을 입력한다. (단위: 분)
- 월별 사용 요금은 (매월 통화 시간 × 단위 요금)으로 계산한다. (1초에 2원)
- 매월 통화한 시간과 요금을 출력한다.
- 1년 동안의 통화 시간과 사용 요금을 출력한다.

- 1 프로그램이 처리해야 하는 일은 무엇인지 작은 문제로 분해해 보자.

- 2 프로그램이 처리해야 하는 일을 반복하는 일과 반복하지 않는 일로 각각 구분해 보자.

반복하는 일	반복하지 않는 일

- 3 프로그램을 만드는 데 필요한 핵심요소를 추출해 보자.

작은 문제	핵심요소



#### 4 프로그램에서 사용되는 변수의 이름과 용도를 정해 보자.

변수의 이름	변수의 용도

#### 5 프로그램의 알고리즘을 순서도로 표현해 보자.

#### 6 프로그램 코드를 만들어 실행해 보자.

[실행 화면 1]

```

1월 통화 시간(분)을 입력하세요.:210
1월 통화 시간: 210
1월 통화 요금: 25200
2월 통화 시간(분)을 입력하세요.:120
2월 통화 시간: 120
2월 통화 요금: 14400
3월 통화 시간(분)을 입력하세요.:110
3월 통화 시간: 110
3월 통화 요금: 13200
4월 통화 시간(분)을 입력하세요.:150
4월 통화 시간: 150
  
```

[실행 화면 2]

```

10월 통화 시간: 170
10월 통화 요금: 20400
11월 통화 시간(분)을 입력하세요.:175
11월 통화 시간: 175
11월 통화 요금: 21000
12월 통화 시간(분)을 입력하세요.:195
12월 통화 시간: 195
12월 통화 요금: 23400

1년 동안의 통화 시간: 2025
1년 동안의 통화 요금: 243000
  
```