

파이썬 프로그래밍

6-1: 반복문

2022.4.5



인하대학교
INHA UNIVERSITY

산업경영공학과 임세실

Overview

- Python의 반복문 이해하기
- While문 이해하기
- For문 이해하기

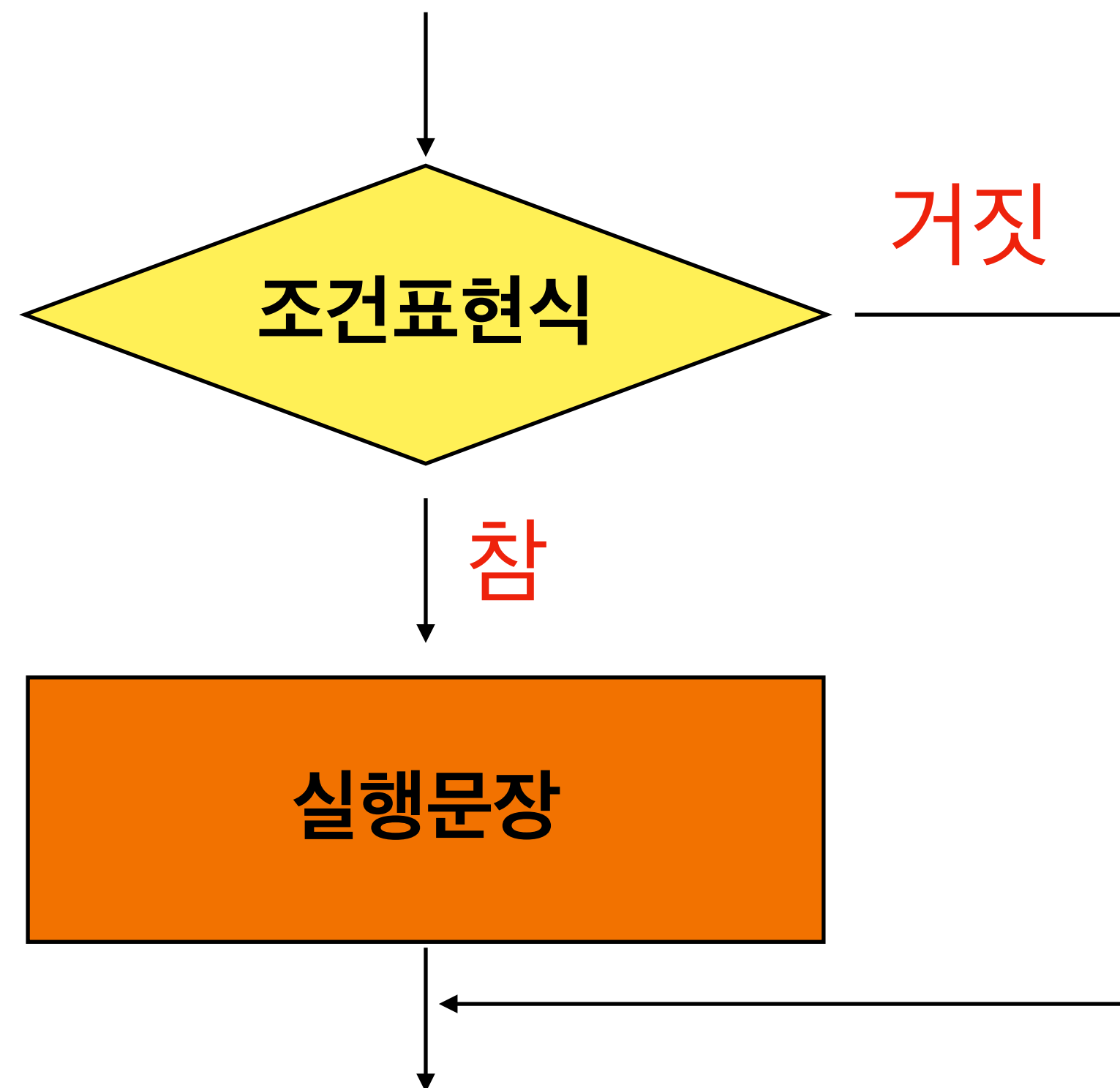


조건문 복습하기



복습: If 조건문이란?

- 특정 조건일때만 코드를 실행하고 싶을 경우 사용



-> 분기: 조건에 따라 실행 흐름을 바꿈

복습: 비교 연산자

- 두값을 비교할 때 사용하며 그 연산 결과: True (참) 혹은 False (거짓)

연산자	설명
<code>==</code>	같음
<code>!=</code>	같지 않음
<code><</code>	작음
<code><=</code>	작거나 같음
<code>></code>	큼
<code>>=</code>	크거나 같음

복습: 논리 연산자

- 불 데이터형 변수를 다루는 연산자

연산자	의미	설명
not	아니다 (논리부정)	불 데이터 값을 반대로 전환
and	그리고 (논리곱)	연산자가 적용된 불 데이터 값이 모두 참일 경우에만 True, 이외에는 False 출력
or	또는 (논리합)	연산자가 적용된 불 데이터 값 중 하나라도 참이면 True, 둘 다 거짓일 경우에만 False 출력



Python의 반복문 이해하기

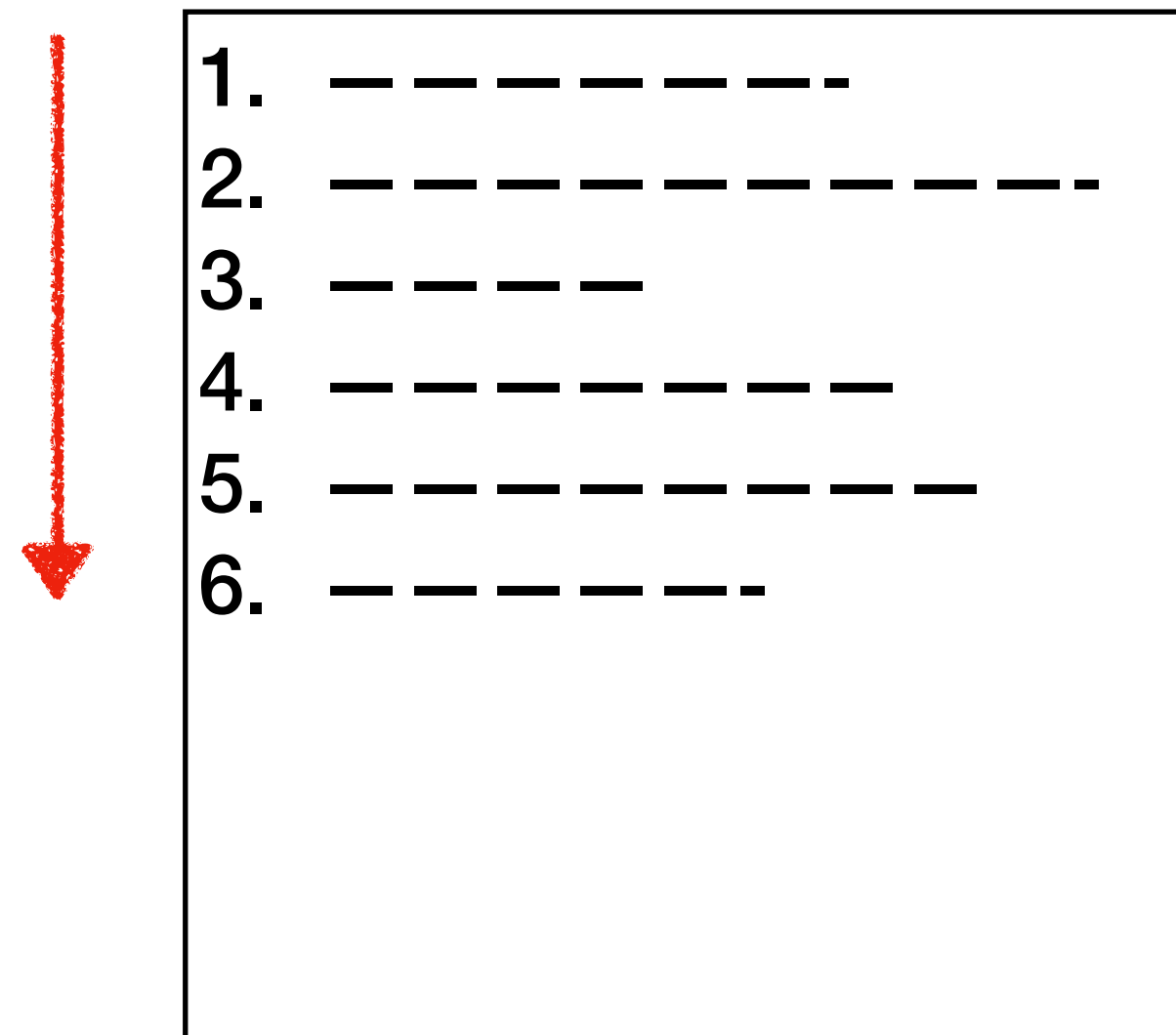
- 반복문이란?
- 파이썬에서 반복문의 종류



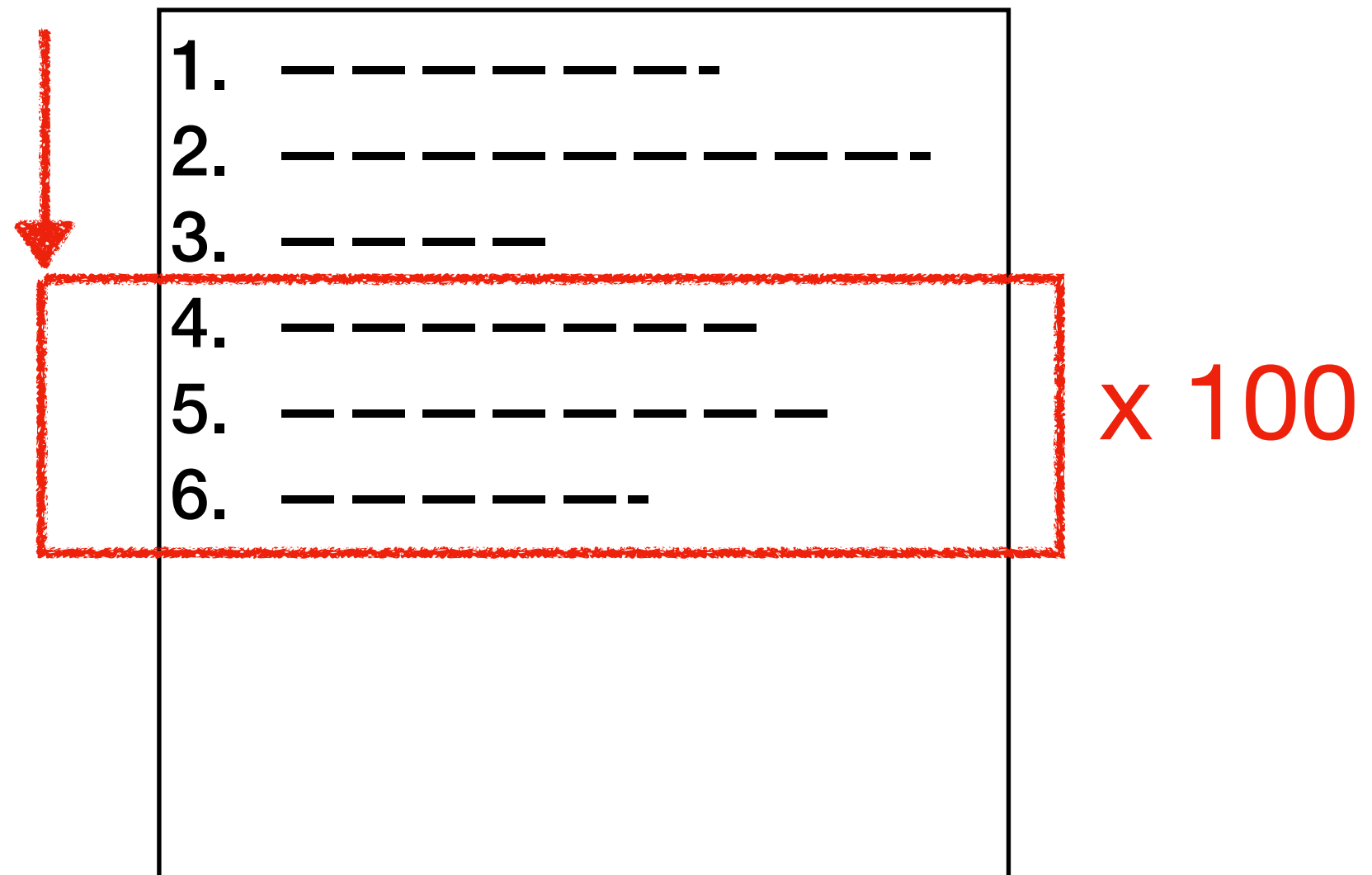
반복문 (loop)이란?

- 컴퓨터가 특정 코드를 반복적으로 수행할 수 있도록 하는 문장

일반적 형식



반복문



파이썬에서의 반복문 (loop)의 종류

- while과 for 반복문 두 가지 종류가 존재
 - while 반복문: 조건문이 참인 동안 실행 문장 반복
 - For 반복문: 순서열의 처음부터 끝까지 도달할 동안 실행 문장 반복



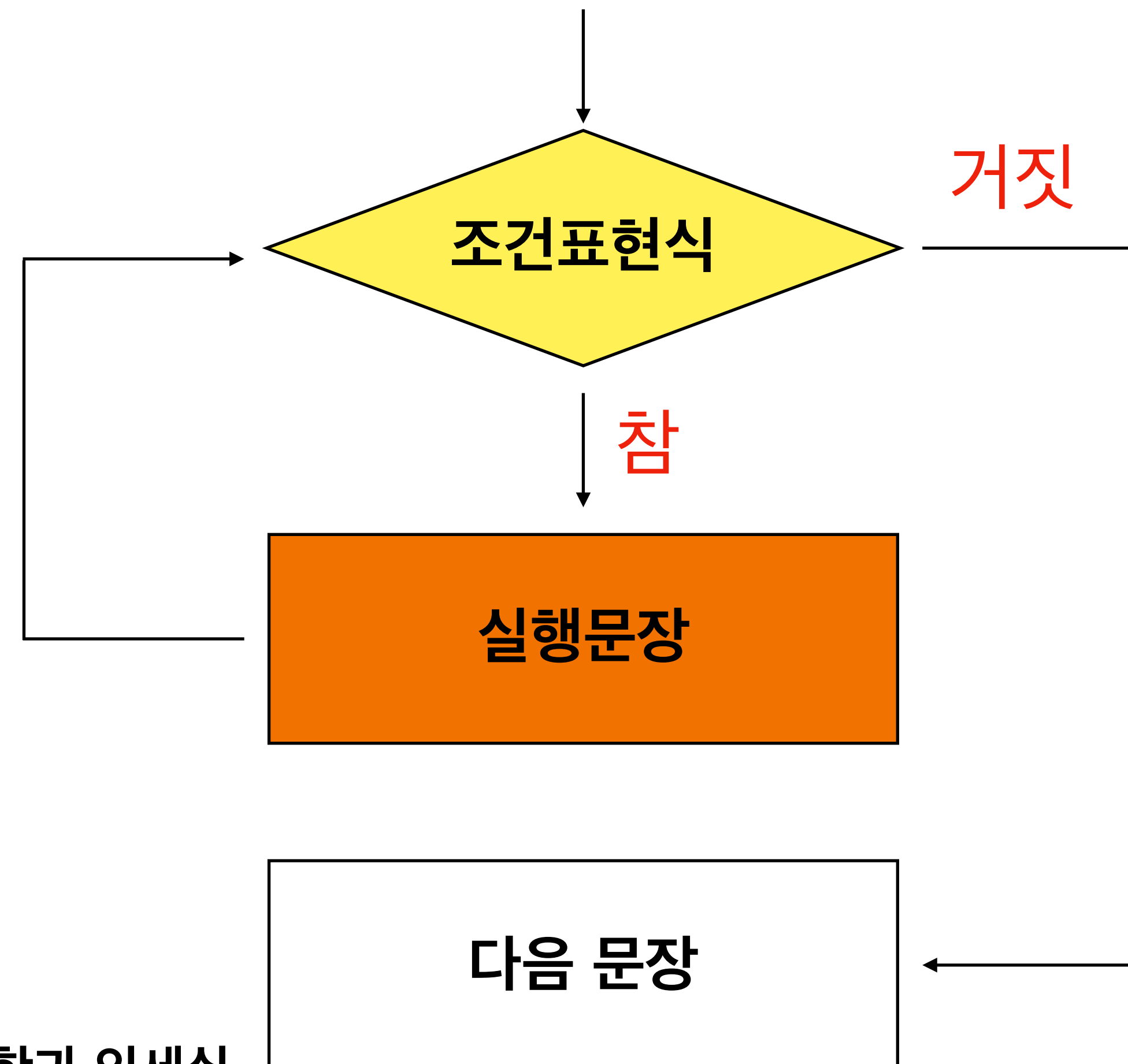
while문 이해하기

- while문이란?
- while문의 기본적 구조
- break으로 while문 탈출하기
- continue로 다음 반복으로 건너뛰기



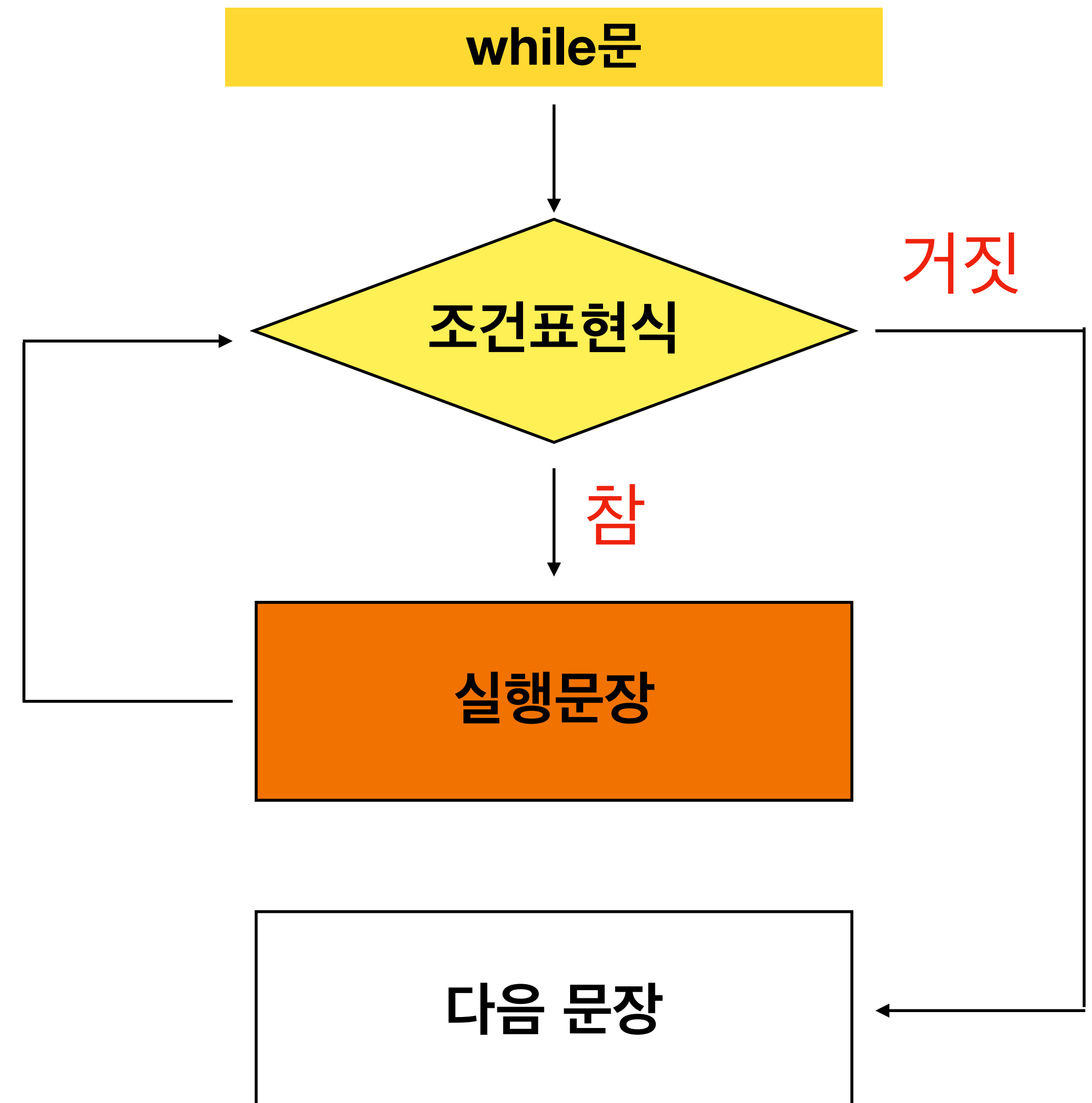
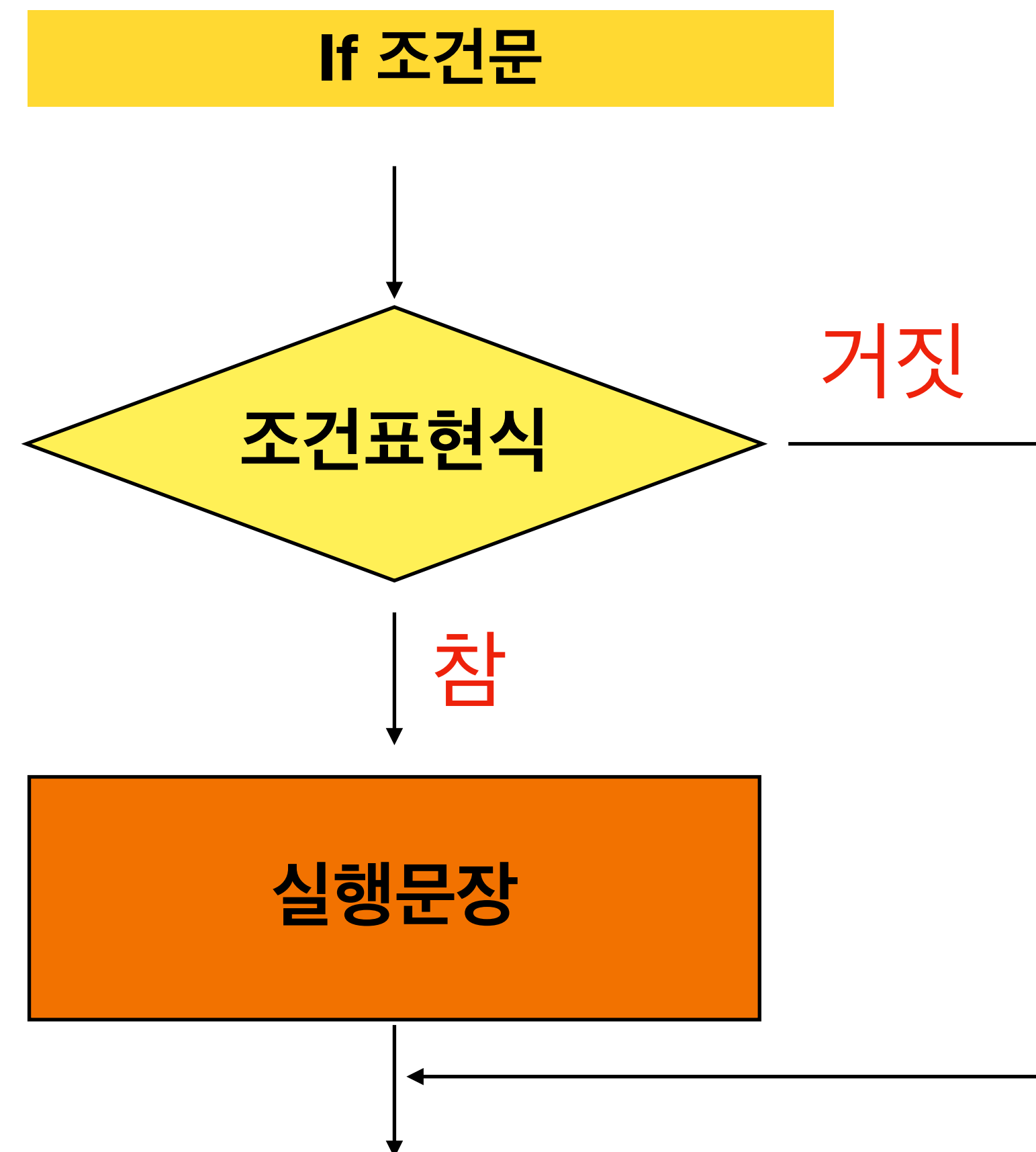
while 문이란?

- 특정 조건이 만족될 때까지 컴퓨터가 반복적인 계산을 수행하도록 하는 문장



If 조건문과 while 문의 차이

- If 조건문은 조건 만족 시 한번 시행인 반면 while문은 반복 시행



while 문의 기본적인 구조

while 조건표현식:

실행문장들

....

- 조건표현식
 - 비교연산자와 논리연산자로 구성된 표현식
 - bool 값을 식의 연산결과로 가짐
 - 표현식 뒤에 반드시 콜론(:)을 붙임
- 실행문장
 - 조건표현식이 참인 동안 실행
 - 실행문장 앞에 반드시 space로 4칸 띄어줌
 - 실행문장은 여러개가 올 수 있음



예제 1:

- 반복할 횟수 입력받아 그만큼“Hello world!” 출력하기

```
count = int(input("반복할 횟수를 입력하세요:"))
print("반복할 횟수: {}번".format(count))
i=0
while i<count:
    print("{}번째 반복: Hello world!".format(i+1))
    i += 1
```

[7] ✓ 3.4s Python

... 반복할 횟수: 5번
1번째 반복: Hello world!
2번째 반복: Hello world!
3번째 반복: Hello world!
4번째 반복: Hello world!
5번째 반복: Hello world!



예제 1-1:

- 반복할 횟수 입력받아 그만큼“Hello world!” 출력하기

```
count = int(input("반복할 횟수를 입력하세요:"))
print("반복할 횟수: {}번".format(count))
i=0
while count > 0:
    i +=1
    print("{}번째 반복: Hello world!".format(i))
    count -= 1
```

[8] ✓ 1.8s Python

... 반복할 횟수: 5번
1번째 반복: Hello world!
2번째 반복: Hello world!
3번째 반복: Hello world!
4번째 반복: Hello world!
5번째 반복: Hello world!



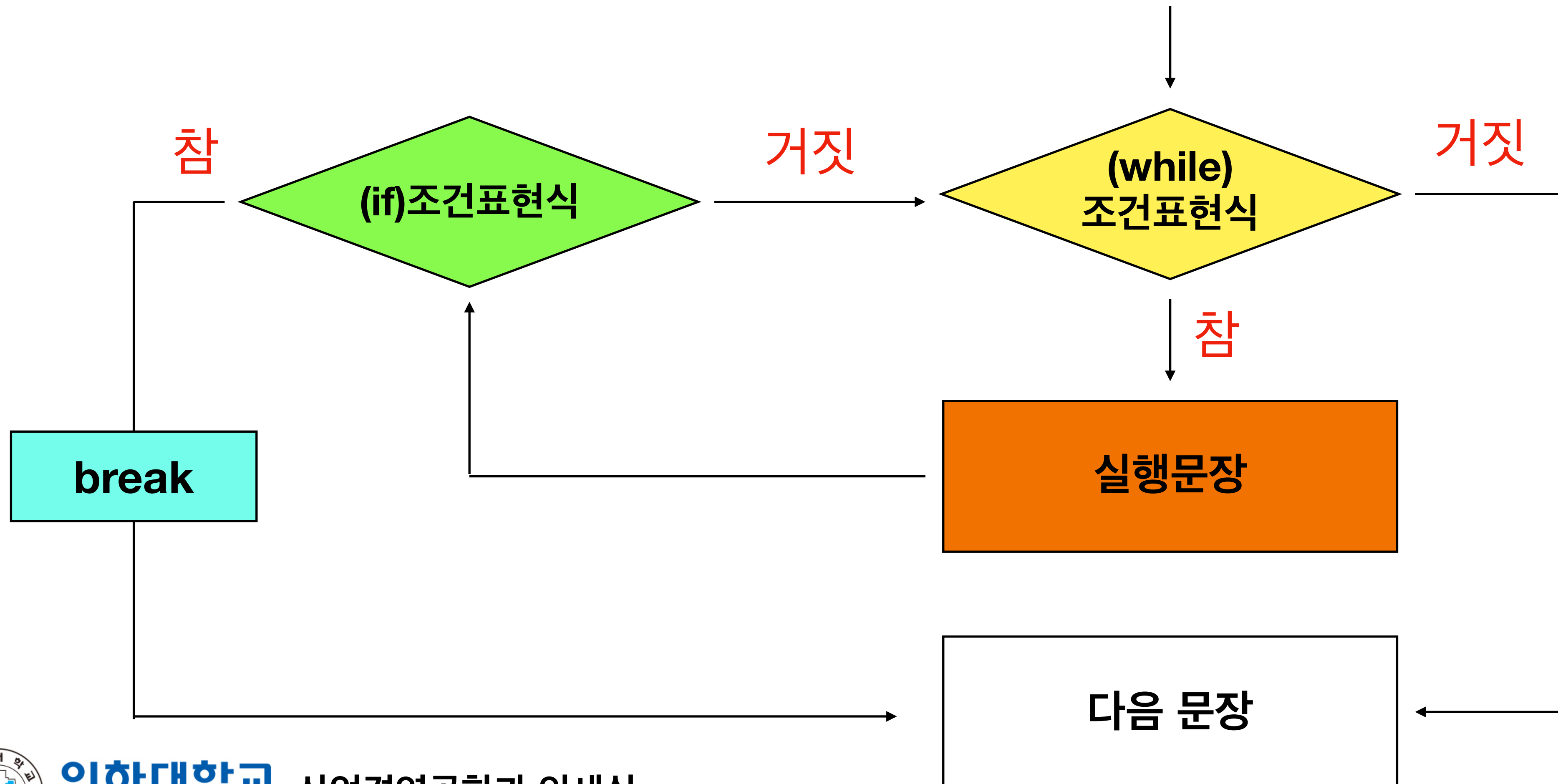
예제 2: (1)

- 커피 자판기 판매 프로그램을 구현하세요. 즉, 자판기 안에 커피가 충분히 있을 때에는 동전을 넣으면 커피가 나옵니다. 하지만 자판기에 커피가 떨어졌다면 판매를 중단하고 판매를 중단한다는 문구를 사용자에게 보여줍니다. 단, 이 자판기는 커피의 값이 정해져있지 않고 단돈 1원이라도 지불하고 싶은 만큼 돈을 내기만 하면 커피를 준다고 가정합니다.



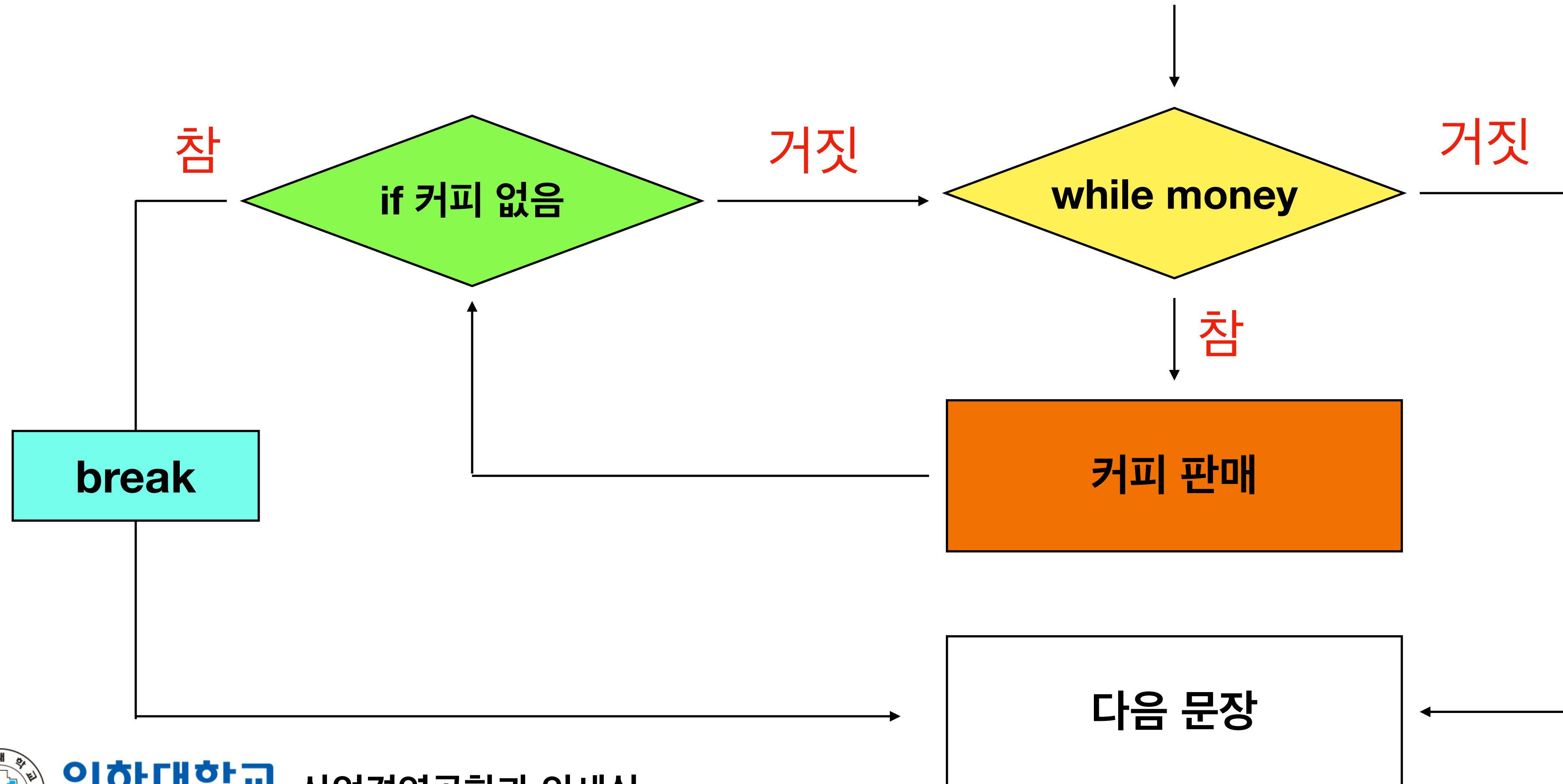
break으로 while문 탈출하기

- 보통 while문 내에서 if 조건문의 실행문으로 나타남



예제 2: (2)

- 보통 while문 내에서 if 조건문의 실행문으로 나타남



예제 2: (3)

```
coffee = 3
money = int(input("돈을 넣어주세요:"))
while money:
    print("맛있게 드세요")
    coffee -= 1
    print("남은 커피의 양은 {}입니다".format(coffee))
    if coffee == 0:
        print("커피가 다 떨어졌습니다. 판매를 중지합니다.")
        break
```

[13] ✓ 2.1s Python

... 맛있게 드세요
남은 커피의 양은 2입니다
맛있게 드세요
남은 커피의 양은 1입니다
맛있게 드세요
남은 커피의 양은 0입니다
커피가 다 떨어졌습니다. 판매를 중지합니다.



continue로 다음 반복으로 건너뛰기

- continue 만나면 현재 반복 중단하고 조건실행문으로 돌아가기

while 조건표현식:

실행문장1

continue

실행문장2



예제 3:

- 1부터 10까지의 숫자 중 짝수만 출력하기

```
count=0
while count<10:
    count +=1
    if count % 2 != 0 : continue
    print(count)
```

[20] ✓ 0.5s Python

... 2
4
6
8
10



for문 이해하기

- for문이란?
- 리스트란?
- 리스트의 연산자 및 함수
- for문의 기본적 구조
- for문과 continue
- for문에서 range 함수 사용해서 숫자 리스트 만들기



for 문이란?

- 반복 변수가 순서열의 처음부터 끝까지 차례로 대입될 동안 실행문장 반복
- 순서열(sequence): 자료가 어떤 순서대로 배열된 것
- 순서열의 예시: 문자열, 리스트, 튜플, 딕셔너리 등

	str1[0]	str1[1]	str1[2]	str1[3]	str1[4]
str1	안	녕	하	세	요

리스트(list)란?

- 자료의 목록을 저장할 수 있는 데이터형
- 숫자, 문자열, bool 등 모두 저장 가능
- 대괄호 [] 내부에 개별 데이터 (요소; element)를 쉼표로 구분해 넣고 정의
 - 예: list1=[1, 2, 3, 4, 5]
list2=["고양이", "개", "토끼", "쥐"]
list3=[253, "안녕", 462, True, "밤"]



리스트(list)의 연산자 및 함수

- 인덱싱: list1[3], list1[-2]
- 슬라이싱: list1[2:4]
- + 연산자: list1의 자료와 list2의 자료 연결해 새로운 list 만듦
- * 연산자: list1의 자료를 * 뒤의 숫자만큼 반복해 새로운 list 만듦
- in 연산자: list 내부에 값 있으면 True, 없으면 False
- len(list): list 내부의 요소의 개수 세서 반환



for 문의 기본적인 구조

for 변수 in 순서열:

실행문장들

....

- 순서열
 - 리스트, 튜플, 문자열 등
 - 변수가 순서열의 각 요소를 순회하면서 대입
- 실행문장
 - 변수가 순서열 순회할 동안 실행
 - 실행문장 앞에 반드시 space로 4칸 띄어줌
 - 실행문장은 여러개가 올 수 있음



예제 4 (1):

- 리스트 내의 모든 요소값 출력하기

```
▷ list1 = [2,5,7,3,2]
  for i in list1:
    print(i)

[21] ✓ 0.4s Python

... 2
    5
    7
    3
    2
```



예제 4 (2):

- 리스트 내의 모든 요소값 출력하기

```
▶ list1 = ["Twee", "Vijf", "Zeven", "Drie", "Twee"]  
  for i in list1:  
    print(i)  
[23] ✓ 0.6s Python  
... Twee  
    Vijf  
    Zeven  
    Drie  
    Twee
```



예제 5:

- 5명의 학생이 시험을 보았다. 60점이 넘으면 합격일 때, 각 학생에 대해 합격인지 불합격인지 결과를 알려주는 프로그램을 작성하라.

```
▶ score = [40,79,85,97,59]

number = 0
for i in score:
    number +=1
    if i >= 60:
        print("{}번 학생-{}점: 합격입니다".format(number,i))
    else:
        print("{}번 학생-{}점: 불합격입니다".format(number,i))
```

[26] ✓ 0.1s Python

... 1번 학생-40점: 불합격입니다
2번 학생-79점: 합격입니다
3번 학생-85점: 합격입니다
4번 학생-97점: 합격입니다
5번 학생-59점: 불합격입니다



for문과 continue

- continue 만나면 현재 반복 중단하고 for문의 조건실행문으로 돌아가기

for 변수 in 리스트:

실행문장1

continue

실행문장2



예제 5-1:

- 5명의 학생이 시험을 보았음. 60점이 넘으면 합격일 때, 합격인 학생에게만 축하메시지를 보내는 프로그램을 작성하라.

```
score = [40,79,85,97,59]

number = 0
for i in score:
    number +=1
    if i < 60:
        continue
    print("{}번 학생-{}점: 축하합니다, 합격입니다!".format(number,i))
```

[28] ✓ 0.4s Python

... 2번 학생-79점: 축하합니다, 합격입니다!
3번 학생-85점: 축하합니다, 합격입니다!
4번 학생-97점: 축하합니다, 합격입니다!



for문에서 range 함수 이용해 숫자 리스트 만들기 (1)

- range() 함수를 이용하면 숫자 리스트를 만들 수 있음
 - range(b): 0부터 b-1까지 정수 리스트 만들기 (숫자 간격: 1)
 - range(a,b): a부터 b-1까지 정수 리스트 만들기 (숫자 간격: 1)
 - range(a,b,c): a부터 b-1까지 숫자간 간격이 c인 정수 리스트 만들기



for문에서 range 함수 이용해 숫자 리스트 만들기 (2)

- list(range())를 사용하면 range 함수로 정의된 값 확인 가능

```
▶ print(list(range(4)))  
print(list(range(1,4)))  
print(list(range(1,4,2)))  
[31] ✓ 0.7s Python  
... [0, 1, 2, 3]  
      [1, 2, 3]  
      [1, 3]
```



for문에서 range 함수 이용해 숫자 리스트 만들기 (3)

- 반복할 횟수 입력받아 그만큼 "Hello world!" 출력하기

```
count=int(input("반복할 횟수를 입력하세요:"))
print("반복할 횟수: {}번".format(count))

for i in range(count):
    print("{}번째 반복: Hello world!".format(i+1))
```

[32] ✓ 2.4s Python

... 반복할 횟수: 5번
1번째 반복: Hello world!
2번째 반복: Hello world!
3번째 반복: Hello world!
4번째 반복: Hello world!
5번째 반복: Hello world!



예제 5-2:

- 5명의 학생이 시험을 보았음. 60점이 넘으면 합격일 때, 합격인 학생에게만 축하메시지를 보내는 프로그램을 작성하라.

```
▶ score = [40,79,85,97,59]

for i in range(len(score)):
    if score[i] < 60:
        continue
    print("{}번 학생-{}점: 축하합니다, 합격입니다!".format(i,score[i]))
```

[33] ✓ 0.9s Python

... 1번 학생-79점: 축하합니다, 합격입니다!
2번 학생-85점: 축하합니다, 합격입니다!
3번 학생-97점: 축하합니다, 합격입니다!



예제 6:

- 구구단 출력하기

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

[37] ✓ 0.6s Python

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



실습 (4/7)

- 강의실에서 출석 확인 예정
 - 온라인 출석확인 대상자는 과제 제출 여부로 출석 확인
- 실습과제는 실습시간 정시에 iClass에 업로드될 예정
- 4월 8일 자정까지 제출해야 과제 점수 인정

