

파이썬 기초

06 반복문



홍필두 교수
(파이썬기초)



학습내용

- 01 반복문
- 02 제어문
- 03 다중반복구조





학습목표

- 반복문을 이해하고 사용할 수 있다.
- 반복문 내에서 탈출, 건너뛰기 등 제어문을 이해하고 사용할 수 있다.
- 다중반복문을 이해하고 사용할 수 있다.



사전 학습

“반복처리는 프로그래밍의 기본”

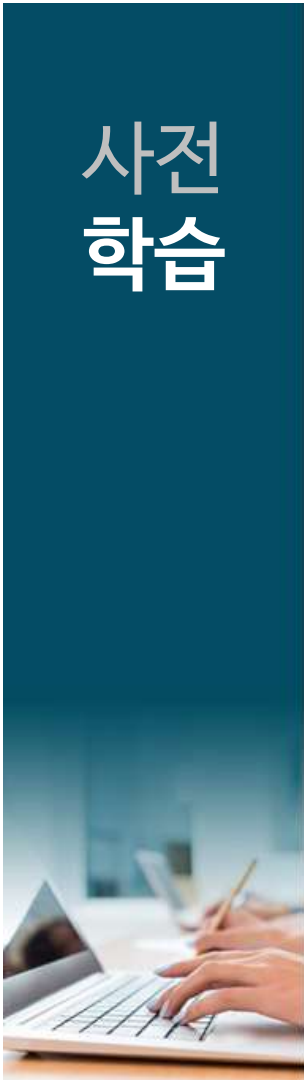
프로그래밍의 주요 요소로 순차처리, 변수, 연산, 비교, 반복임을 배웠습니다. 반복처리와 비교로 주요 로직과 알고리즘을 구성할 수 있습니다.

2의 100승을 구하는 작업은 2에 2를 곱한 결과에 다시 2를 곱하고 하는 일을 반복하는 일입니다.

우리가 대형 마트에서 카트에 물건을 담고 계산대에서 계산한 후 영수증을 받으면 구입항목, 단가, 수량 총액이 구입한 물건만큼 반복되어 나옵니다.

C, java 등 모든 프로그래밍 언어에서 표기하는 형식이 조금 다를 뿐, 모두 **반복문**이 기본적으로 사용됩니다.

여러분은 반복문의 기본적인 사항에 대하여 검색을 통하여 미리 알아보도록 합시다.



01

반복문

- 1) 반복문의 개요
- 2) while 반복문
- 3) for 반복문

1) 반복문의 개요



1, 1+1, 1+1+1, 1+1+1+1, ... 1+1+...+1 (100번 더함)



이러한 계산을 한다면 반복적으로 100번 처리

```
i=0
i=i+1
i=i+1
...
...
i=i+1

(100번의 반복)
```




```
i=0
while j < 100:
    i+=1
```

```
i=0
for j in
range(100):
    i+=1
```

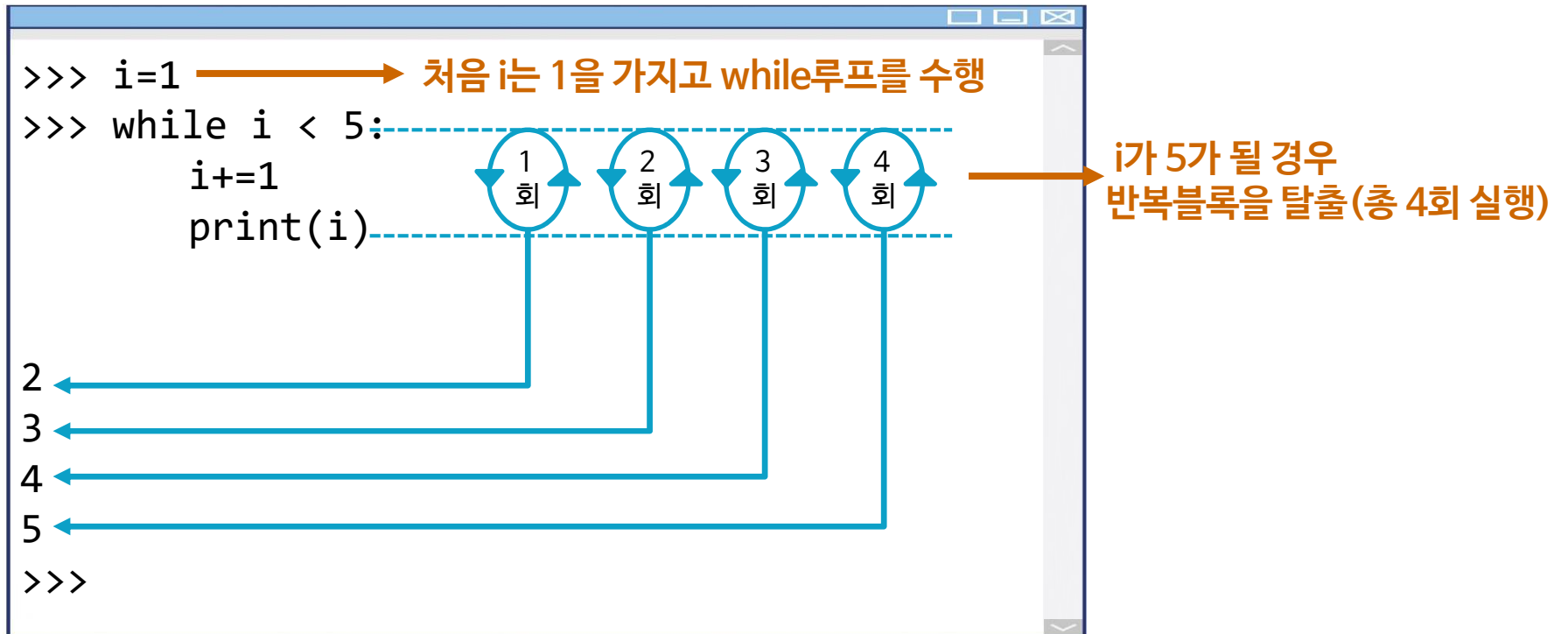
2) while 반복문

“ 명령블록을 해당 조건이
만족할 때까지 실행함 ”



```
while 조건:  
    명령블록
```

2) while 반복문



2) while 반복문

 반복문을 통하여 누적을 구하는 예제



가장 많이 사용하는 패턴임




1부터 100까지의 숫자를 더하기

2) while 반복문

반복문을 통하여 누적을 구하는 예제

```
i=0          #반복 카운터, 초기화
sum=0        #누적할 변수, 초기화
while i <100 : #0부터 100까지 실행, 만일 실행횟수는 100회
    i+=1      #i=0+1, i=1+1, i=2+1, i=3+1, i는 1씩 증가
    sum = sum + i #sum=0+1, sum=1+2, sum=3+3, sum=6+4 ...
                #sum은 더한 값이 누적됨
print("sum=",sum) #while 반복을 빠져 나와서 출력
```



```
>>>
== RESTART: ...==
sum= 5050
>>>
```

2) while 반복문



세 경우는 수행 방법이 다르며 완전히 이해하고 프로그래밍 하여야 함



카운터 증가 후 누적, 누적 후 카운터 증가,
while문 조건의 비교순서에 유의할 것

2) while 반복문

01

```
i=0
sum=0
while i <100 :
    i+=1
    sum = sum + i
print("sum=",sum)
```

```
>>>
== RESTART: ...==
sum= 5050
>>>
```

02

```
i=0
sum=0
while i <100 :
    sum = sum + i
    i+=1
print("sum=",sum)
```

```
>>>
== RESTART: ...==
sum= 4950
>>>
```


03

```
i=0
sum=0
while i <101 :
    sum = sum + i
    i+=1
print("sum=",sum)
```

```
>>>
== RESTART: ...==
sum= 5050
>>>
```

3) for 반복문

“ 명령블록에 대하여 컬렉션을
하나씩 대입하여 실행함 ”



```
for 변수 in 컬렉션:  
    명령블록
```

3) for 반복문



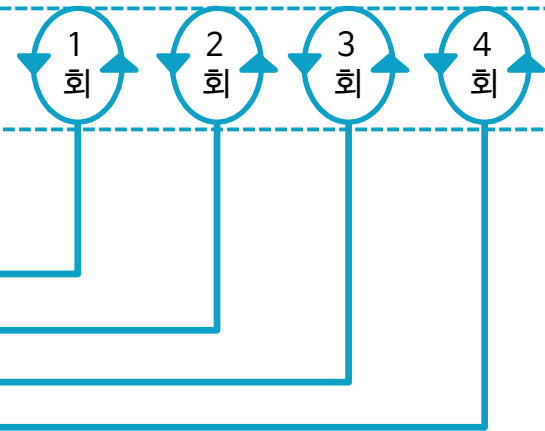
a는 컬렉션 ["1",2,"3",55]을 하나씩 가지고 for루프를 수행



컬렉션 값을 다 수행하면 반복블록을 탈출(총 4회 실행)

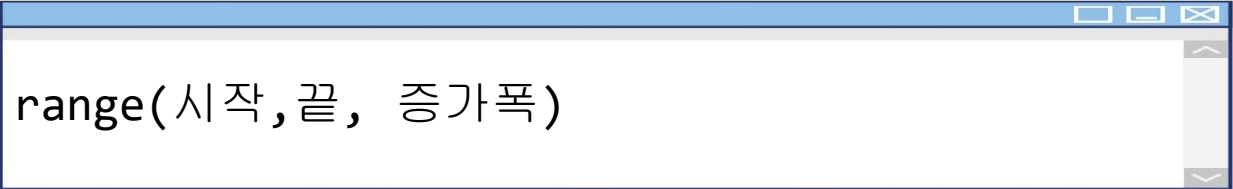
```
>>> i=0
>>> for a in ["1",2,"3",55]:
    print("a[",i,"] 자료형:",type(a)," 값:",a)
    i+=1
```

```
a[ 0 ] 자료형: <class 'str'> 값: 1
a[ 1 ] 자료형: <class 'int'> 값: 2
a[ 2 ] 자료형: <class 'str'> 값: 3
a[ 3 ] 자료형: <class 'int'> 값: 55
>>>
```



3) for 반복문

“ range()를 사용하여 순차적으로
증가(감소)하는 반복문을 수행함 ”



```
range(시작, 끝, 증가폭)
```

3) for 반복문



시작과 증가폭은 각각 생략할 수 있음

↳ 기본값 : 시작=0, 증가폭=1

```
>>> for i in range(1,10,1):  
    print(i, ", ", end="")  
  
1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 ,  
>>>
```

→ 1,2,3,...,9까지
9번 실행함에
주의할 것

3) for 반복문

range 사용 사례

```
>>> for i in range(10):  
        print(i, ", ", end="")
```

```
0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 ,
```

```
>>> for i in range(1,10):  
        print(i, ", ", end="")
```

```
1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 ,
```

3) for 반복문

range 사용 사례

```
>>> for i in range(1,10,2):  
    print(i,",",end="")
```

```
1 ,3 ,5 ,7 ,9 ,
```

```
>>> for i in range(10,0,-1):  
    print(i,",",end="")
```

```
10 ,9 ,8 ,7 ,6 ,5 ,4 ,3 ,2 ,1 ,
```



02

제어문

- 1) 제어문
- 2) 제어문 예제

1) 제어문

제어문장을
사용하는 경우

반복문에서 루프를 탈출하는 경우

해당 문장 아래를 실행하지않고
다시 루프를 실행할 경우

1) 제어문

제어문장을
사용하는 경우

반복문에서 루프를 탈출하는 경우

해당 문장 아래를 실행하지않고
다시 루프를 실행할 경우

```
>>> for i in range(10):  
    if(i==3):  
        break  
    print(i, ", ", end="")  
  
0 ,1 ,2 ,
```

1) 제어문

제어문장을
사용하는 경우

반복문에서 루프를 탈출하는 경우

해당 문장 아래를 실행하지 않고
다시 루프를 실행할 경우

```
>>> for i in range(10):
```

```
    if(i==3):
```

```
        continue
```

```
        print(i, ", ", end="")
```

문장 아래를 실행하지
않고 반복문의 처음으로
이동 후 계속 반복수행

```
0 , 1 , 2 , 4 , 5 , 6 , 7 , 8 , 9 ,
```

2) 제어문 예제

```
>>> i=0;sum=0
>>> while True: → 문장으로 무한루프 실행
    i+=1;sum+=i
    if(i==100): break → break로 탈출

>>> print("i=",i,"sum=",sum)
i= 100 sum= 5050
```

2) 제어문 예제



별을 30씩 찍기, 30번째에는 줄을 바꿔야 함



range(0,120)이 아니고 range(1,121)을 한 이유는 i%30과 연관이 있음

```
>>> for i in range(1,121):
        print("*",end="")
        if(i%30==0): print(end="\n")

*****
*****
*****
*****

>>>
```


03

다중반복구조

- 1) 다중반복구조
- 2) 다중반복구조 예제
- 3) 심화실습

1) 다중반복구조

다중반복구조

반복문안에 반복문이 중첩되는 구조

↳ 첨자(반복문 변수)에 따라 실행과 제어에 유의해야 함

1) 다중반복구조

구구단 예제

```
for i in range(1,10):  
    for j in range(1,10):  
        print( i,"*",j,"=",i*j)  
    print("*****")
```

※ range와 단 변경표시(*****)의 순서에 유의

```
>>>  
== RESTART: ...==  
2 * 2 = 4  
2 * 3 = 6  
2 * 4 = 8  
2 * 5 = 10  
2 * 6 = 12  
2 * 7 = 14  
2 * 8 = 16  
2 * 9 = 18  
*****  
~~~~~  
9 * 8 = 72  
9 * 9 = 81  
*****  
>>>
```

1) 다중반복구조



첨자를 주고 받음에 따라 루프수행이 변동됨



다음의 다중반복구조를 분석하여 어떻게 수행되는지 이해하도록 함

```
for i in range(1,10):  
    for j in range(i,10):  
        print( "*",end="")  
    print(end="\n")
```

```
>>>  
== RESTART: ...==  
*****  
*****  
*****  
*****  
*****  
****  
***  
**  
*  
>>>
```

2) 다중반복구조 예제



앞에서 수행한 for반복문을 while True: 의 무한 반복문으로 고쳐서 작성 할 것

```
for i in range(1,10):  
    for j in range(i,10):  
        print( "*",end="")  
    print(end="\n")
```



```
i=1  
while True:  
    j=i  
    while True:  
        if(j==10): break  
        print( "*",end="")  
        j+=1  
    if(i==10): break  
    print(end="\n")  
    i+=1
```

3) 심화실습

실습 ①

- 다음 결과와 같이 달력을 인쇄할 것(윤달은 없는 것으로 가정)
- 간단히 문제 소개만 하고 풀이 및 설명은 실습강의에서 다룸

1 월	:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,
2 월	:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,
3 월	:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,
4 월	:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,
5 월	:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,
6 월	:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,
7 월	:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,
8 월	:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,
9 월	:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,
10 월	:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,
11 월	:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,
12 월	:1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,

3) 심화실습

실습 ①

- 다음 결과와 같이 달력을 인쇄할 것(윤달은 없는 것으로 가정)
- 간단히 문제 소개만 하고 풀이 및 설명은 실습강의에서 다룸

프로그램 작성 예

```
Mon=1
for LastDay in [31,28,31,30,31,30,31,31,30,31,30,31]:
    print(Mon,"월 :",end="")
    for Day in range(1,LastDay+1):
        print(Day,end=",")
    print(end="\n")
    Mon+=1
```

확 코딩내용 지우려다 참음
파이선의 느낌을 감 잡도록!!

3) 심화실습

실습 ②

- 다음 결과와 같이 구구단을 인쇄할 것
- 간단히 문제 소개만 하고 풀이 및 설명은 실습강의에서 다룸

2 * 2 = 4	3 * 2 = 6	4 * 2 = 8
2 * 3 = 6	3 * 3 = 9	4 * 3 = 12
2 * 4 = 8	3 * 4 = 12	4 * 4 = 16
2 * 5 = 10	3 * 5 = 15	4 * 5 = 20
2 * 6 = 12	3 * 6 = 18	4 * 6 = 24
2 * 7 = 14	3 * 7 = 21	4 * 7 = 28
2 * 8 = 16	3 * 8 = 24	4 * 8 = 32
2 * 9 = 18	3 * 9 = 27	4 * 9 = 36
5 * 2 = 10	6 * 2 = 12	7 * 2 = 14
5 * 3 = 15	6 * 3 = 18	7 * 3 = 21
5 * 4 = 20	6 * 4 = 24	7 * 4 = 28
5 * 5 = 25	6 * 5 = 30	7 * 5 = 35
5 * 6 = 30	6 * 6 = 36	7 * 6 = 42
5 * 7 = 35	6 * 7 = 42	7 * 7 = 49
5 * 8 = 40	6 * 8 = 48	7 * 8 = 56
5 * 9 = 45	6 * 9 = 54	7 * 9 = 63
8 * 2 = 16	9 * 2 = 18	
8 * 3 = 24	9 * 3 = 27	
8 * 4 = 32	9 * 4 = 36	
8 * 5 = 40	9 * 5 = 45	
8 * 6 = 48	9 * 6 = 54	
8 * 7 = 56	9 * 7 = 63	
8 * 8 = 64	9 * 8 = 72	
8 * 9 = 72	9 * 9 = 81	

3) 심화실습

실습 ②

- 다음 결과와 같이 구구단을 인쇄할 것
- 간단히 문제 소개만 하고 풀이 및 설명은 실습강의에서 다룸

프로그램 작성 예

```
for i in range(2,10,3):
    for j in range(2,10):
        print(i,"*",j,"=",i*j,end="\t")
        print(i+1,"*",j,"=",i*j,end="\t")
        if(i+2!=10): print(i+2,"*",j,"=",i*j,end="\n")
        else: print(end="\n")
    print(end="\n")
```

확 코딩내용 지우려다 참음
파이썬의 느낌을 감 잡도록!!

3) 심화실습

실습 ③

- 다음 결과와 같이 구구단을 인쇄할 것(앞의 경우와 인쇄 순서가 다름)
- 간단히 문제 소개만 하고 풀이 및 설명은 실습강의에서 다룸

2 * 2 = 4	5 * 2 = 4	8 * 2 = 4
2 * 3 = 6	5 * 3 = 6	8 * 3 = 6
2 * 4 = 8	5 * 4 = 8	8 * 4 = 8
2 * 5 = 10	5 * 5 = 10	8 * 5 = 10
2 * 6 = 12	5 * 6 = 12	8 * 6 = 12
2 * 7 = 14	5 * 7 = 14	8 * 7 = 14
2 * 8 = 16	5 * 8 = 16	8 * 8 = 16
2 * 9 = 18	5 * 9 = 18	8 * 9 = 18
3 * 2 = 6	6 * 2 = 6	9 * 2 = 6
3 * 3 = 9	6 * 3 = 9	9 * 3 = 9
3 * 4 = 12	6 * 4 = 12	9 * 4 = 12
3 * 5 = 15	6 * 5 = 15	9 * 5 = 15
3 * 6 = 18	6 * 6 = 18	9 * 6 = 18
3 * 7 = 21	6 * 7 = 21	9 * 7 = 21
3 * 8 = 24	6 * 8 = 24	9 * 8 = 24
3 * 9 = 27	6 * 9 = 27	9 * 9 = 27
4 * 2 = 8	7 * 2 = 8	
4 * 3 = 12	7 * 3 = 12	
4 * 4 = 16	7 * 4 = 16	
4 * 5 = 20	7 * 5 = 20	
4 * 6 = 24	7 * 6 = 24	
4 * 7 = 28	7 * 7 = 28	
4 * 8 = 32	7 * 8 = 32	
4 * 9 = 36	7 * 9 = 36	

3) 심화실습

실습 ③

- 다음 결과와 같이 구구단을 인쇄할 것(앞의 경우와 인쇄 순서가 다름)
- 간단히 문제 소개만 하고 풀이 및 설명은 실습강의에서 다룸

프로그램 작성 예

```
for i in range(2,5):
    for j in range(2,10):
        print(i,"*",j,"=",i*j,end="\t")
        print(i+3,"*",j,"=",i*j,end="\t")
        if(i+6!=10): print(i+6,"*",j,"=",i*j,end="\n")
        else: print(end="\n")
    print(end="\n")
```

확 코딩내용 지우려다 참음
파이선의 느낌을 감 잡도록!!

04

실습하기 I

- 1) 반복문 - while 반복문
- 2) 반복문 - for 반복문
- 3) 제어문 - 제어문 예제

실습내용

- 1) 반복문 - while 반복문
- 2) 반복문 - for 반복문
- 3) 제어문 - 제어문 예제

05

실습하기 II

- 1) 다중반복구조 - 다중반복구조 예제
- 2) 다중반복구조 - 심화실습1
- 3) 다중반복구조 - 심화실습2, 3

실습내용

- 1) 다중반복구조 - 다중반복구조 예제
- 2) 다중반복구조 - 심화실습1
- 3) 다중반복구조 - 심화실습2, 3



학습활동

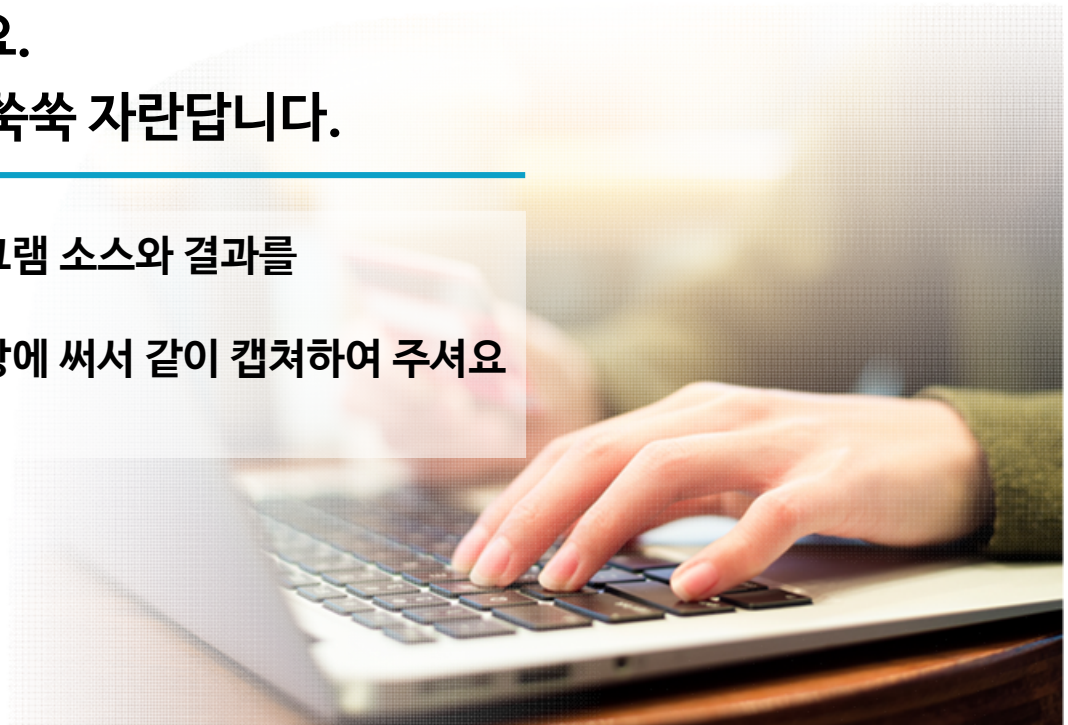
일시정지 버튼을 누른 후, 아래의 학습활동에 참여하세요.

Q

오늘 배운 내용을 스스로 실습하여
자유게시판에 올려 주세요.

이렇게 정리하면 실력이 쑥쑥 자란답니다.

- ① 본인이 실습한 내용을 프로그램 소스와 결과를 캡처하여 올려주세요
- ② 본인의 학번과 이름을 메모장에 써서 같이 캡처하여 주세요
- ③ 그리고 설명도 달아 주세요





학습활동에 대한 교수님 의견

Q

오늘 배운 내용을 스스로 실습하여 자유게시판에 올려 주세요.
이렇게 정리하면 실력이 쑥쑥 자란답니다.

A

[오늘 학습한 내용의 실습 사항]

- ① 반복문 - while 반복문
- ② 반복문 - for 반복문
- ③ 제어문 - 제어문 예제
- ④ 다중반복구조 - 다중반복구조 예제
- ⑤ 다중반복구조 - 심화실습1
- ⑥ 다중반복구조 - 심화실습2, 3

학습 평가

Q1

Q2

Q3

Q4

Q5

Q1

다음 while i < 5 : print("a") 문장에서
a라고 인쇄되는 경우는?

- 1 a="5"가 들어 있는 경우
- 2 i=5가 들어 있는 경우
- 3 i=4가 들어 있는 경우
- 4 a="4"가 들어 있는 경우



학습 평가

Q1

Q2

Q3

Q4

Q5

Q1

다음 while i < 5 : print("a") 문장에서
a라고 인쇄되는 경우는?

- 1 a="5"가 들어 있는 경우
- 2 i=5가 들어 있는 경우
- ☒ i=4가 들어 있는 경우
- 4 a="4"가 들어 있는 경우

정답

3번

해설

조건 i < 5 가 참인 경우 print("a")를 실행합니다.

학습 평가

Q1

Q2

Q3

Q4

Q5

Q2

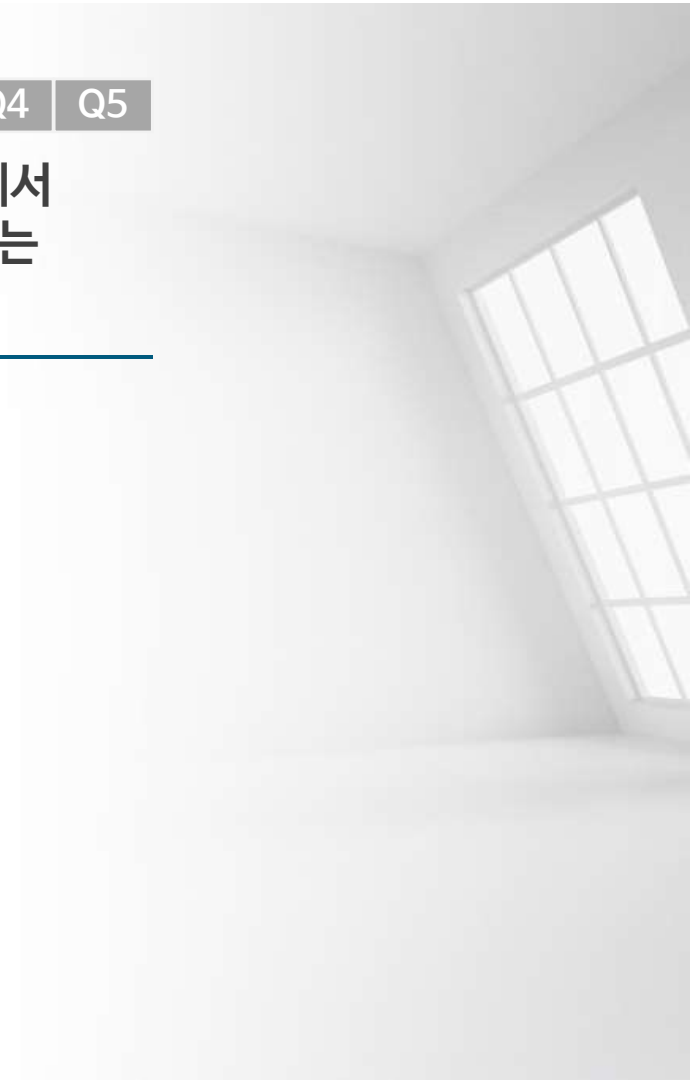
다음 `for i in range(1,10):`의 표현에서
`for`문 블록 안에서 `i`값으로 가질 수 있는
것은 ?

1 0

2 1

3 10

4 11



학습 평가

Q1

Q2

Q3

Q4

Q5

Q2

다음 `for i in range(1,10):`의 표현에서
`for`문 블록 안에서 `i`값으로 가질 수 있는
것은 ?

1 0

☒ 1

3 10

4 11

정답

2번

해설

`i`는 1, 2, 3, 4, 5, 6, 7, 8, 9의 값을 같습니다.

학습 평가

Q1

Q2

Q3

Q4

Q5

Q3

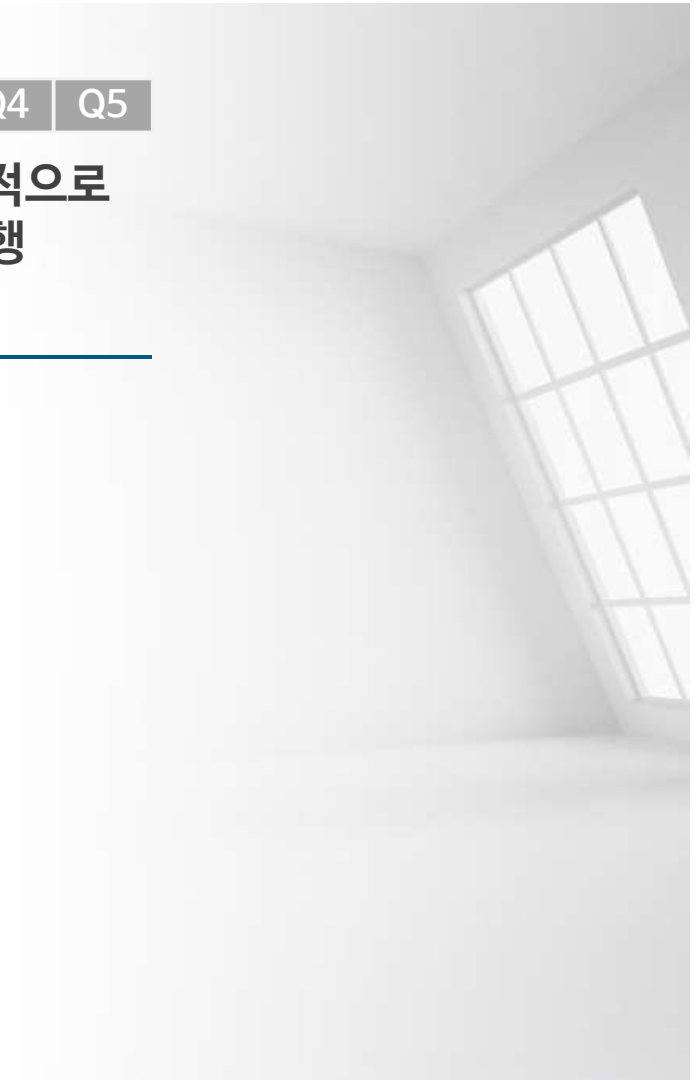
다음 `for i in (1,9):`의 표현에서 정상적으로 작동하는 경우 `for`문 블록은 몇 번 수행되는가?

1 1

2 2

3 8

4 9



학습 평가

Q1

Q2

Q3

Q4

Q5

Q3

다음 `for i in (1,9):`의 표현에서 정상적으로 작동하는 경우 `for`문 블록은 몇 번 수행되는가?

1 1

☒ 2

3 8

4 9

정답

2번

해설

만일 `for l in range(1,9)`이면
1,2,3,4,5,6,7,8로 8회 수행하나 `range`가
없으면 해당 (1,9)를 콜렉션으로 인식하여 1,9로
2회 수행합니다.

학습 평가

Q1

Q2

Q3

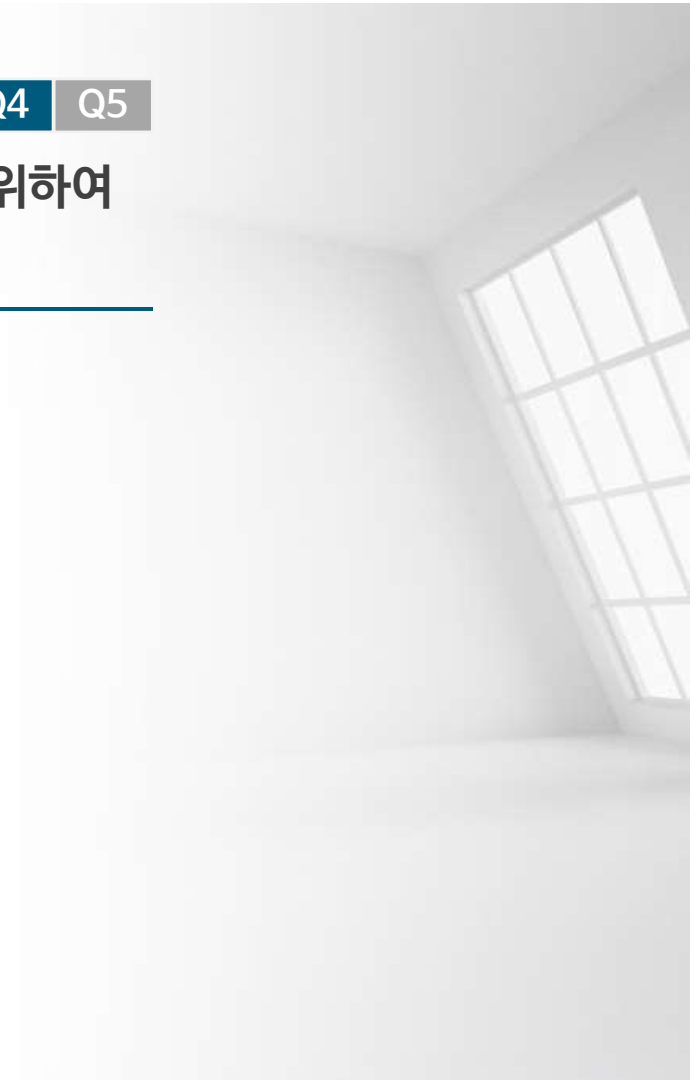
Q4

Q5

Q4

다음중 반복문을 강제로 빠져나오기 위하여 실행하는 문장은?

- 1 exit
- 2 goto
- 3 continue
- 4 break



학습 평가

Q1

Q2

Q3

Q4

Q5

Q4

다음중 반복문을 강제로 빠져나오기 위하여 실행하는 문장은?

1 exit

2 goto

3 continue

☒ break

정답

4번

해설

제어문 중 break문장입니다.

학습 평가

Q1

Q2

Q3

Q4

Q5

Q5

다음 for i in range(10): for j in range(5): 문장을 만났을 때 for j문장 블록의 수행 횟수는?

1 13회

2 15회

3 36회

4 50회



학습 평가

Q1 Q2 Q3 Q4 Q5

Q5

다음 for i in range(10): for j in range(5): 문장을 만났을 때 for j문장 블록의 수행 횟수는?

1 13회

2 15회

3 36회

☒ 50회

정답

4번

해설

다중반복구조로 총 50회 수행됩니다.



정리하기

반복문

- ✓ while 조건
 - 명령블록의 while조건문은 명령 블록을 해당 조건이 만족할 때까지 실행함
- ✓ for 변수 in 컬렉션
 - 명령블록의 for조건문은 명령블록에 대하여 컬렉션을 하나씩 대입하여 실행함
- ✓ for문에서 컬렉션 대신 range()를 사용하여 순차적 증가(감소)하는 반복문을 수행함





정리하기

제어문

- ✓ break 제어문은 해당 루프를 탈출함
- ✓ continue 제어문은 문장 아래를 실행하지 않고 반복문의 처음으로 이동 후 계속 반복수행함

다중반복구조

- ✓ 반복문안에 반복문이 중첩되는 구조를 다중반복구조라 함
- ✓ 첨자(반복문 변수)에 따라 실행과 제어에 유의하여야 함

