

>>>>>>>>>> 제어 구조의 설계 원리를 중심으로 배우는 >>>>>>>>>>

프로그래밍의 정석

파이썬

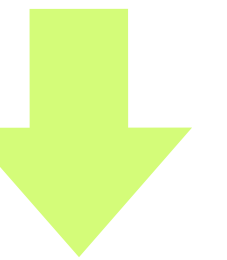
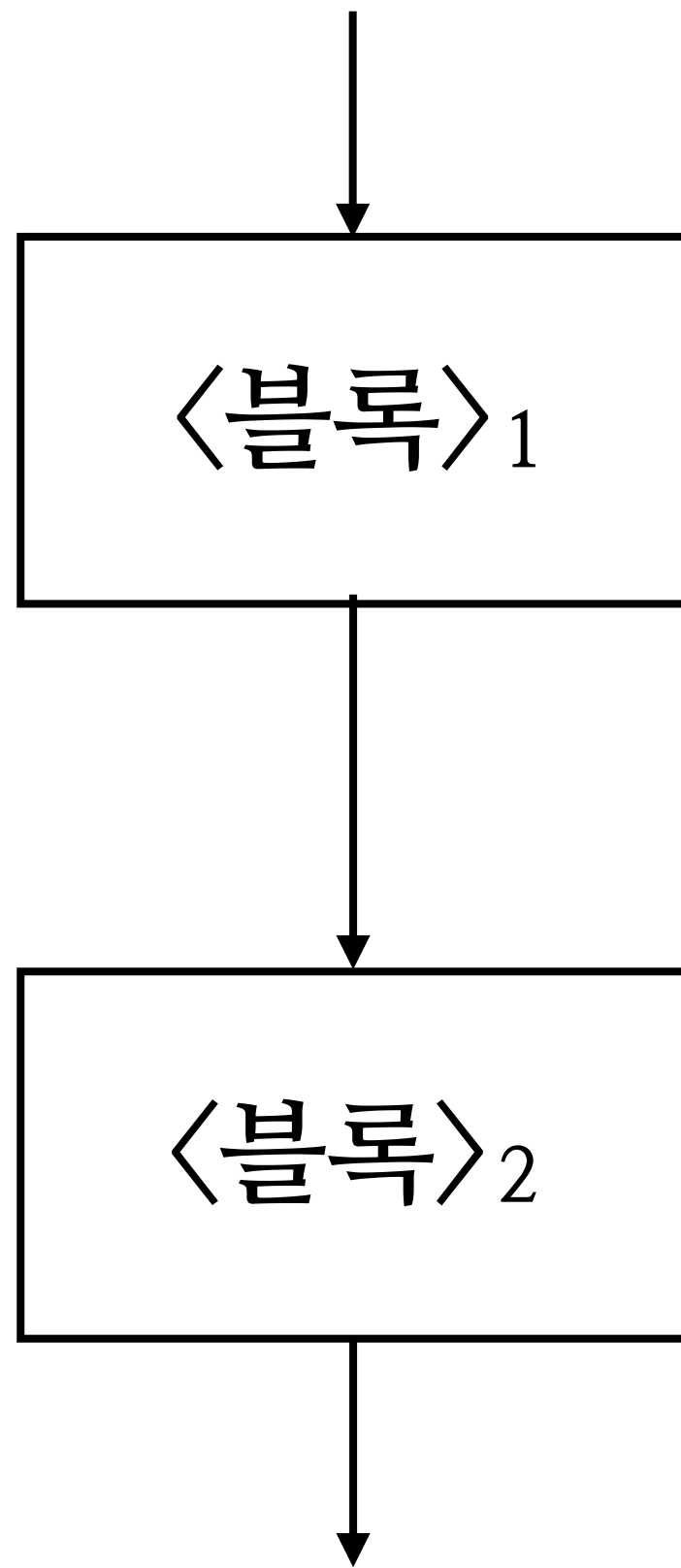
도경구 지음



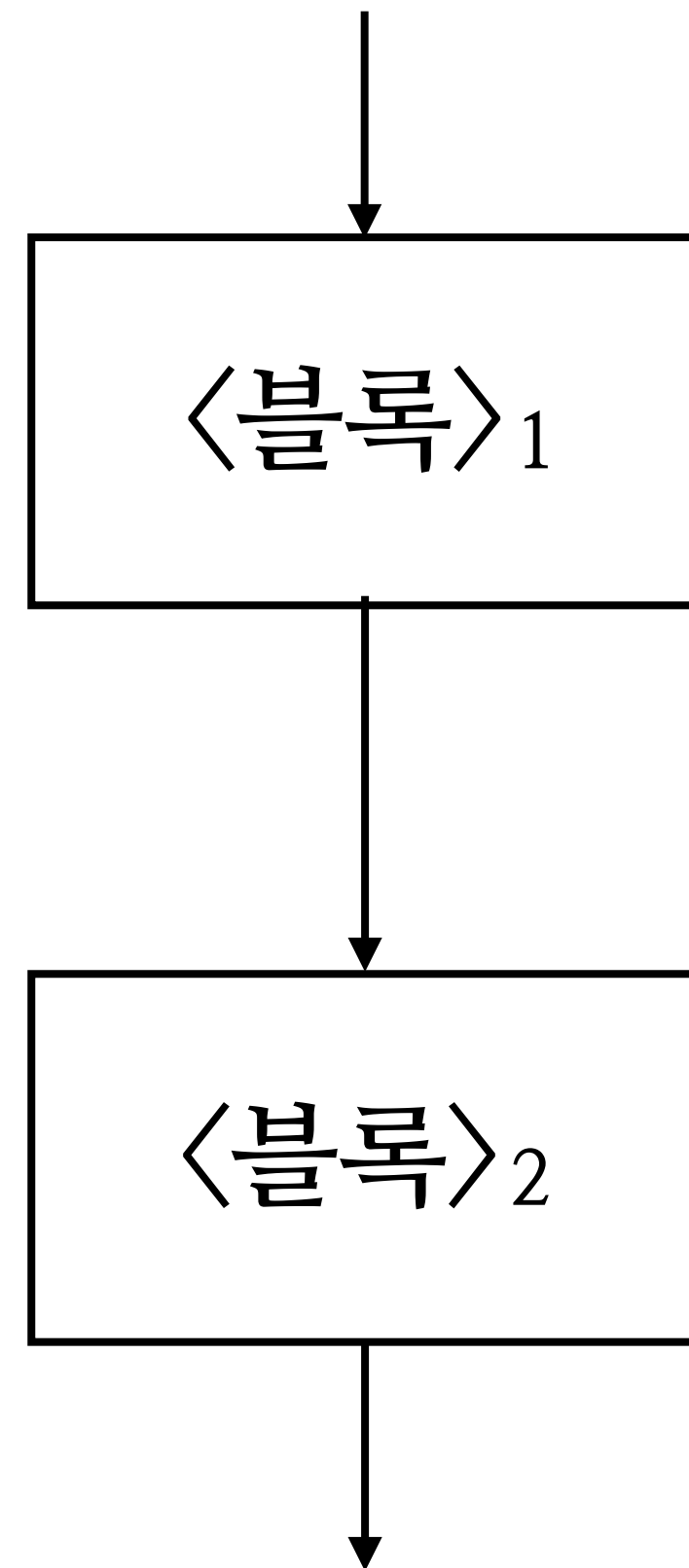
CHAPTER 3

제어 구조

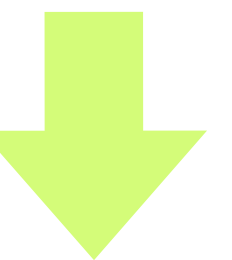
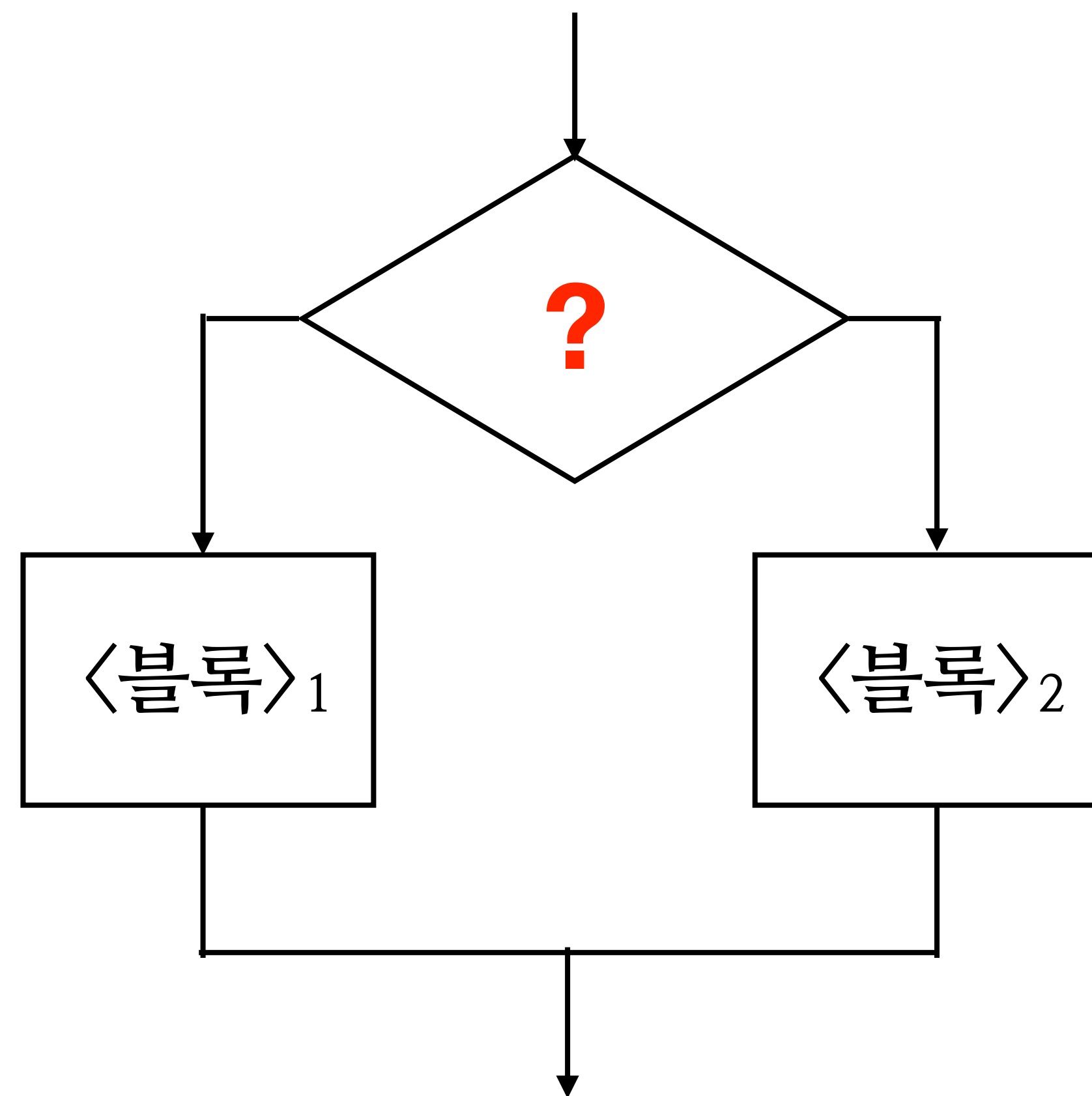
순차 제어



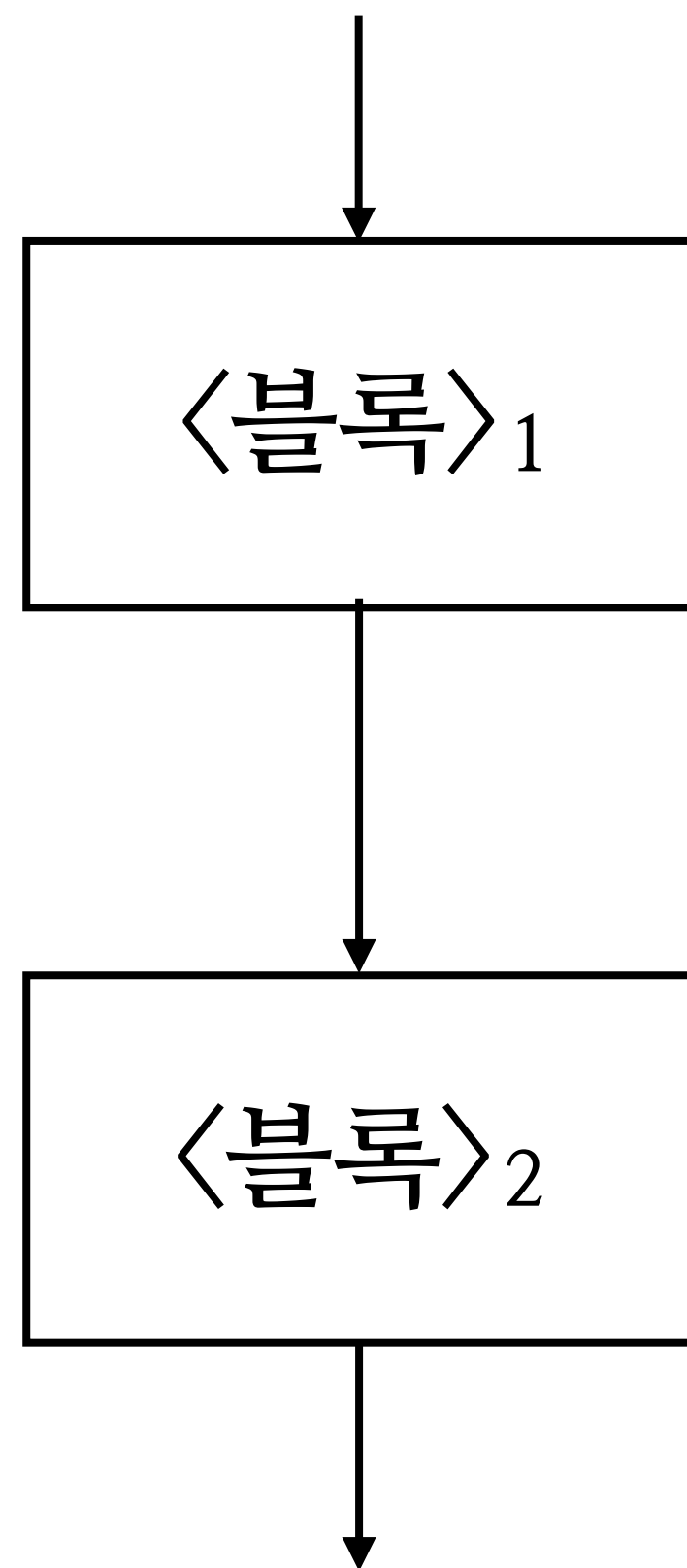
순차 제어



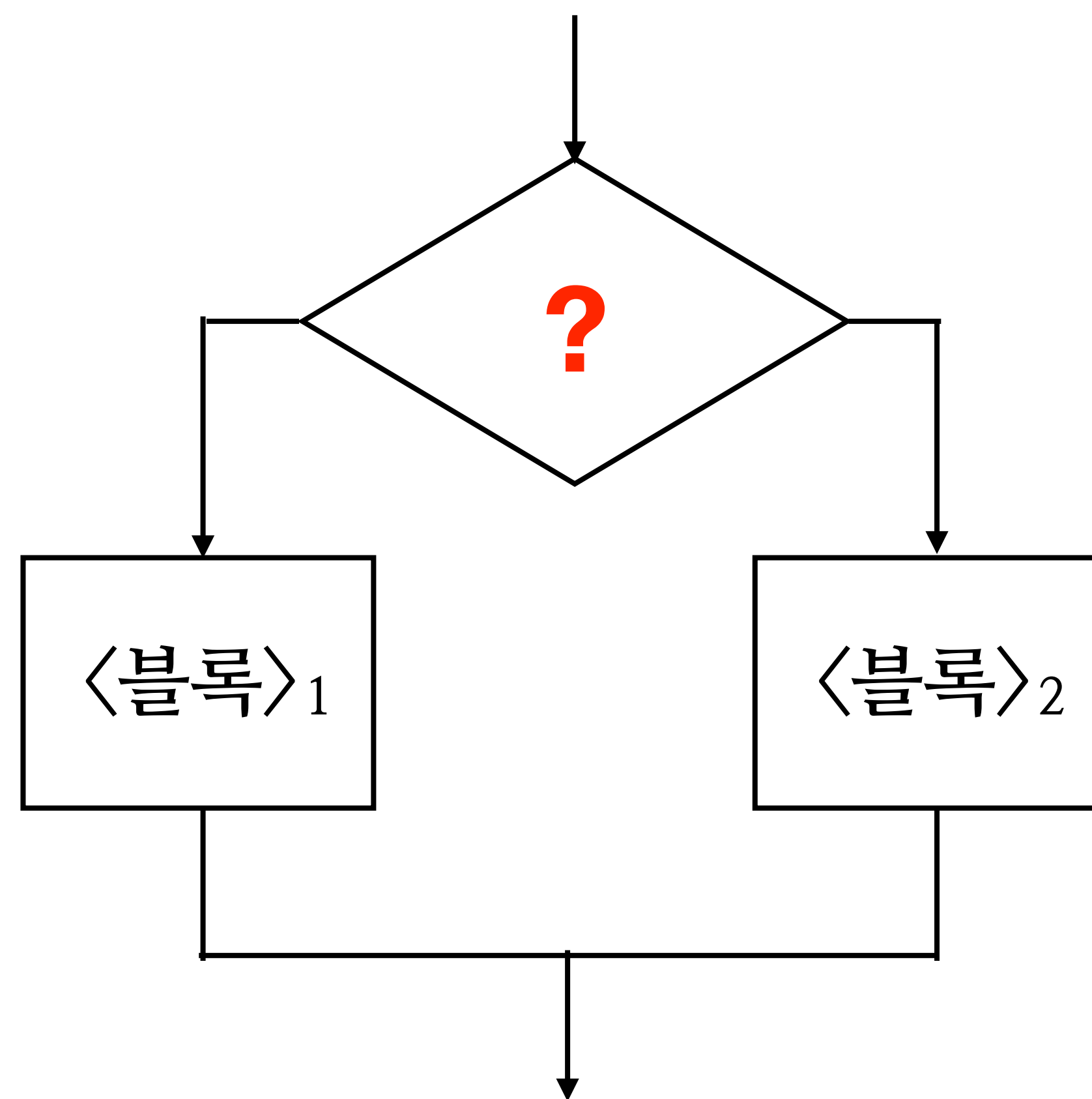
선택 제어



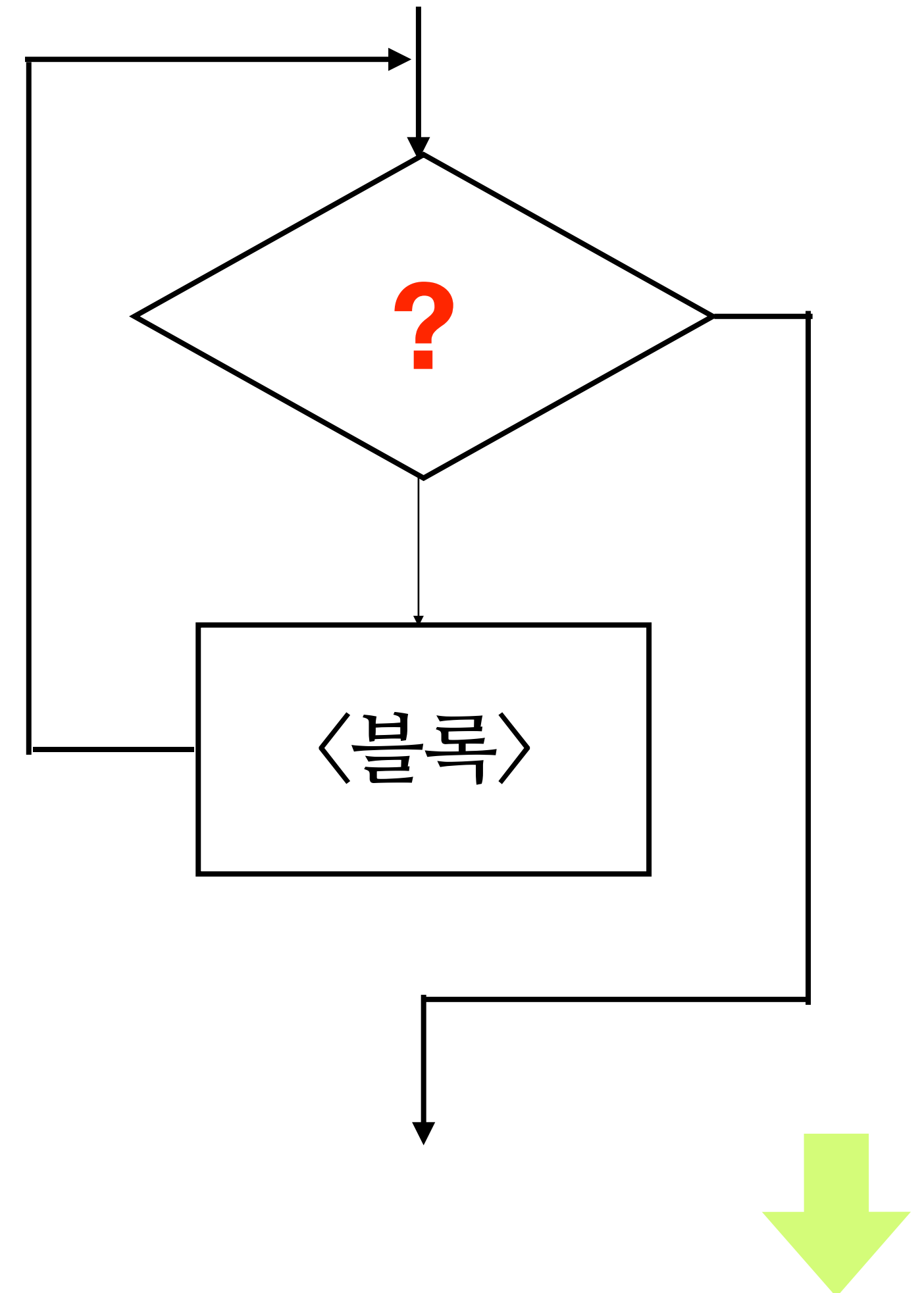
순차 제어



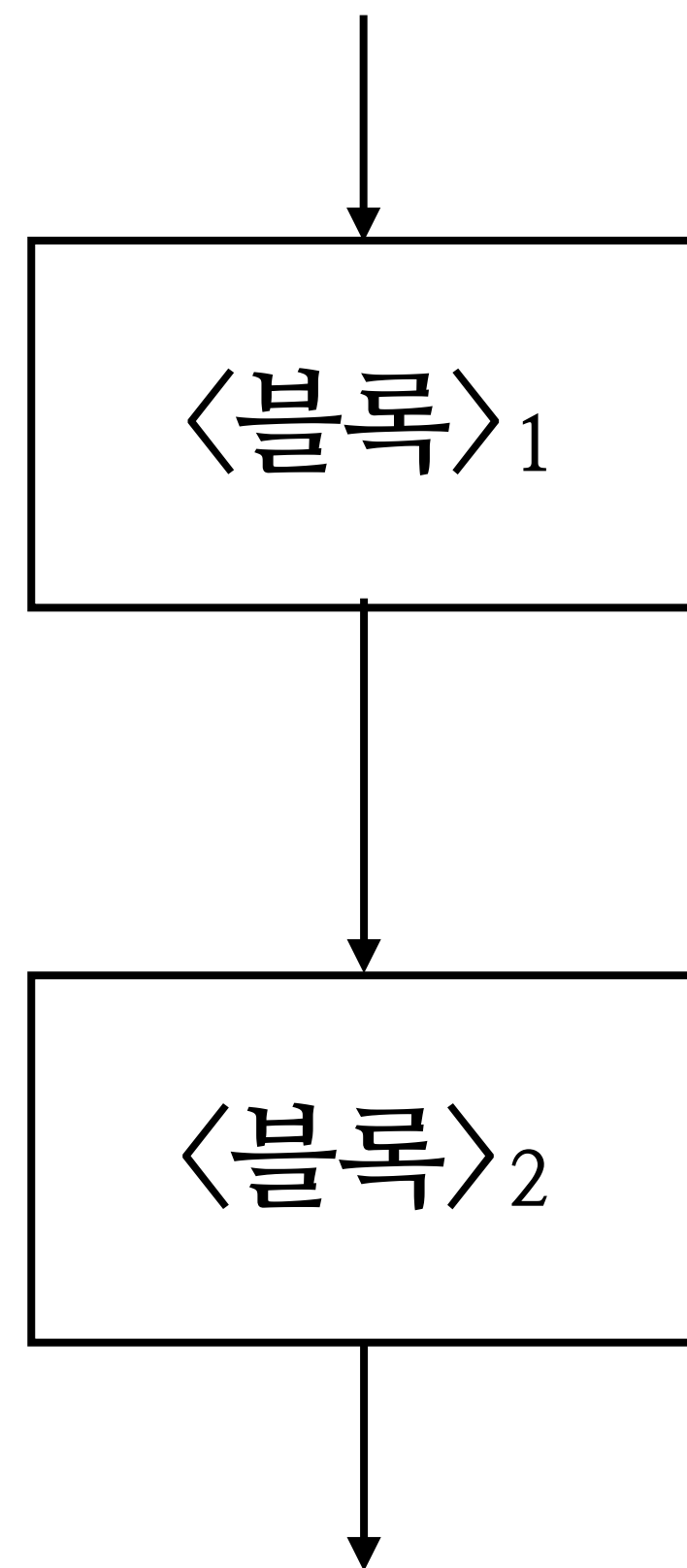
선택 제어



반복 제어

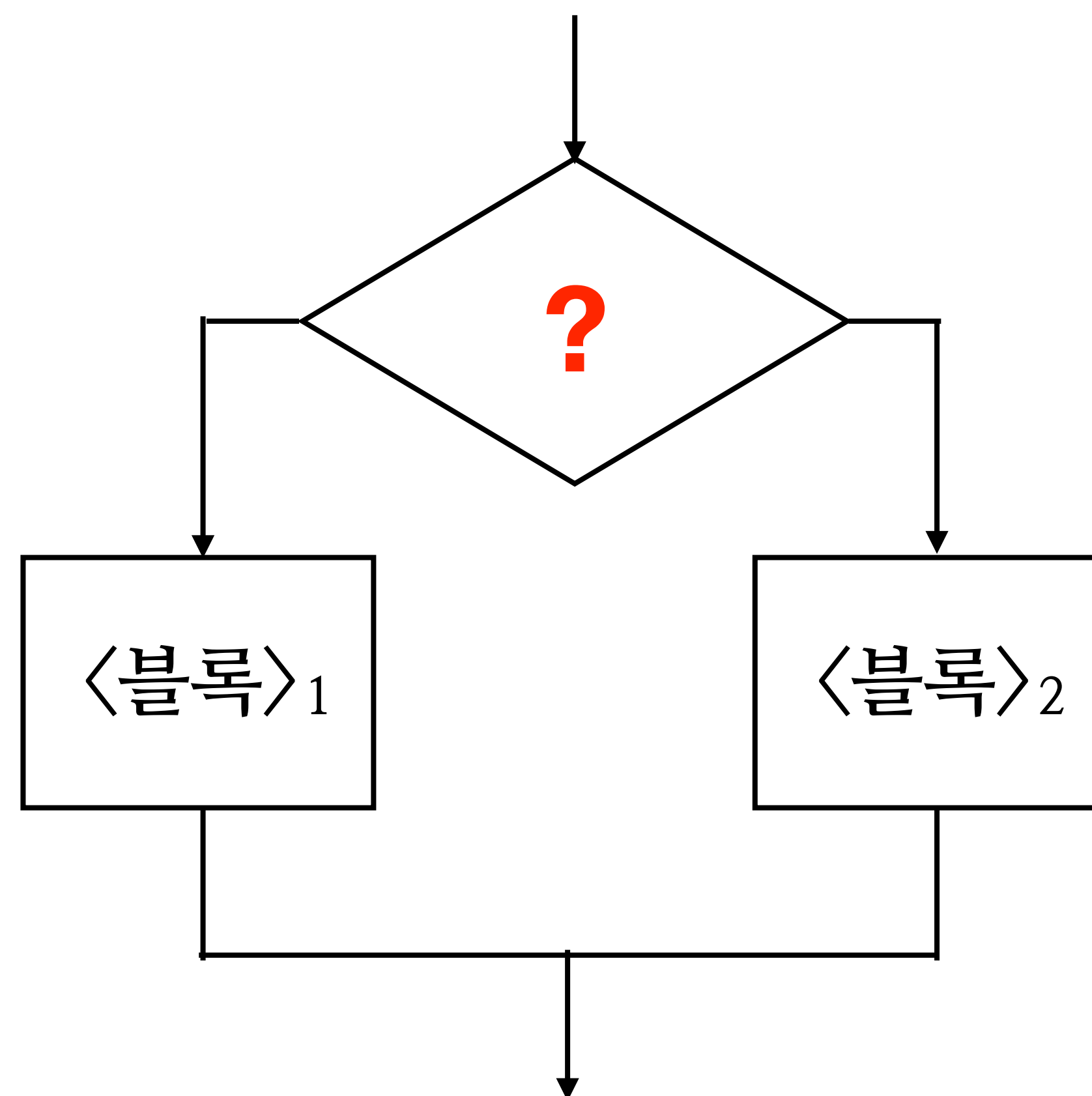


순차 제어



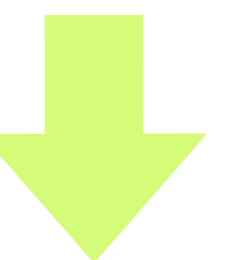
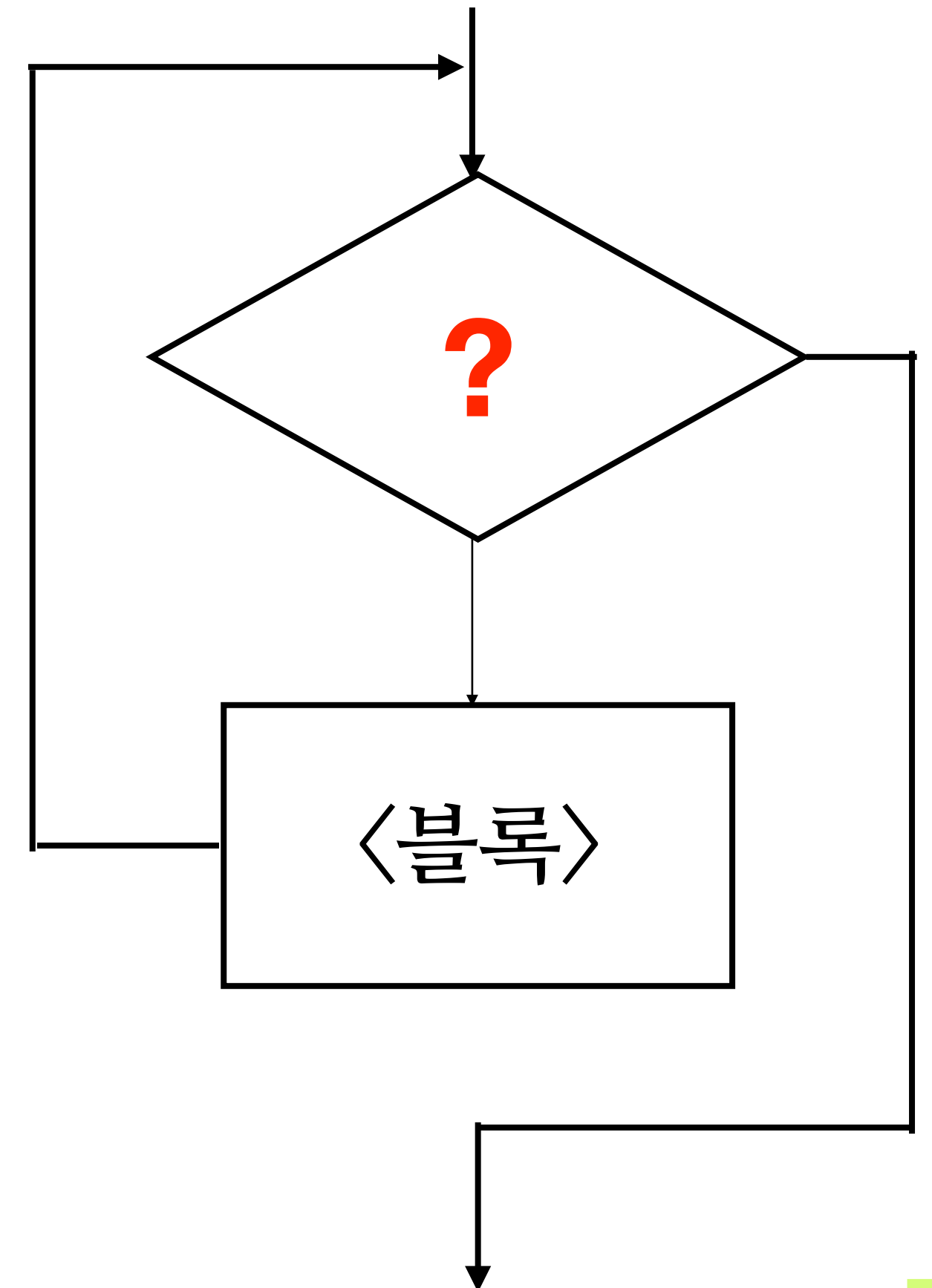
선택 제어

if-else

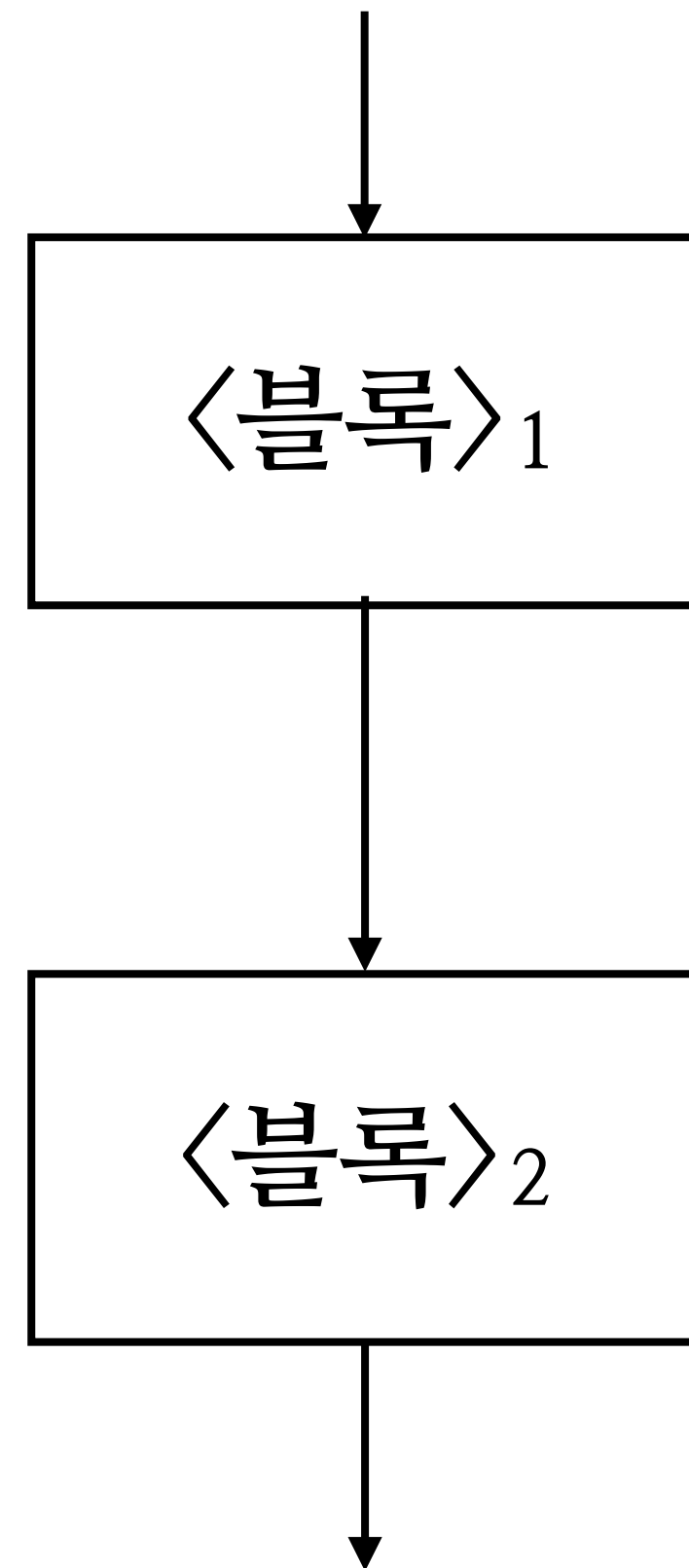


반복 제어

while

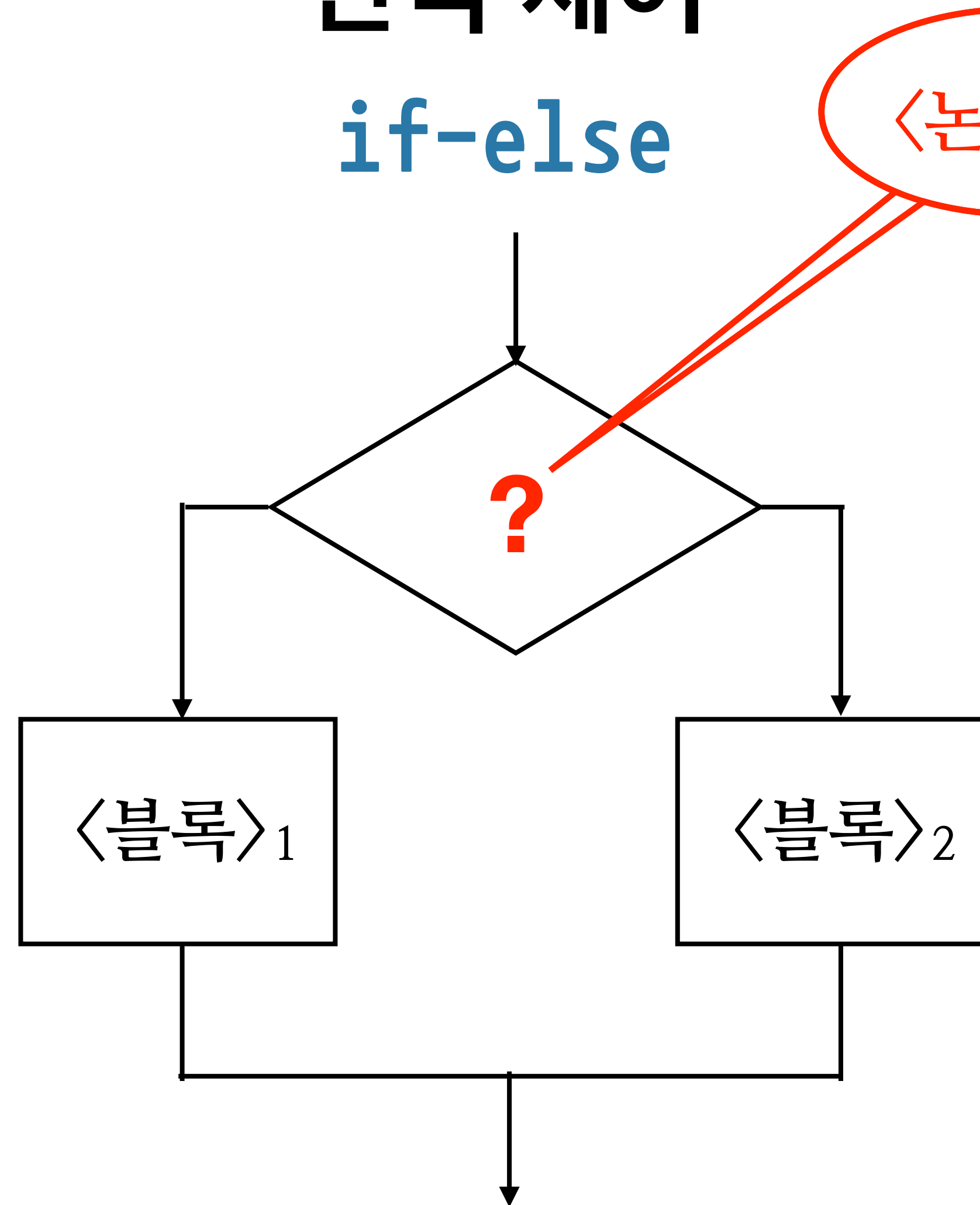


순차 제어



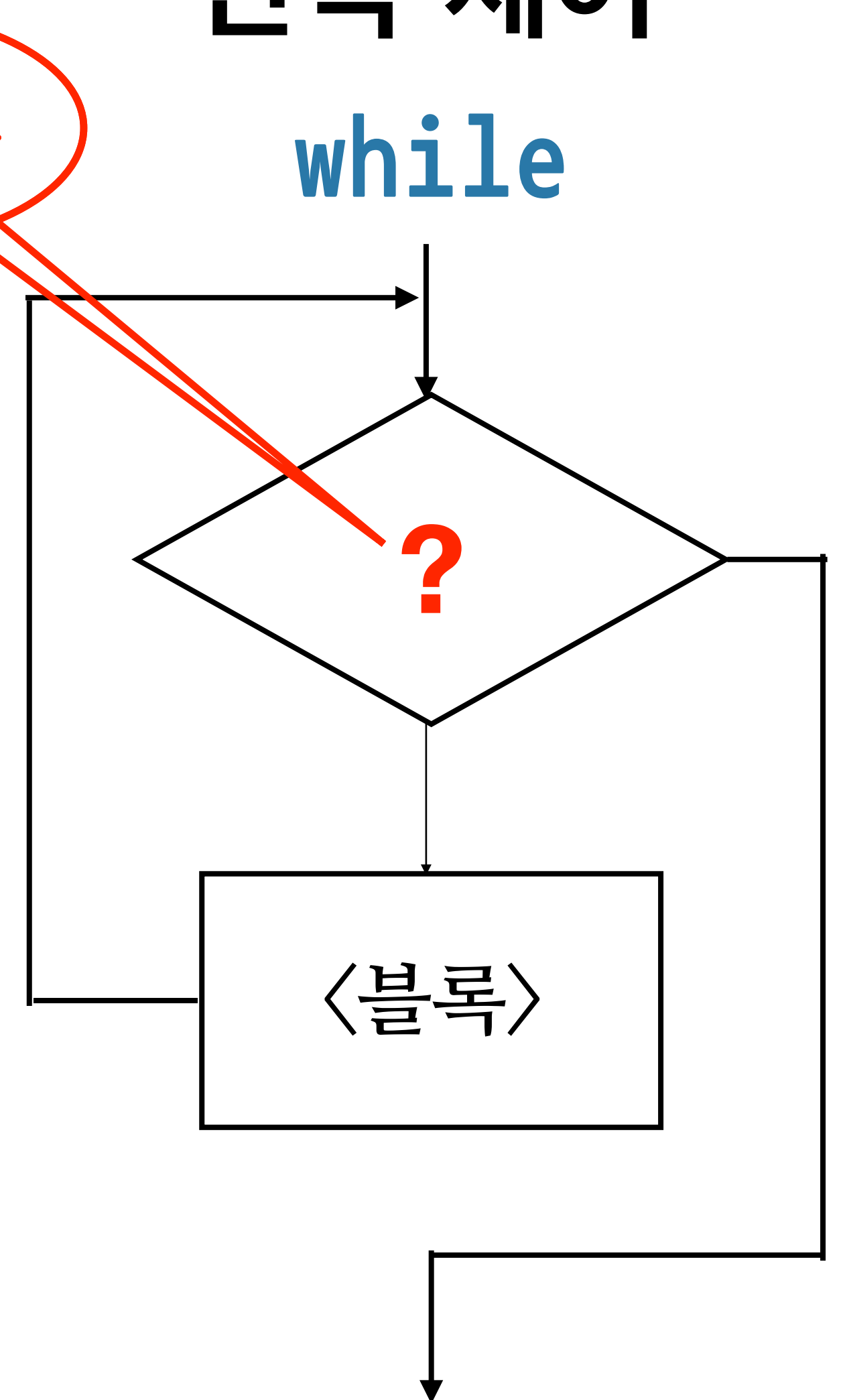
선택 제어

if-else



반복 제어

while



프로그래밍의 정석
파이썬

3

제어 구조

3.1 논리식 · 3.2 선택문 · 3.3 반복문 · 3.4 문자열 해부

CHAPTER 3

제어 구조



3.1 논리식

3.2 선택문

3.3 반복문

3.4 문자열 해부

논리식

Logical Expression

Boolean Expression

논리값

Logical Value

Boolean Value

True

False

논리 연산자

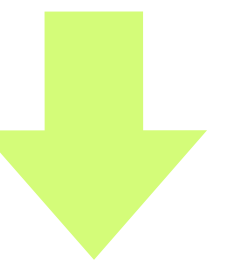
논리곱	논리합	논리역
and	or	not

〈논리식〉 and 〈논리식〉
〈논리식〉 or 〈논리식〉
not 〈논리식〉


진리표 Truth Table

p	q	p and q	p or q
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

p	not p
True	False
False	True

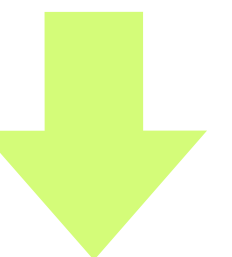


진리표 Truth Table




p	q	p and q	p or q
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

p	not p
True	False
False	True

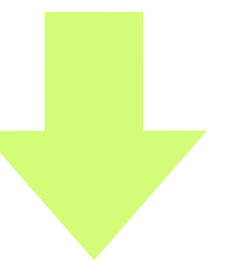


진리표 Truth Table



p	q	p and q	p or q
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False


p	not p
True	False
False	True



진리표 Truth Table

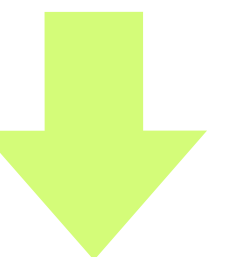
p	q	p and q	p or q
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

p	not p
True	False
False	True



연산자

연산자	우선순위	결합 순서
not	가장 높음	-
and	높음	-
or	낮음	-



연산자

연산자	우선순위	결합 순서
not	가장 높음	-
and	높음	-
or	낮음	-

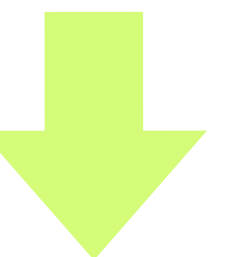
$(p \text{ and } q) \text{ and } r = p \text{ and } (q \text{ and } r)$

$(p \text{ or } q) \text{ or } r = p \text{ or } (q \text{ or } r)$

단축 계산

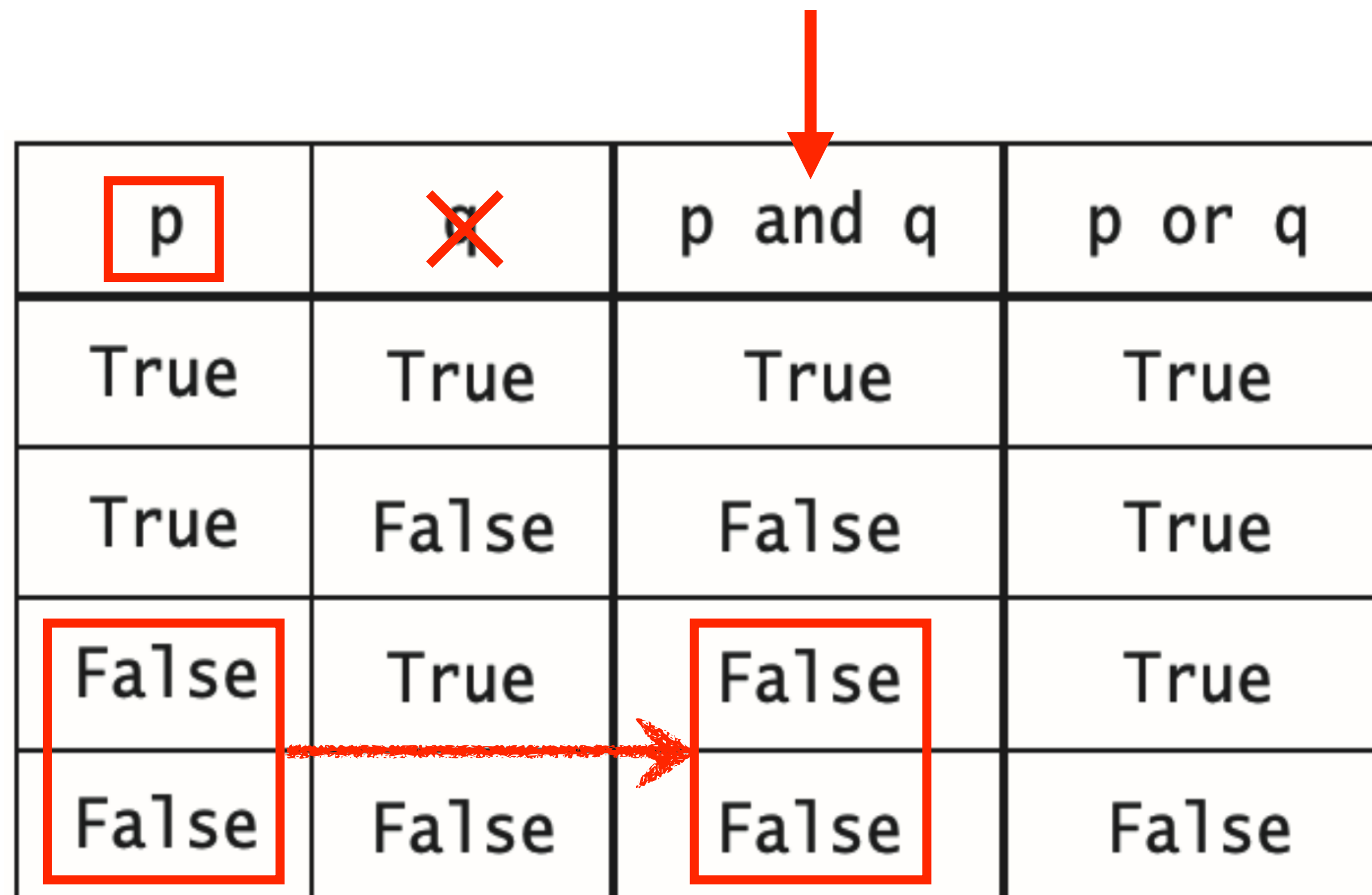
Short-circuit Evaluation

p	q	p and q	p or q
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

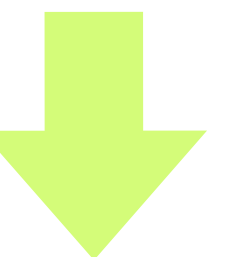


단축 계산

Short-circuit Evaluation



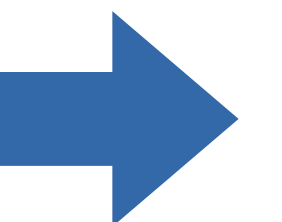
p	q	p and q	p or q
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False



단축 계산

Short-circuit Evaluation

p	q	p and q	p or q
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False



비교 논리식

비교 연산자

같다	다르다	크다	작다	크거나 같다	작거나 같다
<code>==</code>	<code>!=</code>	<code>></code>	<code><</code>	<code>>=</code>	<code><=</code>

`<식> == <식>`

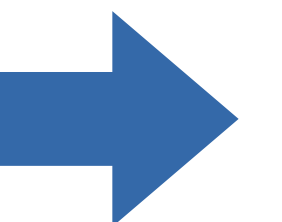
`<식> != <식>`

`<식> > <식>`

`<식> < <식>`

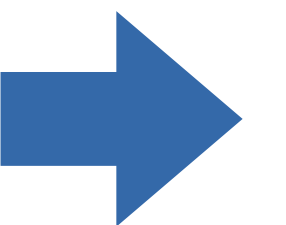
`<식> >= <식>`

`<식> <= <식>`



비교 기준

수	크기
논리값	True는 1로, False는 0으로 처리
문자열	ASCII 코드 / Unicode 값



ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	\$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(72	48	110	H	104	68	150	h
9	9	11		41	29	51)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	
29	1D	35		61	3D	75	=	93	5D	135]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

>>>>>>>>>> 제어 구조의 설계 원리를 중심으로 배우는 >>>>>>>>>>

프로그래밍의 정석 파이썬

도경구 지음



p.100



실습 3.1 짝수 확인 함수

프로그래밍의 정석
파이썬

3

제어 구조

3.1 논리식 · 3.2 선택문 · 3.3 반복문 · 3.4 문자열 해부

CHAPTER 3

제어 구조

3.1 논리식



3.2 선택문

3.3 반복문

3.4 문자열 해부

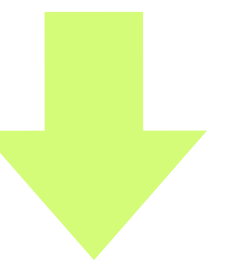
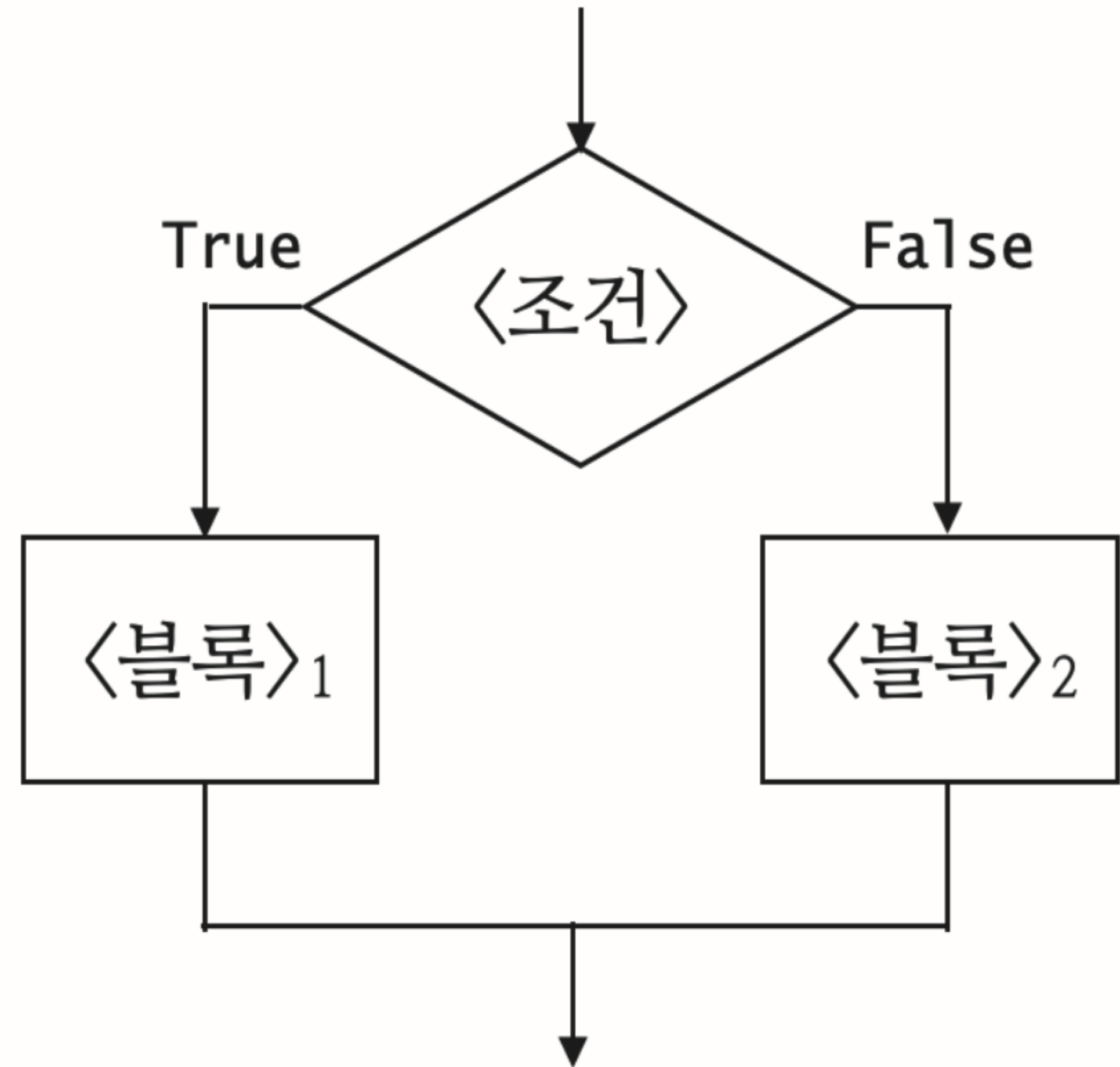
선택문

Conditional Statement

구문

```
if <조건>:  
    <블록>1  
else:  
    <블록>2
```

의미

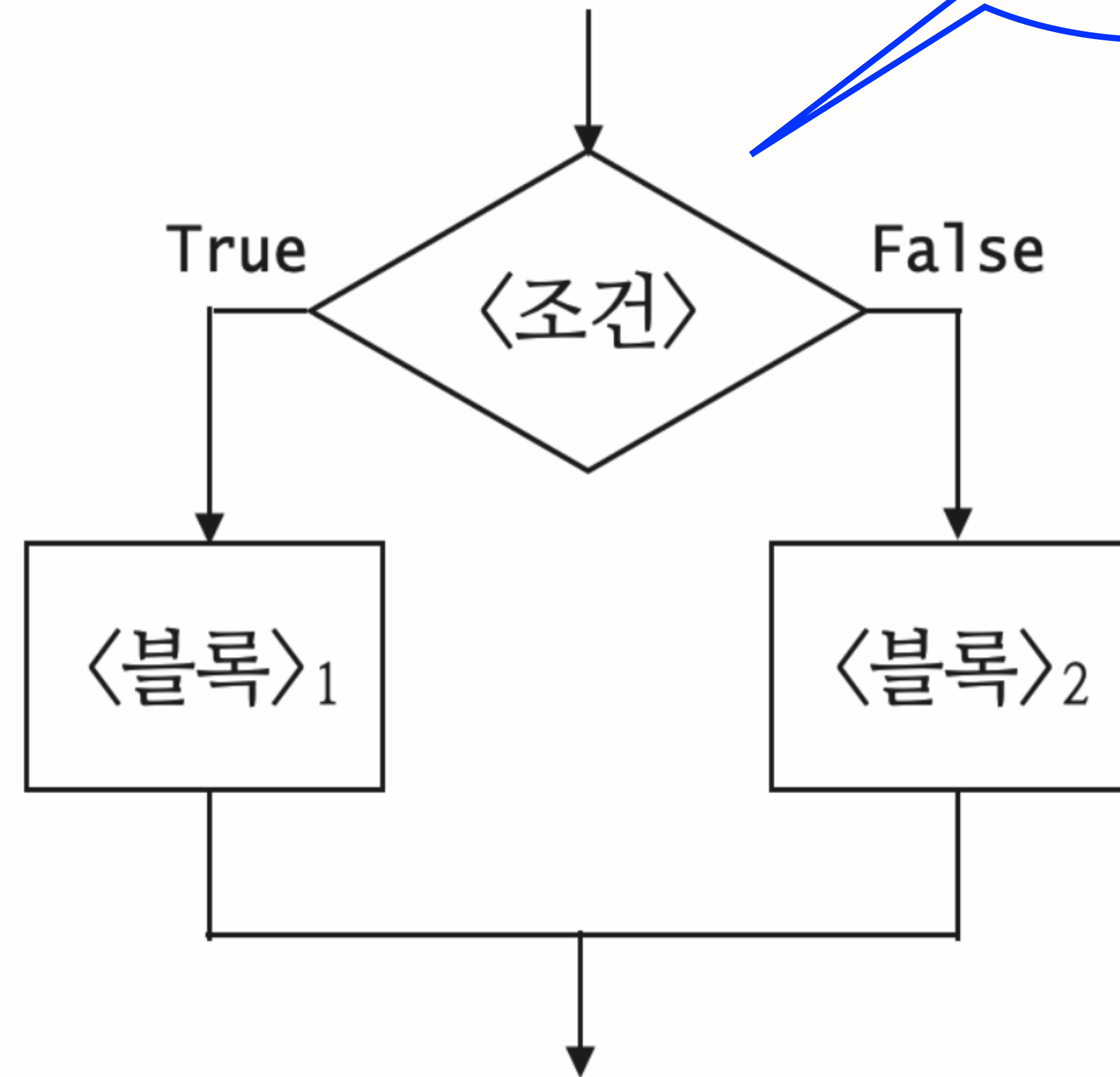


구문

```
if <조건>:  
    <블록>1  
else:  
    <블록>2
```

의미

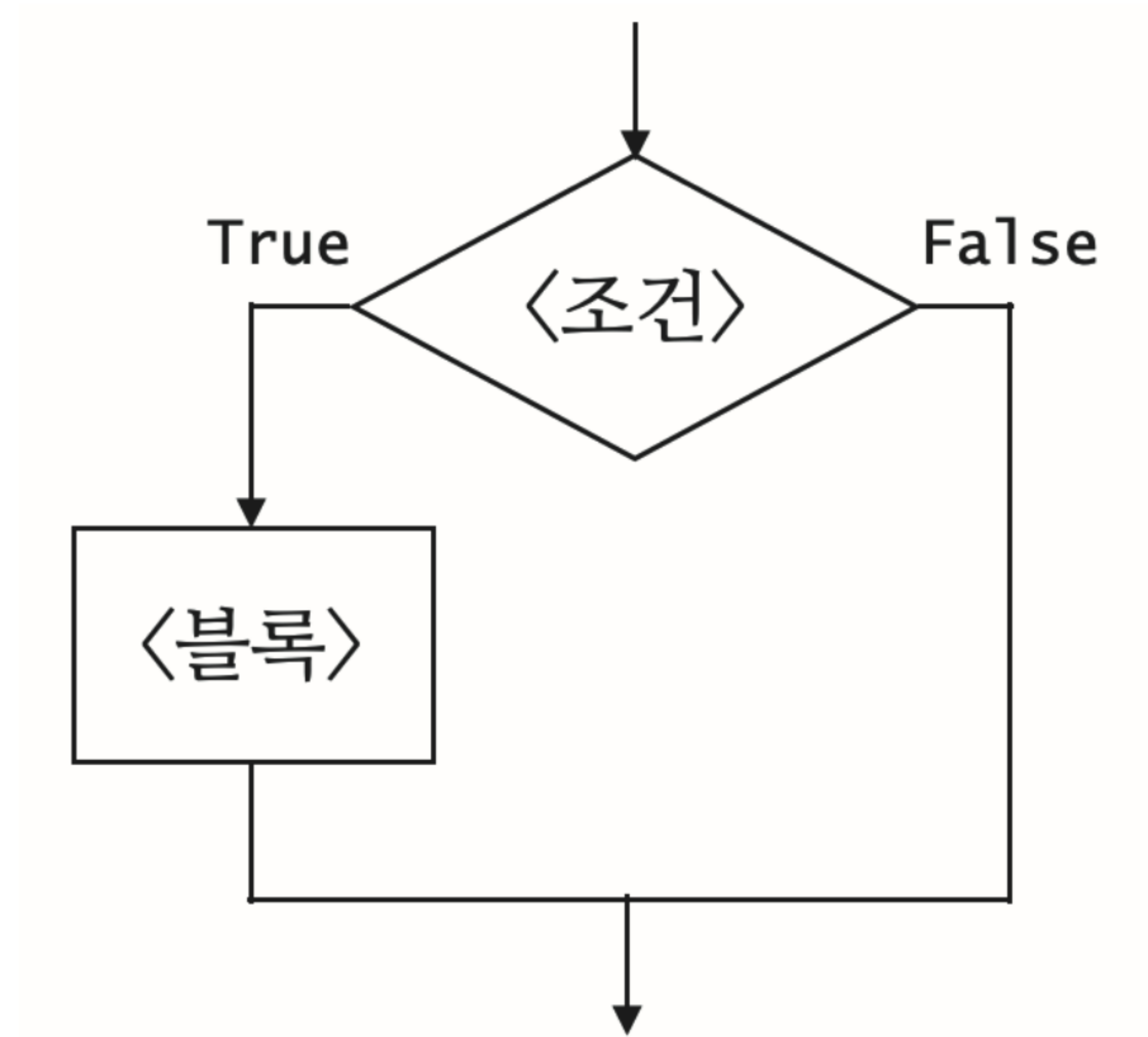
흐름도
Flow Chart



구문

```
if <조건>:  
    <블록>
```

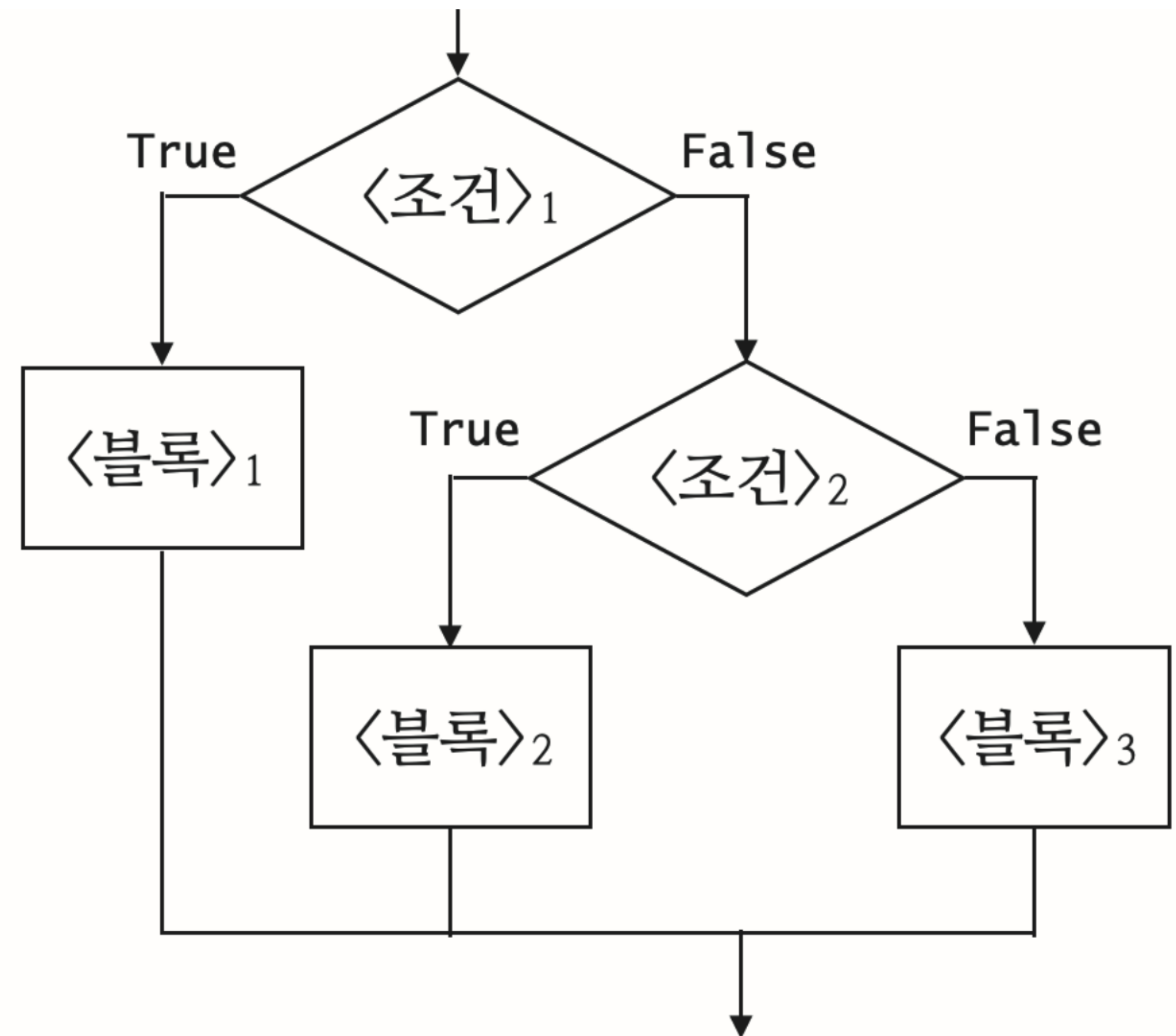
의미



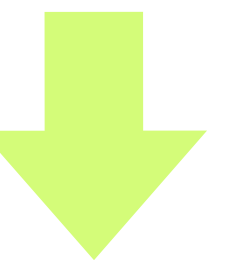
구문

```
if <조건>1:  
    <블록>1  
elif <조건>2:  
    <블록>2  
else:  
    <블록>3
```

의미



```
if <조건>1:  
    <블록>1  
elif <조건>2:  
    <블록>2  
elif ...  
else:  
    <블록>3
```



```
if <조건>1:  
    <블록>1  
elif <조건>2:  
    <블록>2  
elif ...  
else:  
    <블록>3
```

```
if <조건>1:  
    <블록>1  
elif <조건>2:  
    <블록>2  
elif ...
```

>>>>>>>>>> 제어 구조의 설계 원리를 중심으로 배우는 >>>>>>>>>>

프로그래밍의 정석 파이썬

도경구 지음



p.106



실습 3.2 흐름도 그리기

선택문의 중첩

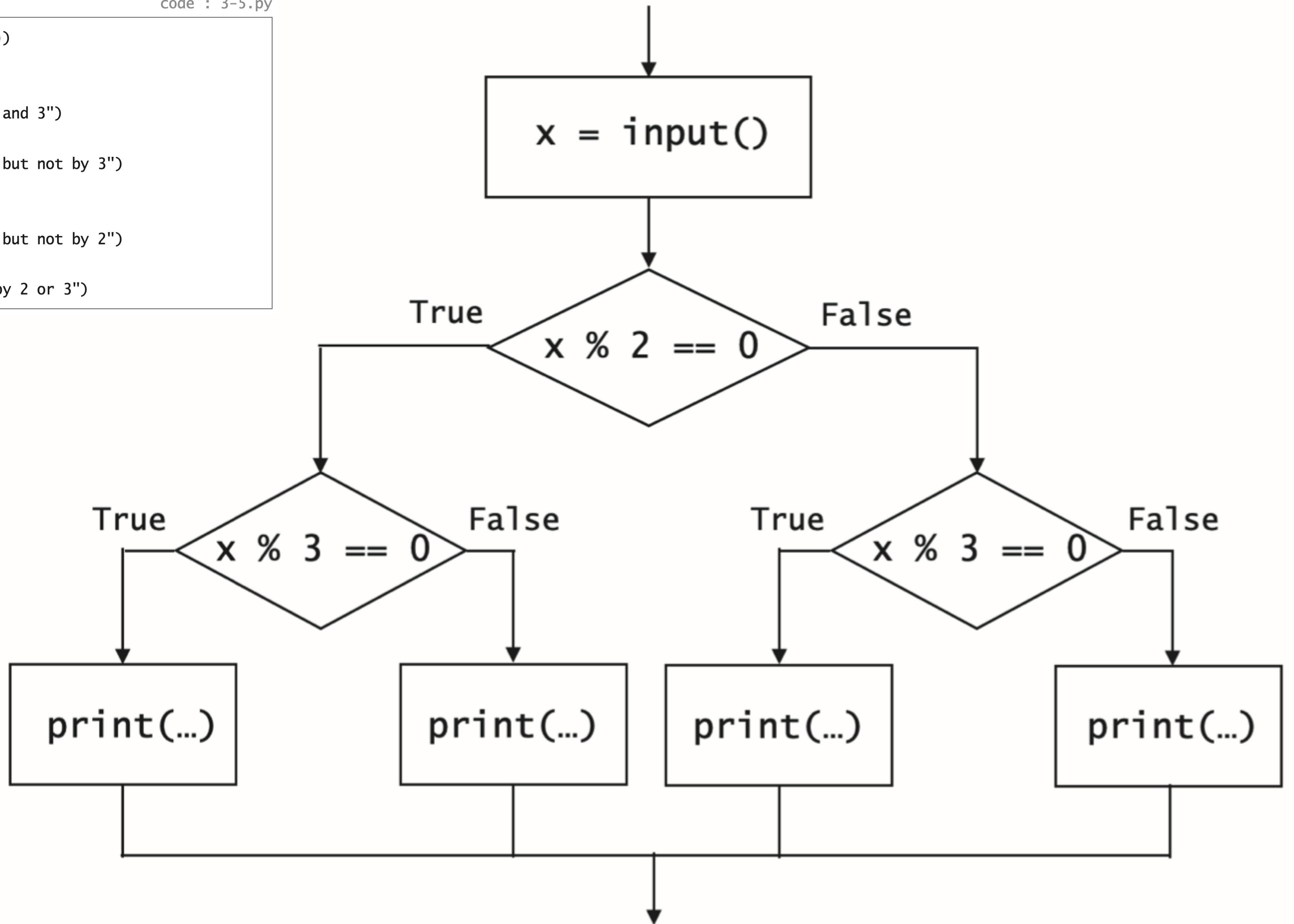
nested

code : 3-5.py

```
1 x = int(input("Enter your number: "))
2 if x % 2 == 0:
3     if x % 3 == 0:
4         print(x, "is divisible by 2 and 3")
5     else:
6         print(x, "is divisible by 2 but not by 3")
7 else:
8     if x % 3 == 0:
9         print(x, "is divisible by 3 but not by 2")
10    else:
11        print(x, "is not divisible by 2 or 3")
```

code : 3-5.py

```
1 x = int(input("Enter your number: "))
2 if x % 2 == 0:
3     if x % 3 == 0:
4         print(x, "is divisible by 2 and 3")
5     else:
6         print(x, "is divisible by 2 but not by 3")
7 else:
8     if x % 3 == 0:
9         print(x, "is divisible by 3 but not by 2")
10    else:
11        print(x, "is not divisible by 2 or 3")
```



code : 3-5.py

```
1 x = int(input("Enter your number: "))
2 if x % 2 == 0:
3     if x % 3 == 0:
4         print(x, "is divisible by 2 and 3")
5     else:
6         print(x, "is divisible by 2 but not by 3")
7 else:
8     if x % 3 == 0:
9         print(x, "is divisible by 3 but not by 2")
10    else:
11        print(x, "is not divisible by 2 or 3")
```

and 연산자로 중첩 선택문 펼치기

code : 3-6.py

```
1 x = int(input("Enter your number: "))
2 if x % 2 == 0 and x % 3 == 0:
3     print(x, "is divisible by 2 and 3")
4 elif x % 2 == 0 and not(x % 3 == 0):
5     print(x, "is divisible by 2 but not by 3")
6 elif not(x % 2 == 0) and x % 3 == 0:
7     print(x, "is divisible by 3 but not by 2")
8 else:
9     print(x, "is not divisible by 2 or 3")
```

>>>>>>>>>> 제어 구조의 설계 원리를 중심으로 배우는 >>>>>>>>>>

프로그래밍의 정석 파이썬

도경구 지음



pp.108~110



실습 3.3 중첩 선택문의 실행 경로 따져보기



실습 3.4 둘 중 작은 수 찾기 함수



실습 3.5 셋 중 가장 작은 수 찾기 함수



실습 3.6 셋 중 가장 작은 수 찾기 함수 (smaller 활용)

프로그래밍의 정석
파이썬

3

제어 구조

3.1 논리식 · 3.2 선택문 · 3.3 반복문 · 3.4 문자열 해부

CHAPTER 3

제어 구조

3.1 논리식

3.2 선택문

✓ 3.3 반복문

3.4 문자열 해부

반복문

Loop

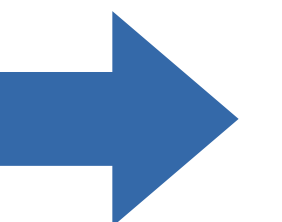
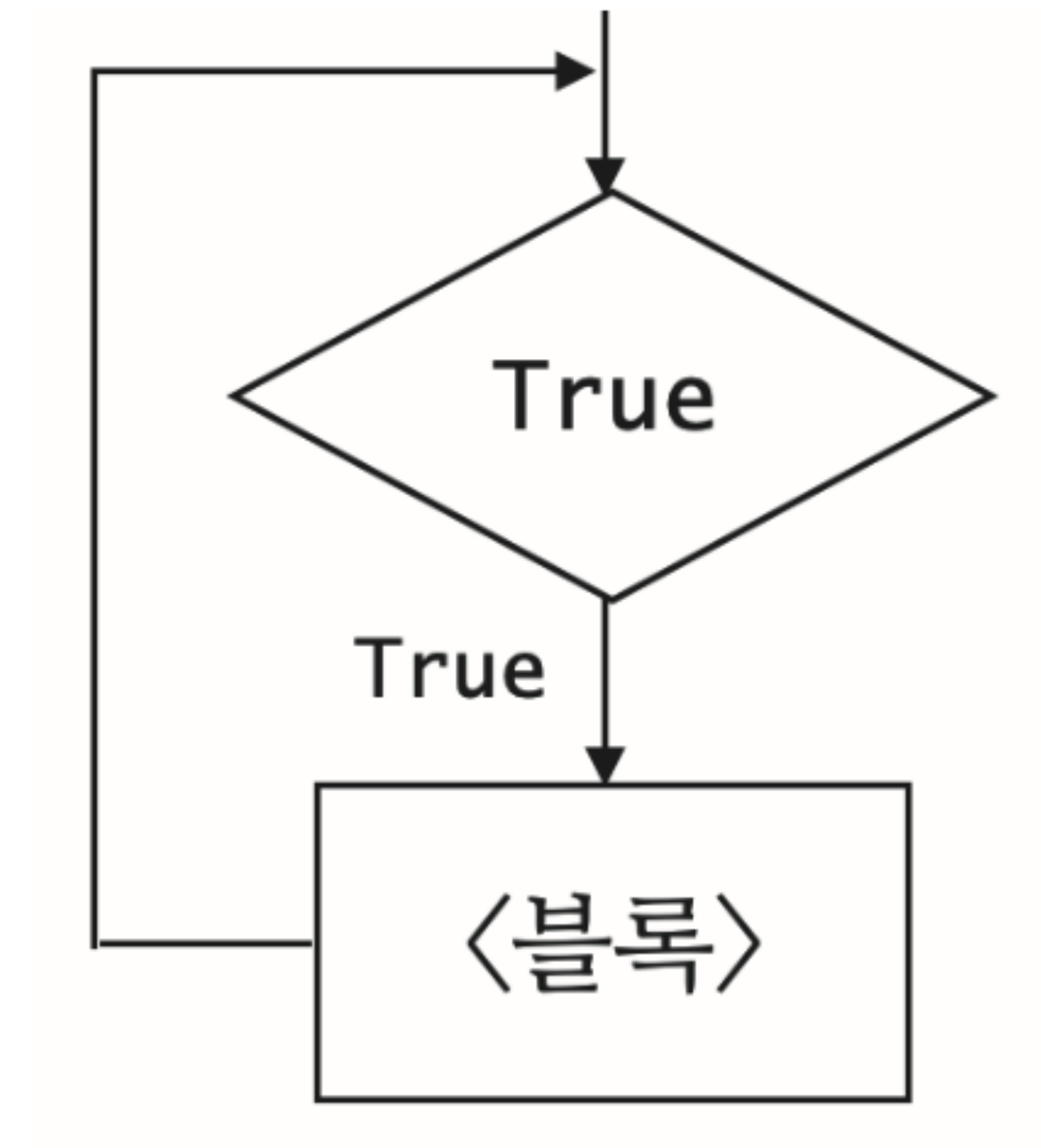
무조건 반복

Infinite Loop

구문

```
while True:  
    <블록>
```

의미



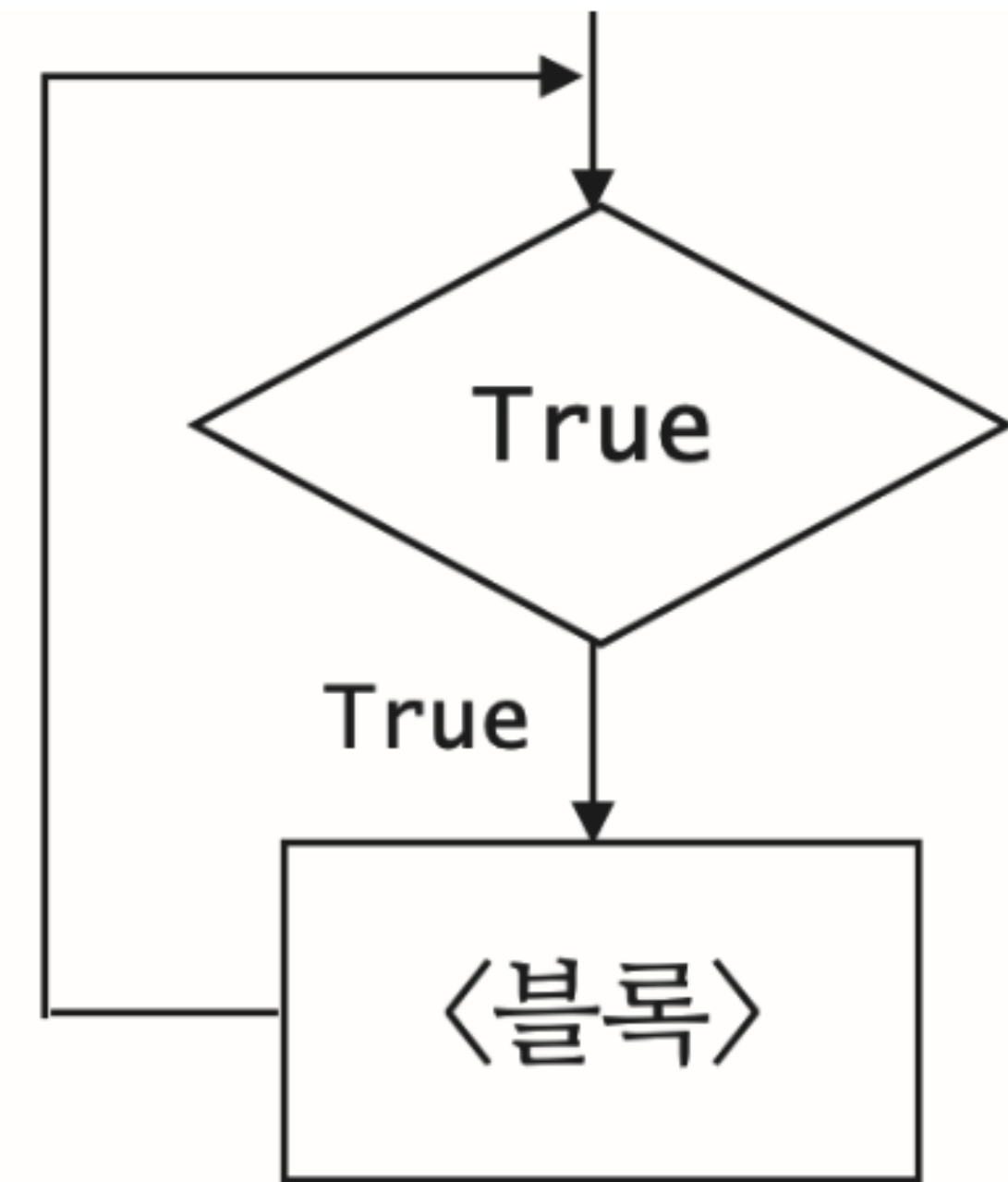
무조건 반복

Infinite Loop

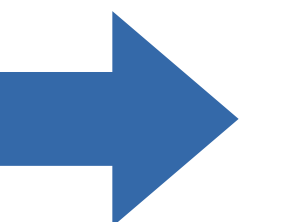
구문

```
while True:  
    <블록>
```

의미



강제 종료 :

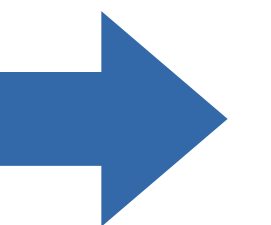
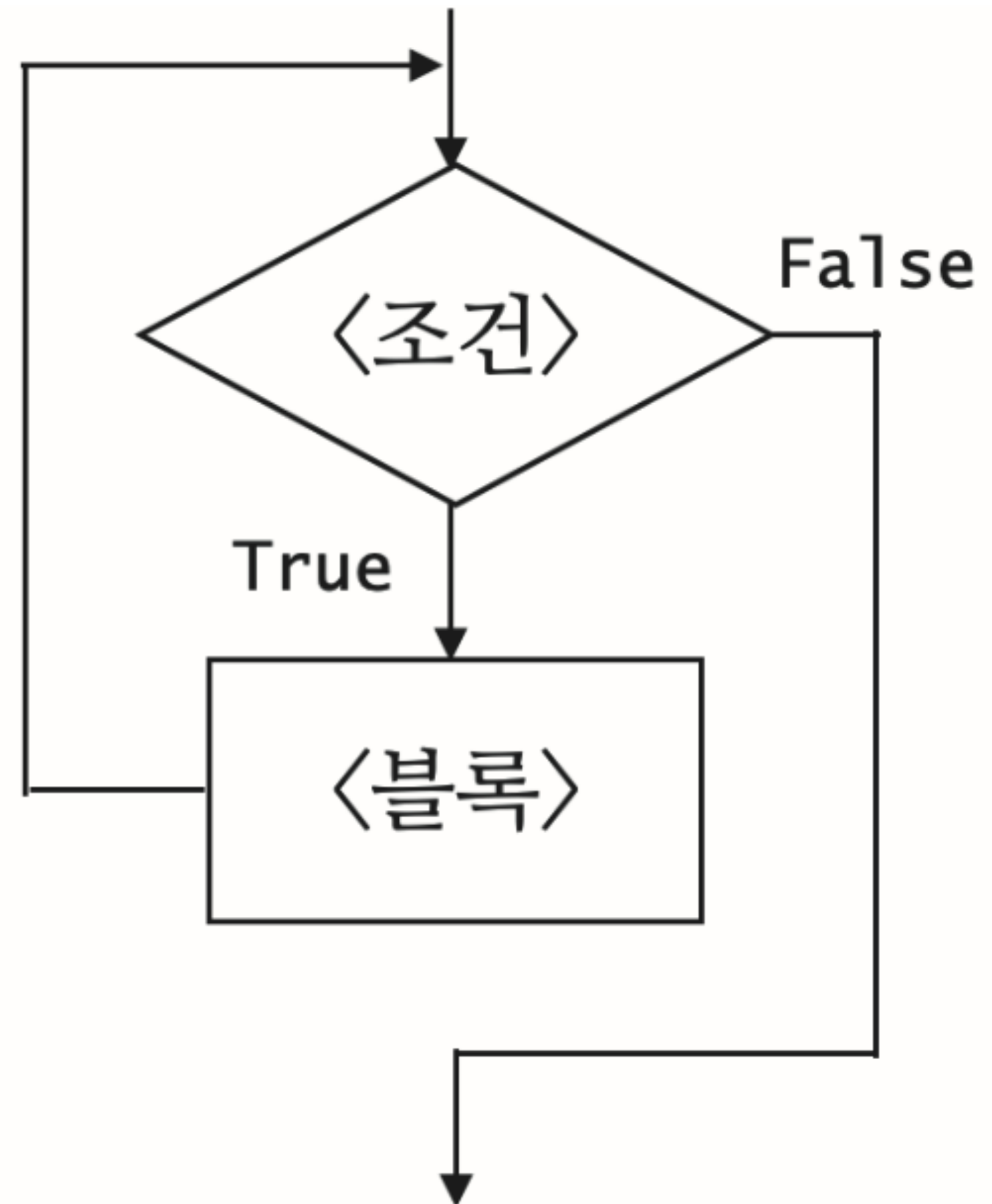


조건 반복

구문

```
while <조건>:  
    <블록>
```

의미



>>>>>>>>>> 제어 구조의 설계 원리를 중심으로 배우는 >>>>>>>>>>

프로그래밍의 정석 파이썬

도경구 지음



pp.115~117



실습 3.7 수강과목 평균 점수 계산 서비스

사례 학습

입력 확인

Input Validation

문자열은 객체이다!

객체 object

"365"

"365.0"

"-17"

"freedom"

클래스 class

str	
속성	?
메소드	. . .
	isdigit()
	. . .

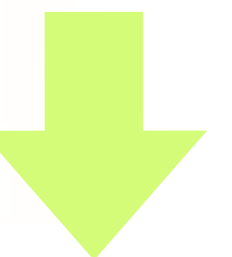
객체 object

"365"

"365.0"

"-17"

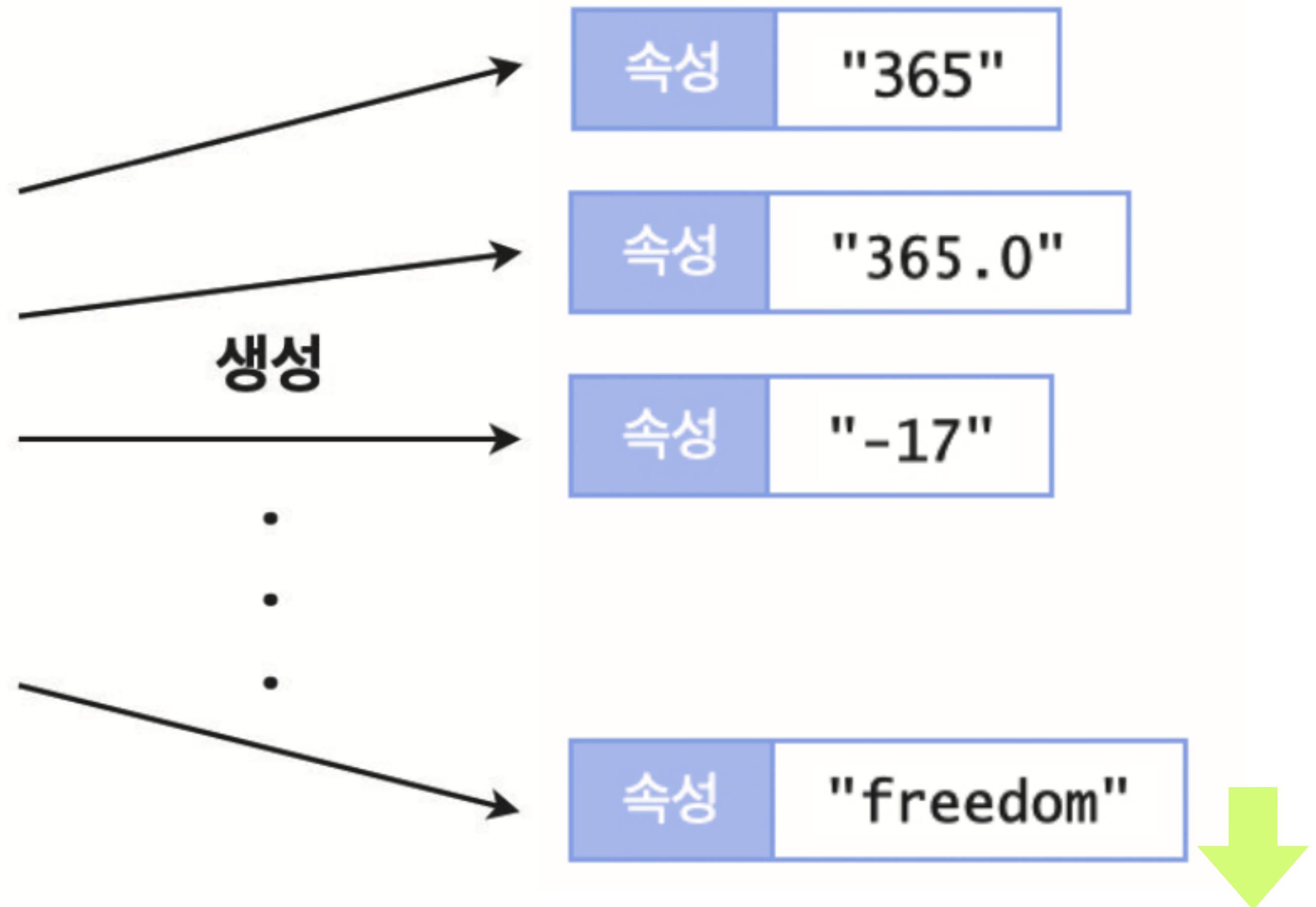
"freedom"



클래스 class

str	
속성	?
메소드	. . .
	isdigit()
	. . .

객체 object



형판

실체

클래스 class

객체 object

str	
속성	?
메소드	. . .
	isdigit()
	. . .

생성

속성

"365"

속성

"365.0"

속성

"-17"

·
·
·

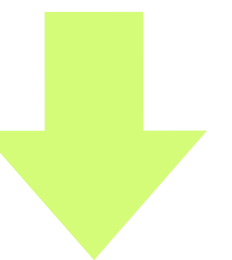
속성

"freedom"

메소드

method

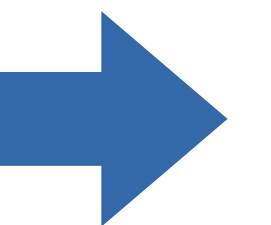
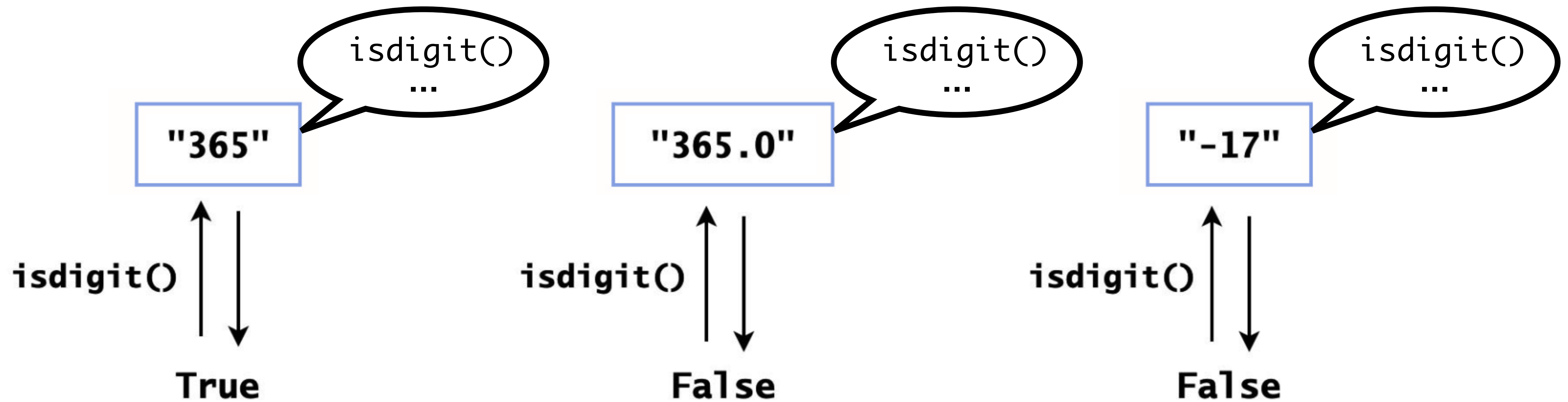
〈문자열〉.isdigit()



메소드

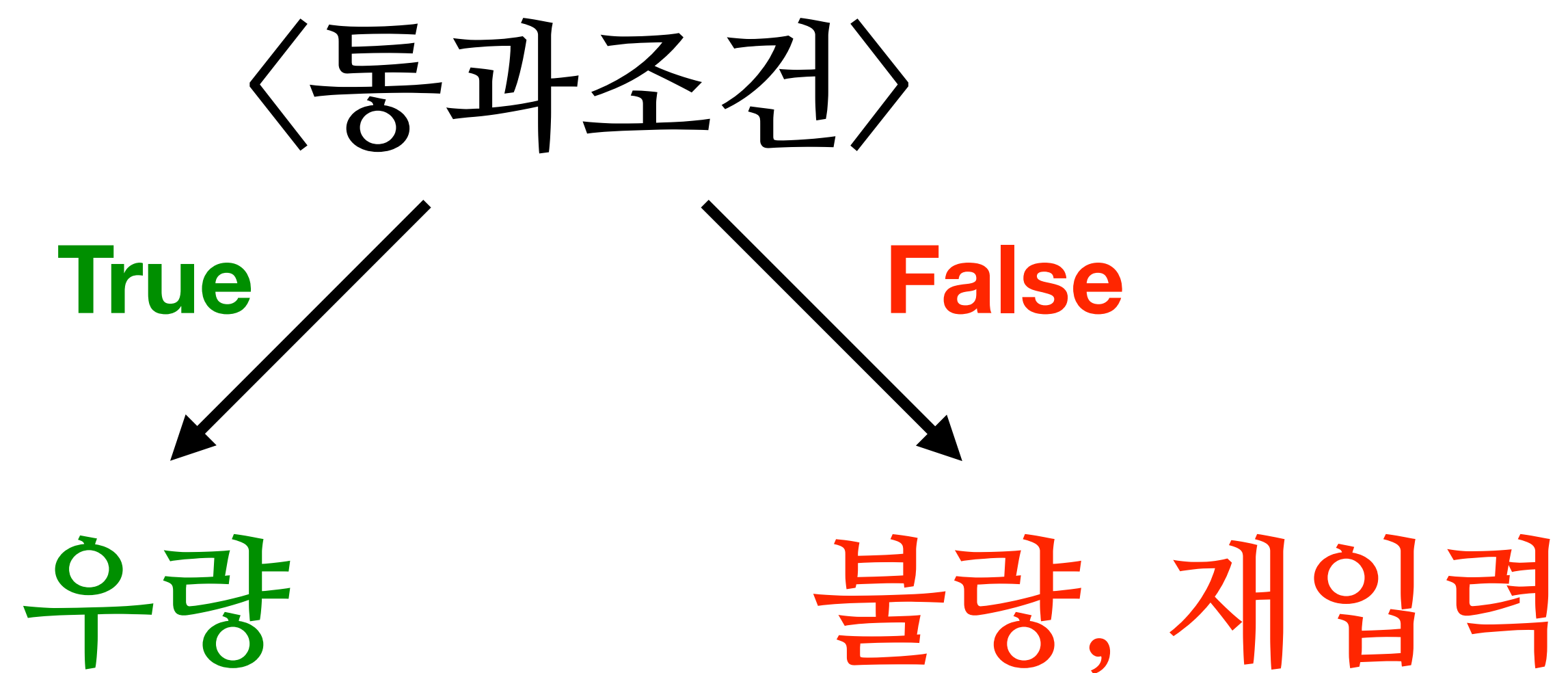
method

〈문자열〉.isdigit()



입력 확인

Input Validation

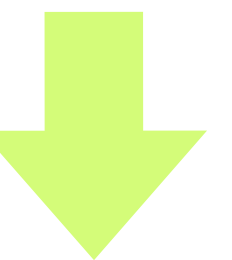


입력 확인 코드 패턴

```
x = input()
```

```
while not <통과조건>:
```

```
    x = input()
```



입력 확인 코드 패턴

```
x = input()
```

```
while not <통과조건>:
```

```
    x = input()
```



x.isdigit()

```
1  from math import pi
2
3  def area_circle(radius, n):
4      if radius > 0:
5          area = pi * radius ** 2
6          return round(area, n)
7      else:
8          return 0.0
9
10 print("Circle Area Calculator")
11 more = 'y'
12 while more == 'y':
13     r = input("Radius? ")
14     while not r.isdigit():
15         r = input("Radius? ")
16     p = input("Precision? ")
17     while not p.isdigit():
18         p = input("Precision? ")
19     area = area_circle(int(r),int(p))
20     print("The area of a circle with radius", r, "is", area, ".")
21     more = input("Press y to continue, any other key to exit. ")
22 print("Please come back again.")
```

>>>>>>>>>> 제어 구조의 설계 원리를 중심으로 배우는 >>>>>>>>>>

프로그래밍의 정석 파이썬

도경구 지음



p.122



실습 3.8 수강과목 평균 점수 계산 서비스 (입력 확인)

프로그래밍의 정석
파이썬

3

제어 구조

3.1 논리식 · 3.2 선택문 · 3.3 반복문 · 3.4 문자열 해부

CHAPTER 3

제어 구조

3.1 논리식

3.2 선택문

3.3 반복문

✓ 3.4 문자열 해부

프로그래밍의 정석

파이썬



pp.127~129



인덱스

index

"컴퓨터과학"

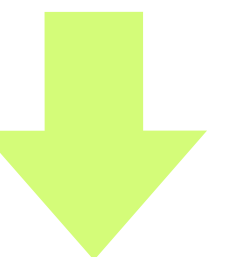
0	1	2	3	4
컴	퓨	터	과	학
-5	-4	-3	-2	-1

인덱스의 범위

$0 \sim n-1$

$-1 \sim -n$

n = 문자열의 길이



인덱스

index

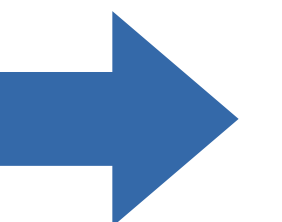
"컴퓨터과학"

0	1	2	3	4
컴	퓨	터	과	학
-5	-4	-3	-2	-1

"컴퓨터과학" [0]

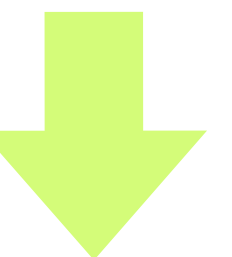
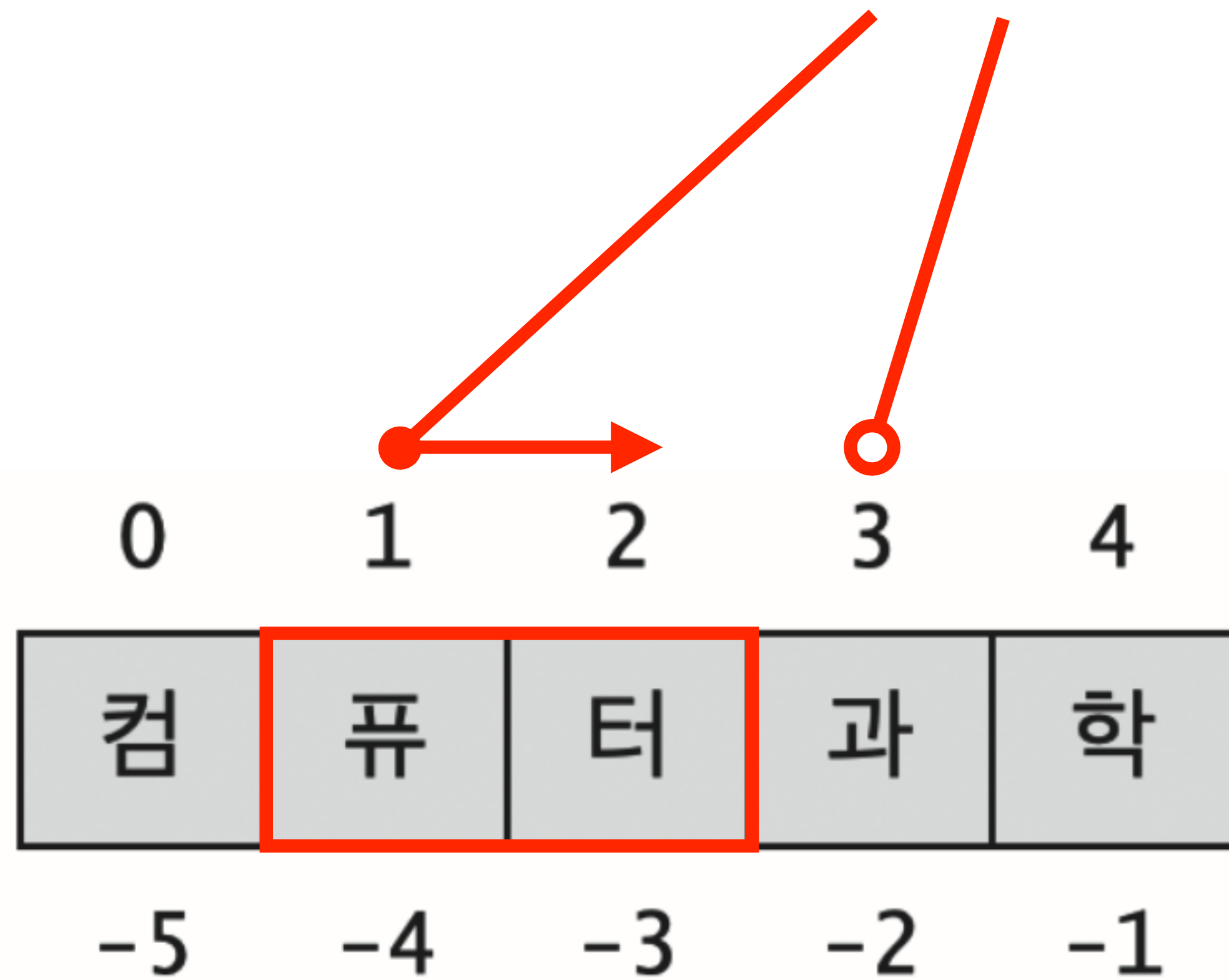
"컴퓨터과학" [2]

"컴퓨터과학" [-1]



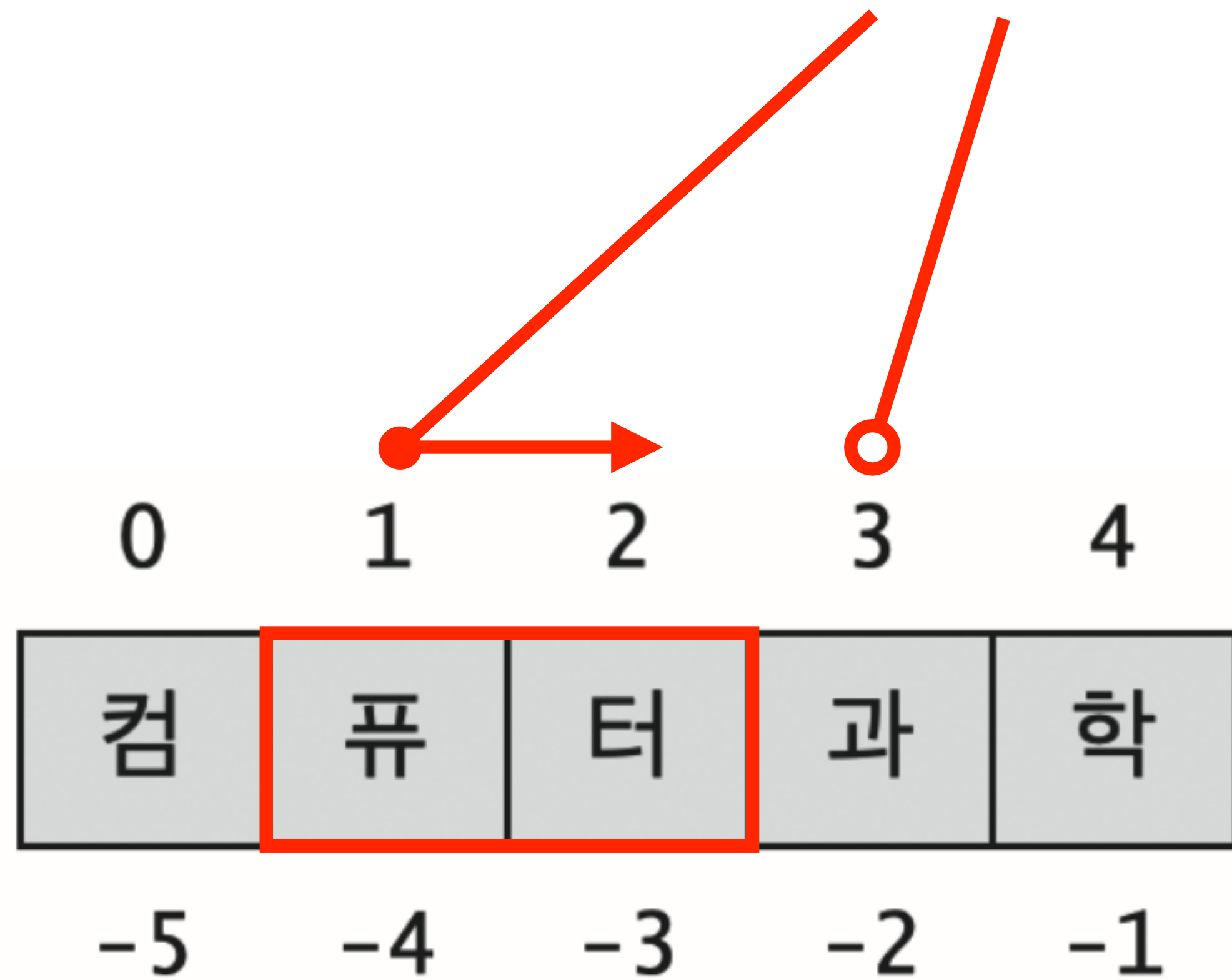
문자열 조각 복제

"컴퓨터과학" [1:3]

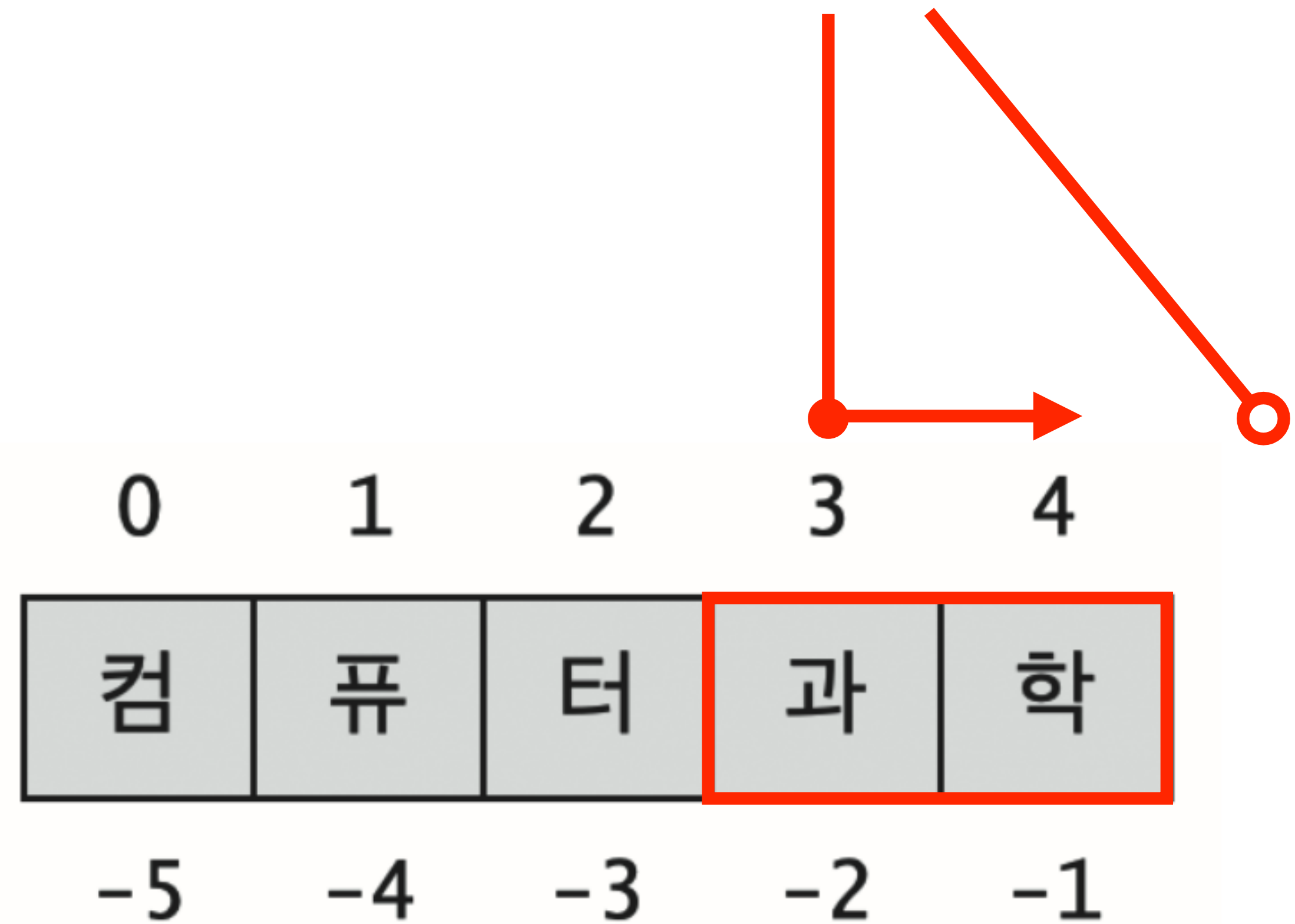


문자열 조각 복제

"컴퓨터과학" [1:3]

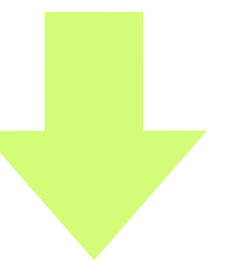
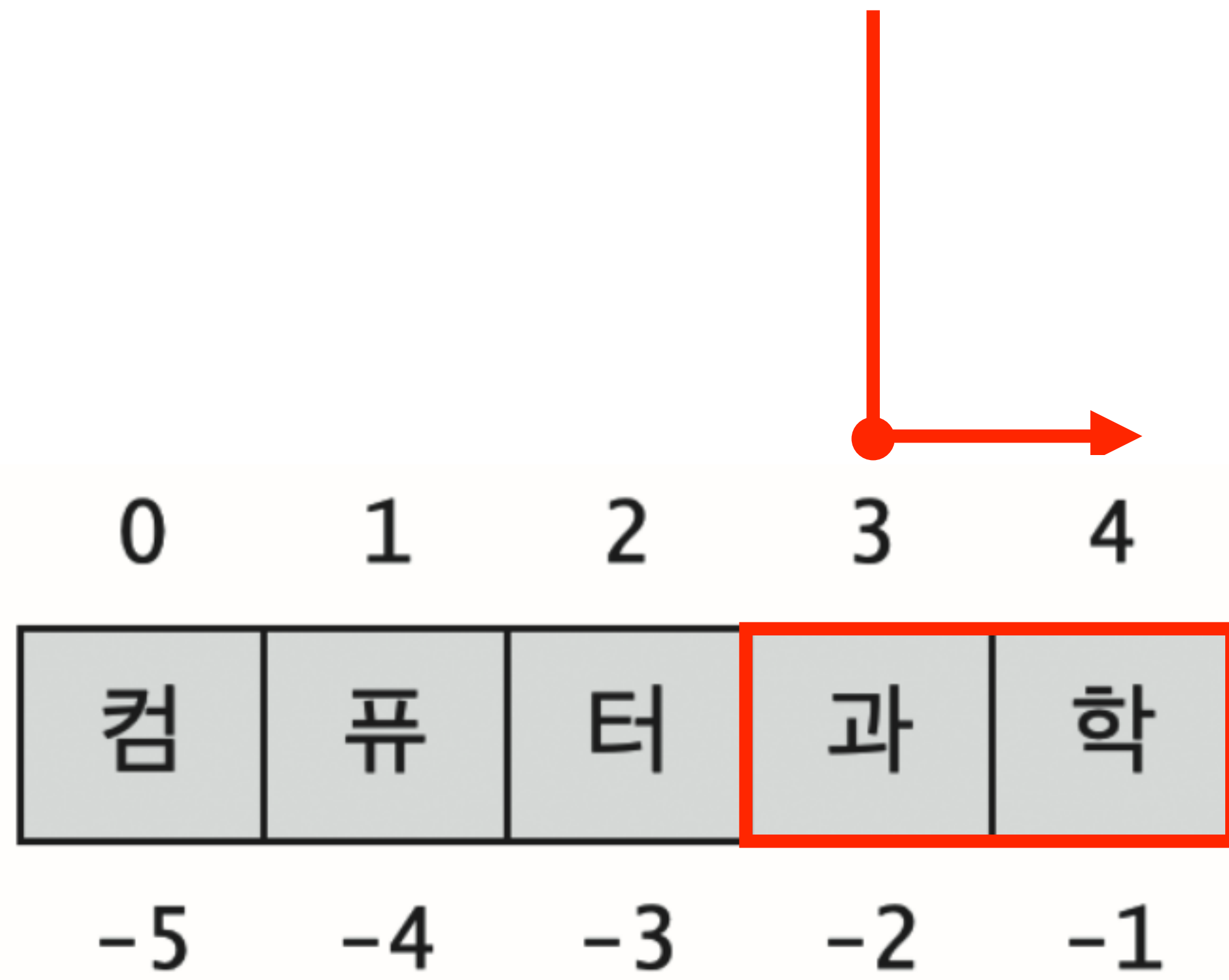


"컴퓨터과학" [3:5]



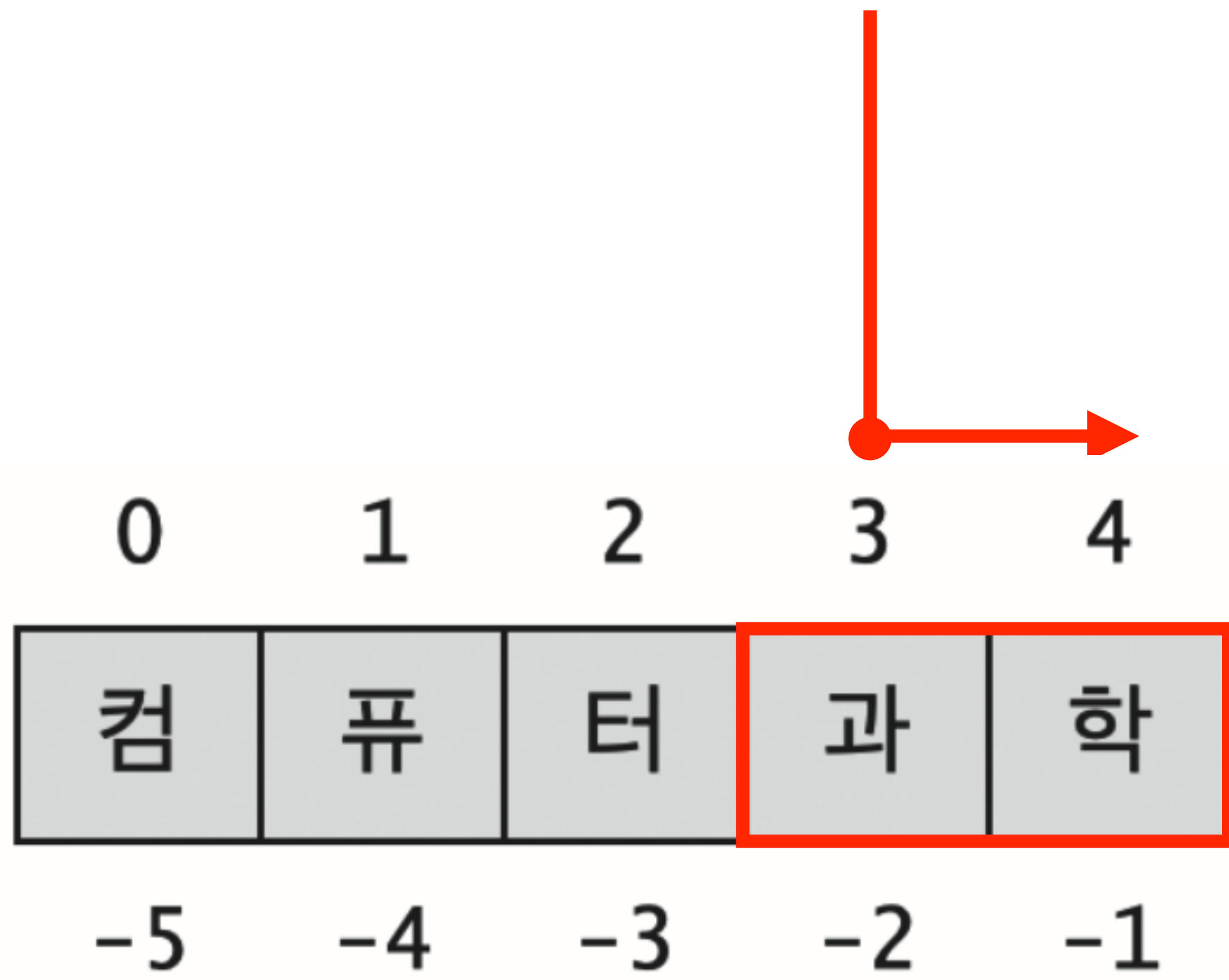
문자열 조각 복제

"컴퓨터과학" [3:]

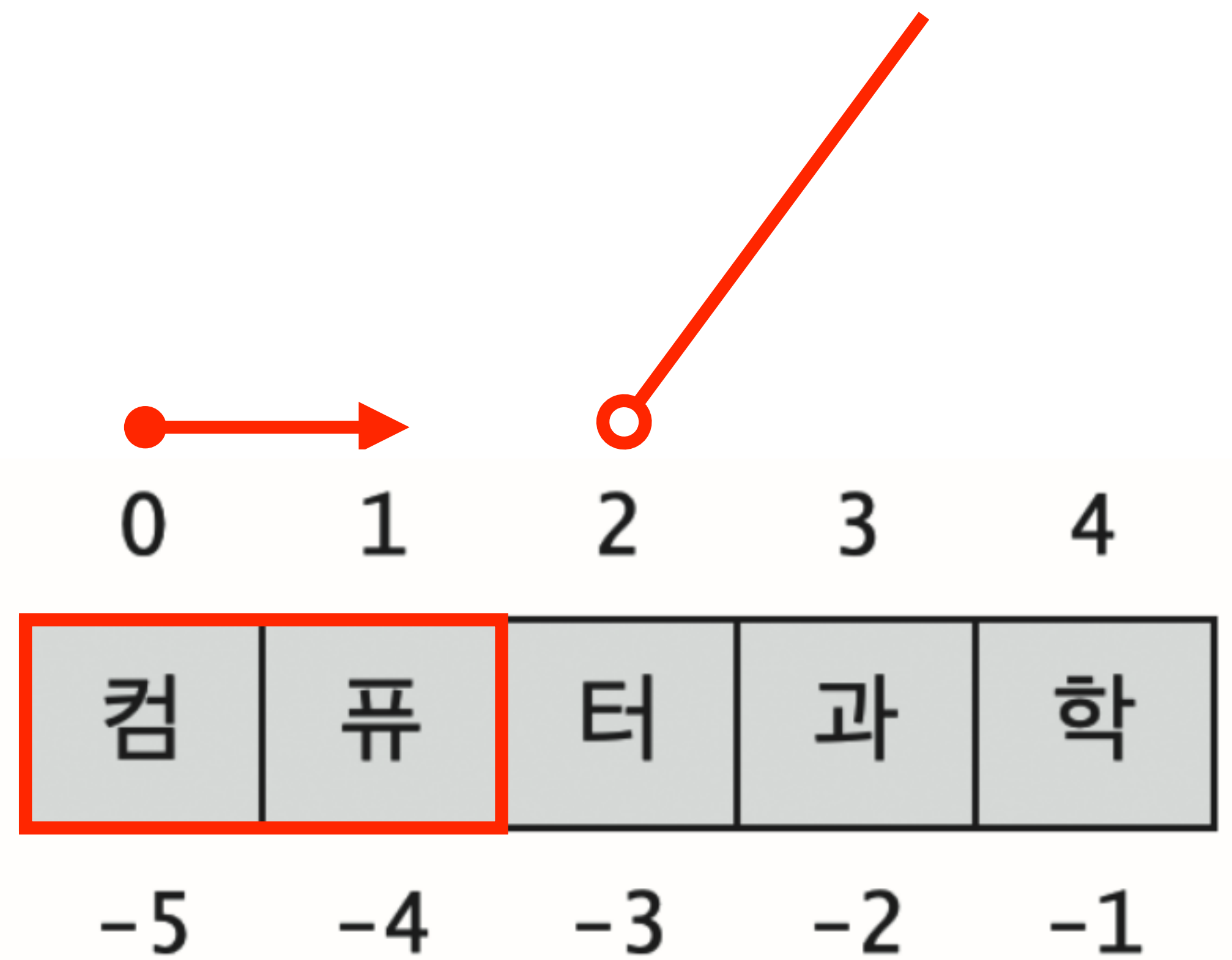


문자열 조각 복제

"컴퓨터과학" [3:]

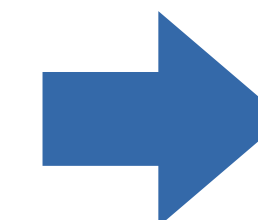
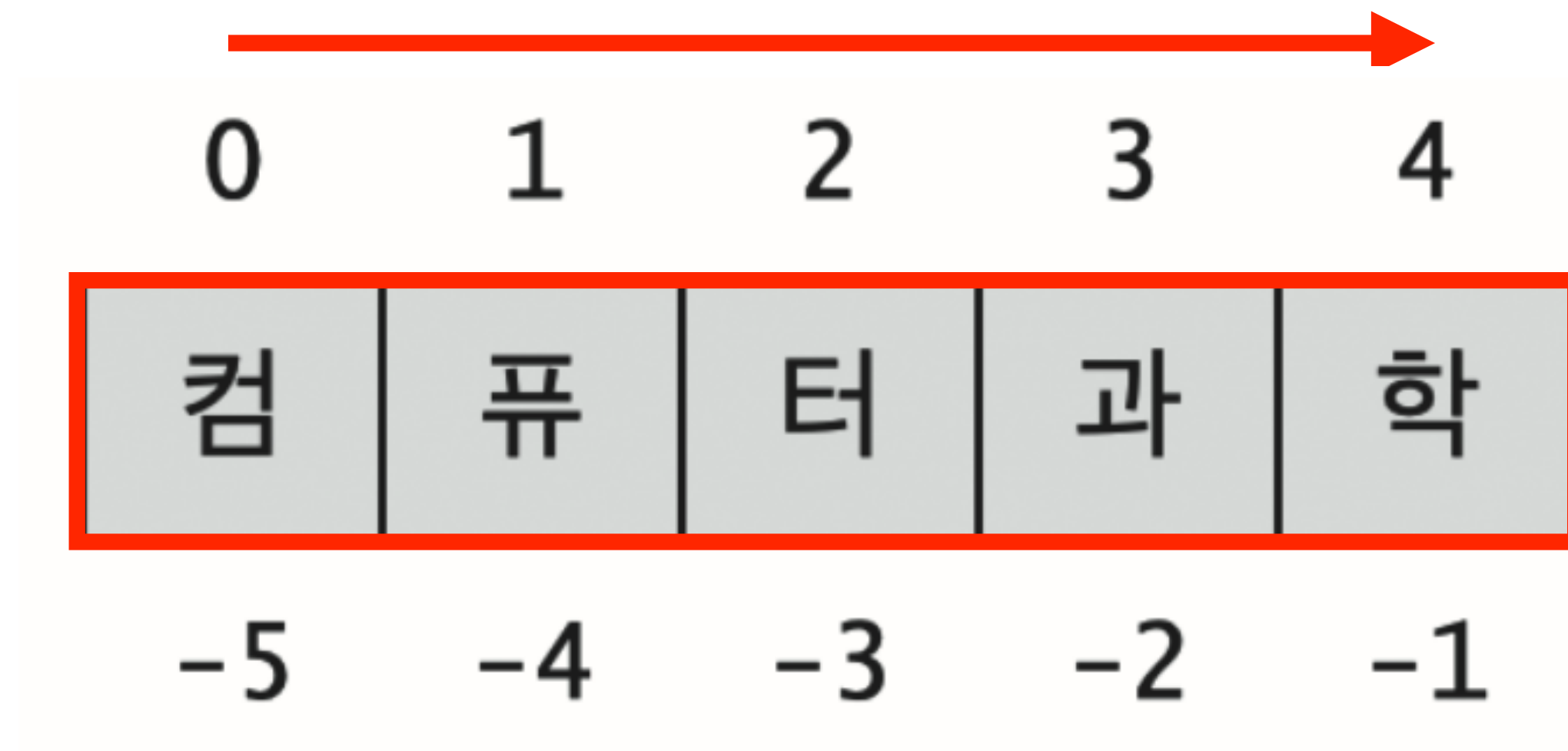


"컴퓨터과학" [:2]



문자열 복제

"컴퓨터과학"[:]



〈문자열〉

〈분리문자열〉



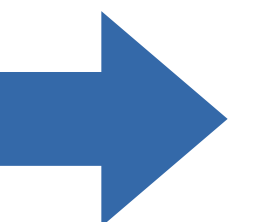
`s.partition(t)`

`"3.14159".partition(".")`



`("3", ".", "14159")`

튜플(tuple)



>>>>>>>>>> 제어 구조의 설계 원리를 중심으로 배우는 >>>>>>>>>>

프로그래밍의 정석 파이썬

도경구 재



pp.127~129



실습 3.9 정수 문자열 확인 함수 (음수 포함)



실습 3.10 실수 문자열 확인 함수 (음수 포함)

>>>>>>>>>> 제어 구조의 설계 원리를 중심으로 배우는 >>>>>>>>>>

프로그래밍의 정석

파이썬

도경구 지음



CHAPTER 3

제어 구조