

# 파이썬 프로그래밍

5. 반복문

## ❖ 수업 목표

- 반복문의 동작순서를 설명할 수 있다.
- range 함수를 사용할 수 있다.
- for문과 while문 문법을 설명할 수 있다.

## ❖ 세부 목표

- 5.1 for문
- 5.2 range
- 5.3 continue
- 5.4 break
- 5.5 while
- 5.6 중첩 for문

## 1. for 문

### ❖ for 문

- 반복 횟수가 정해진 경우 사용
- 시퀀스(sequence) 자료형과 함께 사용
  - 문자열, 리스트, 튜플 등 순서가 있고 길이가 있는 자료형

# 1. for 문

## ❖ for

- 반복 작업에 사용하는 키워드
- 리스트의 각 값을 순서대로 가져와 반복 작업 처리



- **코드블록 (Code Block)**
  - 실행할 명령
  - 리스트의 마지막 값 가져올 때까지 코드블록 반복

# 1. for 문

- in 이후에 리스트 대신 이를 저장한 변수가 올 수도 있음

코드 4-4 리스트의 문자열을 차례대로 출력하는 코드

```
01: characters = ['앨리스', '도도새', '3월토끼']
02: for character in characters:
03:     print(character)
```

↳ 실행화면

```
앨리스
도도새
3월토끼
```

# 1. for 문

- ❖ 문자열 반복하기
- 문자열을 for과 함께 사용할 수 있음

for 변수 in 문자열:  
      실행할\_명령

코드 4-6 문자열의 문자를 하나씩 출력하는 코드

```
01: for letter in '체셔고양이':  
02:     print(letter)
```

↳ 실행화면

체  
셔  
고  
양  
이

# 1. for 문

## ❖ 들여쓰기

- 코드블록을 구분하기 위해 사용
- 띄어쓰기 4칸 혹은 탭(tab)
- 하나의 프로그램 안에서는 한 가지 방법으로 띄어쓰기 통일해야

코드 4-7 리스트의 숫자를 차례대로 출력하는 코드

```

01: nums = [0, 1, 2]
02: for num in nums:
03:     print(num)
04: print(nums)
    
```

↳ 실행화면

```

0
1
2
[0, 1, 2]
    
```

코드 4-8 리스트의 숫자를 차례대로 출력하는 코드

```

01: nums = [0, 1, 2]
02: for num in nums:
03:     print(num)
04: print(nums)
    
```

↳ 실행화면

```

0
[0, 1, 2]
1
[0, 1, 2]
2
[0, 1, 2]
    
```

- 코드 4-7은 2번과 3번 라인 모두 실행한 후 4번 라인 실행

## 2. range

### ❖ range() 함수

- 연속된 숫자(정수)를 만들어 줌
- 내장 함수
- 사용 형태

- range(숫자) : 0부터 해당하는 [숫자-1]까지의 정수를 만들어줌
- 예) range(10) : 0부터 9까지의 정수      # 0,1,2,3,4,5,6,7,8,9

- 확인방법
  - 내용을 리스트로 변환하여 만들어진 숫자 확인 가능
  - list() 함수를 사용하여 변환
  - 예) list(range(숫자))

## 2. range

### ■ 인수가 2개인 경우

- **range(시작숫자, 끝숫자)** : 시작숫자부터 [끝숫자-1]까지의 정수 생성
  - 예) `range(3,10)` : 3부터 9까지의 정수 # 3,4,5,6,7,8,9

### ■ 인수가 3개인 경우

- **range(시작숫자, 끝숫자, 증가값)** : 시작숫자부터 [끝숫자-1]까지, 증가값만큼 증가하며 정수 생성
  - 예) `range(8,14, 2)` : 8부터 13까지 2씩 증가하는 정수 # 8,10,12

## 3. While 문

### ❖ while

- 반복 작업에 사용하는 키워드
- 조건을 더는 만족하지 않을 때까지 반복

**while 조건:**

실행할\_명령 ————— ② 코드블록  
 ↪————— ① 들여쓰기

코드 6-1 `for`를 사용해 거북이에게 인사하는 코드

```
01: for num in range(3):
02:     print('안녕 거북이', num)
```

코드 6-2 `while`을 사용해 거북이에게 인사하는 코드

```
01: num = 0
02: while num < 3:
03:     print('안녕 거북이', num)
04:     num = num + 1
```

↳ 실행화면

```
안녕 거북이 0
안녕 거북이 1
안녕 거북이 2
```

### 3. While 문

- ❖ while의 조건을 True로 고정
- 무한으로 코드블록을 반복

코드 6~9 끝없이 반복해서 출력하는 코드

```
01: while True:  
02:     print('Ctrl+C를 누르세요.')
```

### 3. While 문

#### ❖ 복합 대입(할당) 연산자

- 연산과 할당을 결합한 형태
- l\_value와 r\_value에 동일한 변수가 반복되는 경우 간단히 표현 가능

a=a - b → a -=b	stra="파이썬" strb="프로그래밍" stra=stra+strb → stra += strb
-----------------	---

## 4. continue

### ❖ continue

- 반복의 처음으로 돌아가는 키워드
  - **continue** 문을 만나면 아래의 코드 블록은 실행 안함

#### 코드 6-6 count가 2일 때 넘어가는 코드

```
01: count = 0
02: while count < 3:
03:     count = count + 1
04:     if count == 2:
05:         continue
06:     print(count)
```

#### ↳ 실행화면

```
1
3
```

## 5. break

### ❖ break

#### ■ 반복을 종료하는 키워드

- break문을 만나면 반복문을 중단하고 반복문 외부 코드를 이어서 실행

코드 6-7 count가 2일 때 멈추는 코드

```
01: count = 0
02: while count < 3:
03:     count = count + 1
04:     if count == 2:
05:         break
06:     print(count)
```

↳ 실행화면

1

## 6. 이중 for 문

### ❖ 이중 for 문

- for문 내 코드블록에 다른 for문을 중첩하여 사용
  - 반복문이 다른 반복문에 의해 반복적으로 실행됨을 의미
- 
- 예제) \* 문자를 한 줄에 10개씩 5줄 출력하기

```
for j in range(5):
    for i in range(10):
        print("*", end="")
    print()
```

## 7. 연습문제

❖ 그림과 같은 실행화면을 출력하도록 빈칸을 채워보세요

```
01: for num in ( ):  
02:     print(num)
```

4 3  
1  
2

❖ 다음 코드의 출력 결과를 적어보세요

```
01: for num in range(2):  
02:     print(num)
```

4

## 7. 연습문제

❖ **print()를 한번만 사용해서 아래 코드와 같은 결과가 나오게 하세요**

```
01: clovers = ['클로버1', '클로버2', '클로버3']
02: print(clovers[0])
03: print(clovers[1])
04: print(clovers[2])
```

```
↳ 클로버1  
클로버2  
클로버3
```

```
01: clovers = ['클로버1', '클로버2', '클로버3']
02: 
03: 
```

## 7. 연습문제

❖ 이전 페이지의 문제를 `range()`를 사용해 해결해보세요

```
01: clovers = ['클로버1', '클로버2', '클로버3']  
02:   
03: 
```

```
↳ 클로버1  
클로버2  
클로버3
```

## 7. 연습문제

### ❖ 다음 코드의 출력 결과를 적어보세요

```
01: count = 0
02: while count < 3:
03:     print(count)
04:     count = count + 1
```

↳

## 7. 연습문제

❖ 그림과 같은 실행화면을 출력하도록 빈칸을 채워보세요

```
01: count = 1
02: while [ ]:
03:     count = count + 1
04:     print(count)
```

- ↳ 2
- 3
- 4

## 7. 연습문제

### ❖ 다음 코드의 출력 결과를 적어보세요

```
01: count = 0
02: while count <= 5:
03:     if count % 2 != 0:
04:         print(count)
05:     count = count + 1
```

↳

## 7. 연습문제

### ❖ 아래와 같은 실행화면을 출력하도록 빈칸을 채워보세요

```
01: price = 0
02: while [ ] :
03:     price = int(input('가격을 입력하세요 (종료:-1): '))
04:     if price > 10000:
05:         print('너무 비싸요.')
06:     elif price > 5000:
07:         print('괜찮은 가격이네요.')
08:     elif price > 0:
09:         print('정말 싸요.'
```

```
↳ 가격을 입력하세요 (종료:-1): 3000 ↴
정말 싸요.

가격을 입력하세요 (종료:-1): 15000 ↴
너무 비싸요

가격을 입력하세요 (종료:-1): 7500 ↴
괜찮은 가격이네요.

가격을 입력하세요 (종료:-1): -1 ↴
```

## 7. 연습문제

- ❖ 0부터 100까지의 정수 중 3의 배수만 출력하는 프로그램을 작성하세요.



## 7. 연습문제

- ❖ 3, 6, 9 게임 만들기
- ❖ `continue` 문을 이용하여 1부터 10까지의 수를 차례대로 출력하지만 3의 배수는 출력하지 않는 프로그램을 작성하세요.

## 7. 연습문제

- ❖ 다음과 같이 화면 출력 결과가 나올 수 있게 입력받은 정수까지의 총합을 구하는 프로그램을 작성해 보세요.

〈화면 출력〉

총합을 구하려는 수를 입력하세요. 10  
1부터 10 까지의 총합은 55 이다.

## 7. 연습문제

❖ 다음 프로그램을 확인하고  
화면 출력 결과를 각각  
아래 칸에 써 보세요.

```
for sb in range(1, 11, 1):
    total=0
    total=total+sb
    print(total)
```

```
total=0
for sb in range(1, 11, 1):
    total=total+sb
print(total, end=" ")
print("끝")
```

```
total=0
for sb in range(1,11,1):
    total=total+sb
print(total)
```

```
total=0
for sb in range(1,11,1):
    total=total+1
print(total)
```

```
for sb in range(1,11,1):
    total=0
    total=total+1
print(total)
```

## 7. 연습문제

- ❖ 1부터 입력받은 정수까지의 총합을 구하는 프로그램을 while 문으로 작성하세요.

〈화면 출력〉

총합을 구하려는 수를 입력하세요. 10  
1부터 10 까지의 총합은 55 이다.

## 7. 연습문제

- ❖ 앞서 작성한 1부터 임의의 수까지의 총합을 구하는 프로그램을 break 문을 사용한 것으로 변경하세요

〈화면 출력〉

총합을 구하고 싶은 정수를 입력하세요. 10  
1부터 10 까지의 총합은 55 이다.

## 7. 연습문제

- ❖ 총합을 구할 첫 번째 수와 두 번째 수를 입력 받아 총합을 구하는 프로그램을 `for` 문과 `while` 문으로 작성하세요.
  - 계산할 첫 번째 수를 입력한다.
  - 계산할 마지막 수를 입력한다.

〈for 문〉

〈while 문〉

## 7. 연습문제

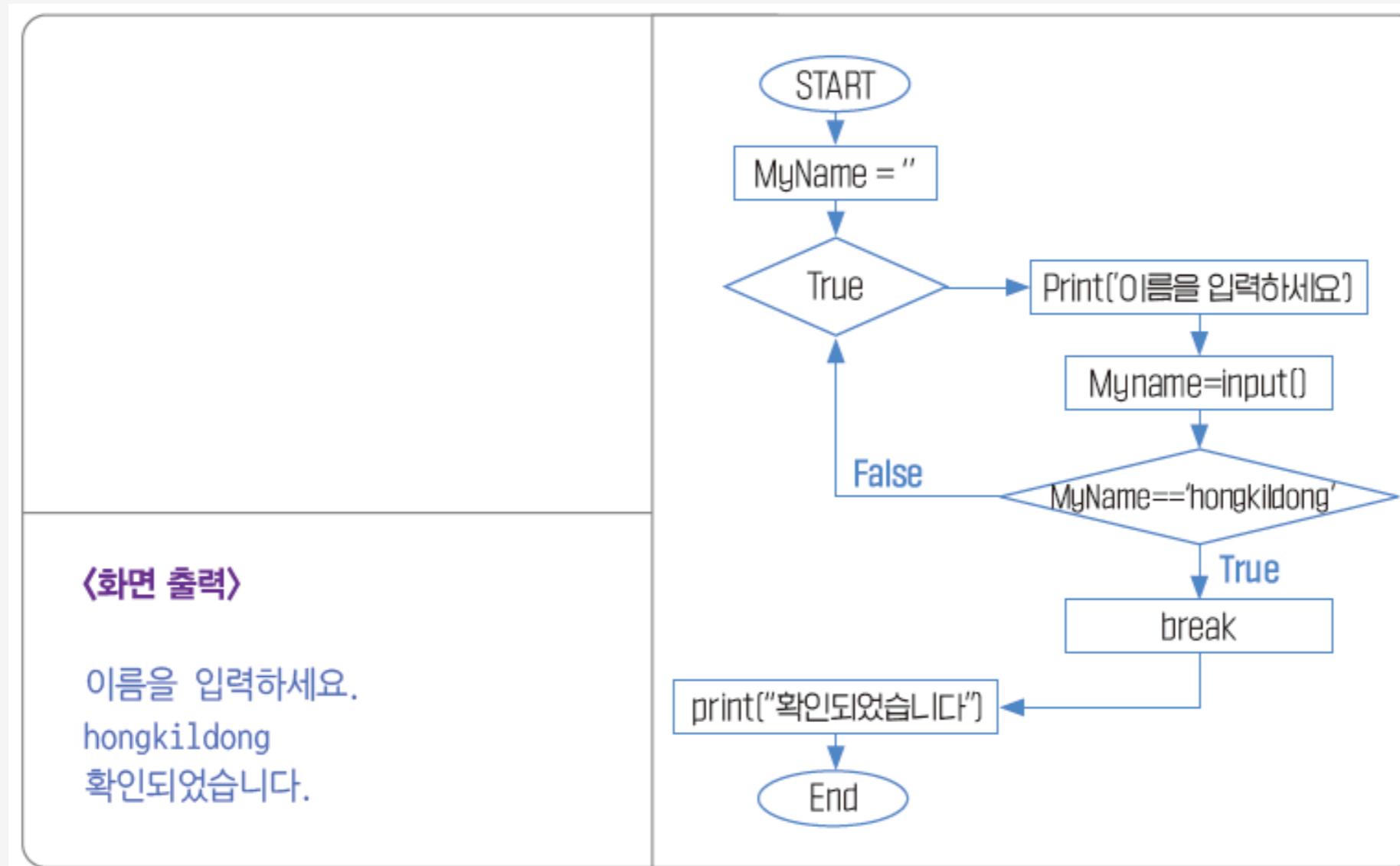
- ❖ 앞서 작성한 프로그램을 3의 배수의 총합을 구하는 프로그램을 **for** 문과 **while** 문으로 변경하세요.

〈for 문〉

〈while 문〉

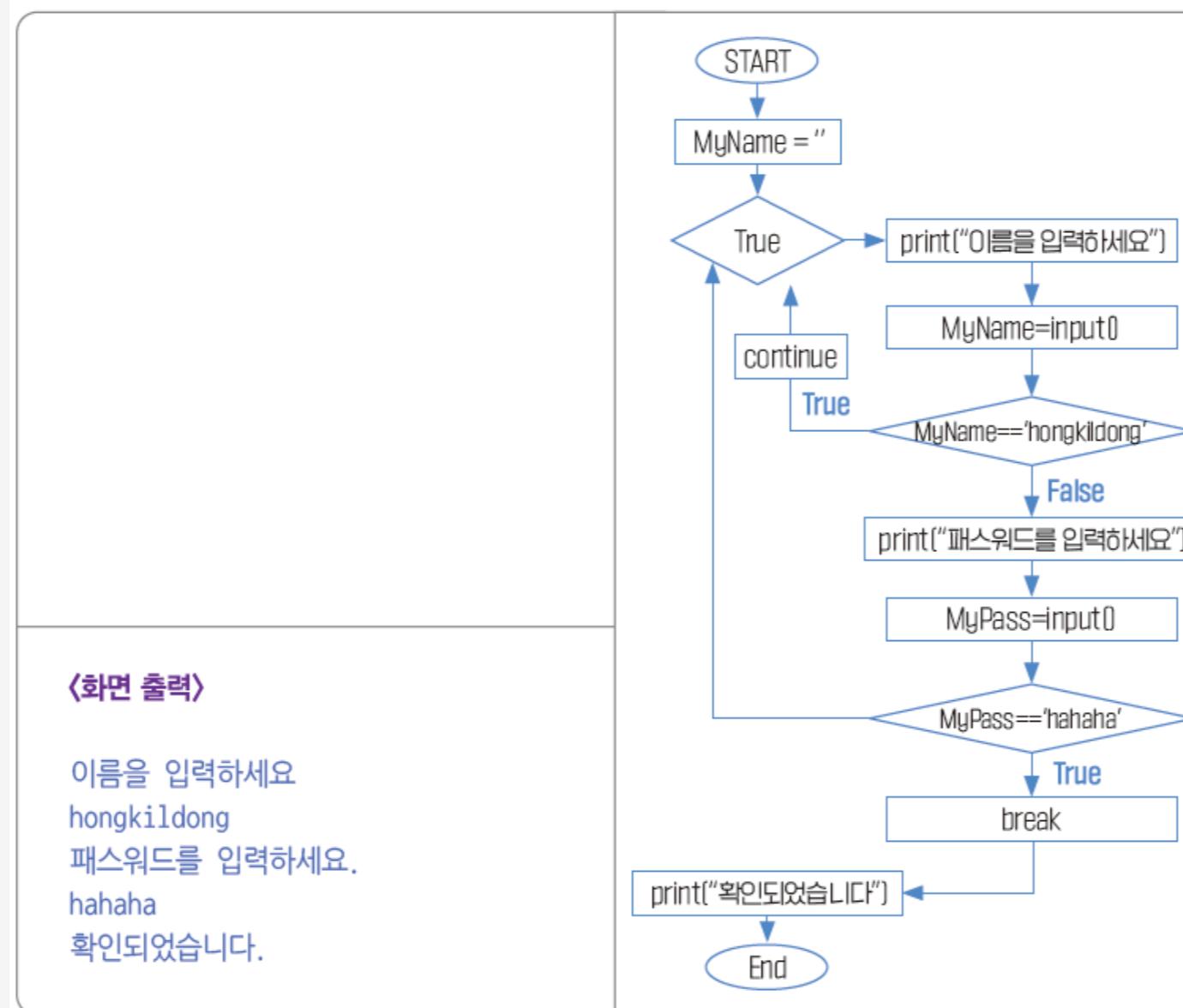
## 7. 연습문제

❖ 아래의 순서도대로 while 문을 사용하여 프로그램을 작성하세요.



## 7. 연습문제

❖ 아래의 순서도대로 while 문을 사용하여 프로그램을 작성하세요.



## 7. 연습문제

- ❖ **for문을 사용하여 1단 구구단을 출력하는 프로그램을 작성해 보자.**



## 7. 연습문제

❖ **for문을 사용하여 1단과 2단 구구단을 출력하는 프로그램을 작성해 보자**



## 7. 연습문제

- ❖ 이중 for문을 사용하여 1단부터 5단까지 구구단을 출력하는 프로그램을 작성해 보자.



## 7. 연습문제

- ❖ 이중 for문을 사용하여 1단부터 5단까지 구구단을 출력하는 프로그램을 작성해 보자.



## 7. 연습문제

- ❖ 이중 **for**문을 사용하여 다음과 같은 출력을 만드는 프로그램을 작성 해보자.

〈화면 출력〉

1  
1 2  
1 2 3  
1 2 3 4  
1 2 3 4 5

〈화면 출력〉

A  
A B  
A B C  
A B C D  
A B C D E

## ❖ 과제

- 1. **for**문 코드 작성하기
- 2. **range** 함수를 인수 개수에 따라 3가지 작성하기
- 3. **while** 문 코드 작성하기
- 4. **continue** 코드 작성하기
- 5. **break** 코드 작성하기
- 6. 이중 **for**문을 사용하여 구구단 작성하기

## ❖ 다음 수업 내용

- **함수**
  - 함수 정의, 호출형식, 매개변수, **return**, 호출순서