

프로그래밍의 정석
파이썬

2

변수와 함수

2.1 변수 · 2.2 함수

CHAPTER 2

변수와 함수

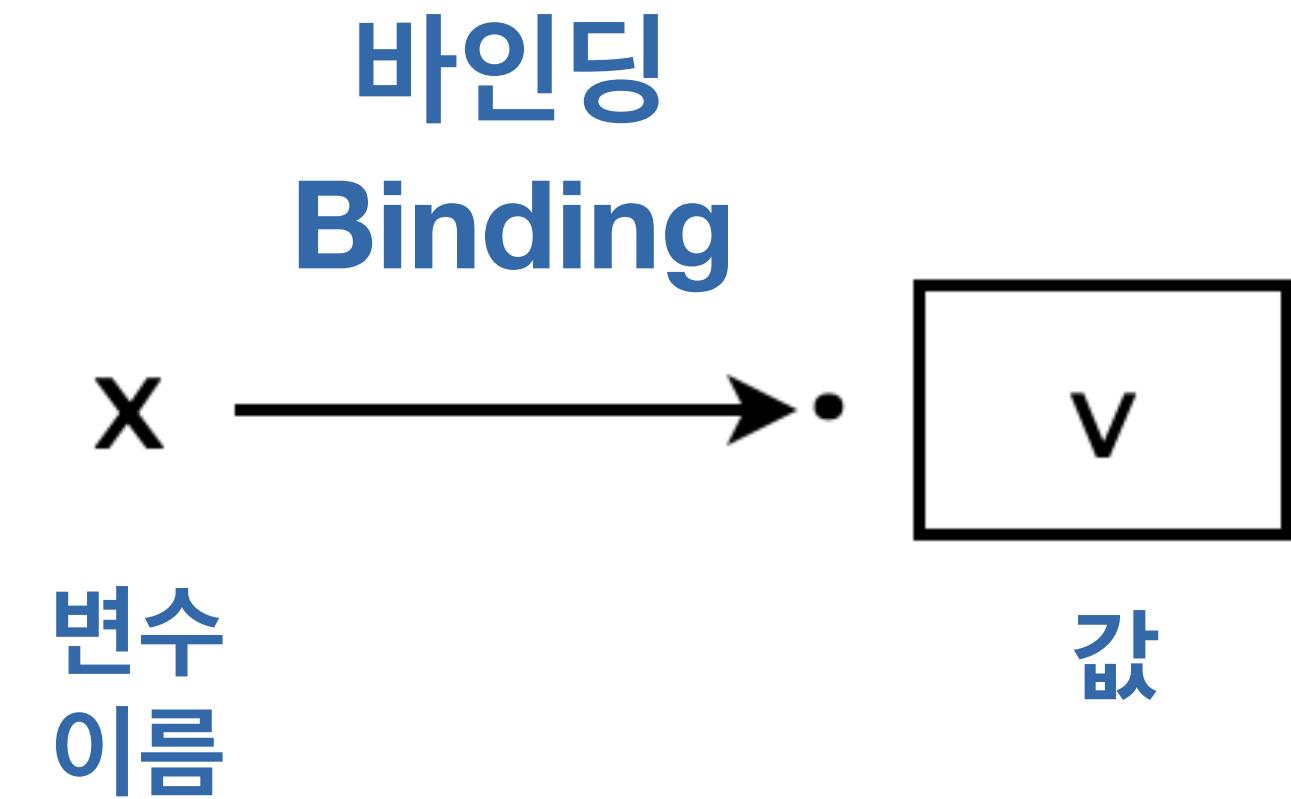
✓ 2.1 변수
2.2 함수

변수

Variable

네임스페이스 Namespace

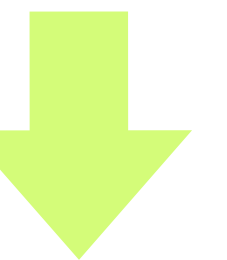
프로그램 실행 중에 생기는 계산 값을
추후 계산과정에서 두고두고 사용하기 위해서
지어두는 이름



지정문

Assignment

<변수> = <식>

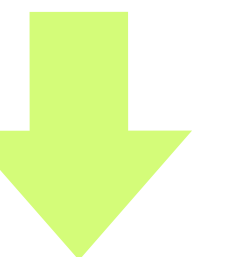


지정문

Assignment

<변수> = <식>

Expression



지정문

Assignment

<변수> = <식>

Variable

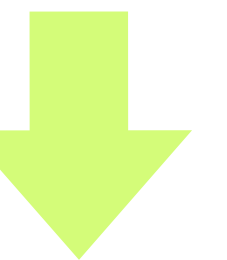
Expression

지정문

Assignment Statement

✓ $x = \underline{3 + 4}$

네임스페이스



지정문

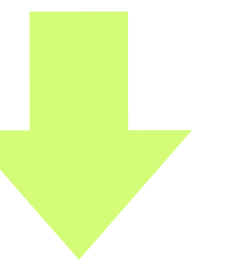
네임스페이스

Assignment Statement

✓ $x = \underline{3 + 4}$

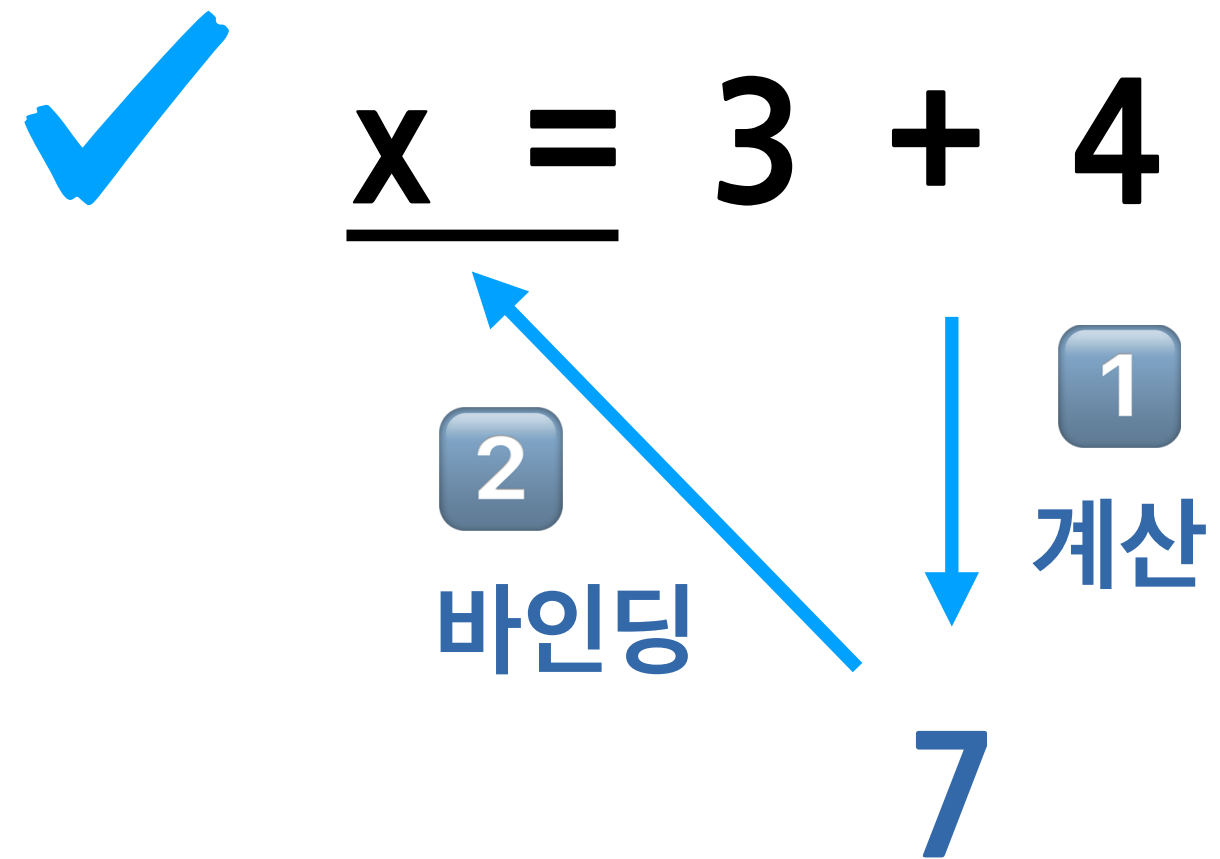
↓ 1
계산

7

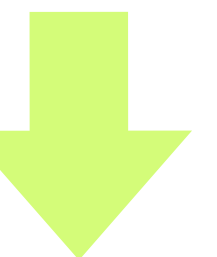
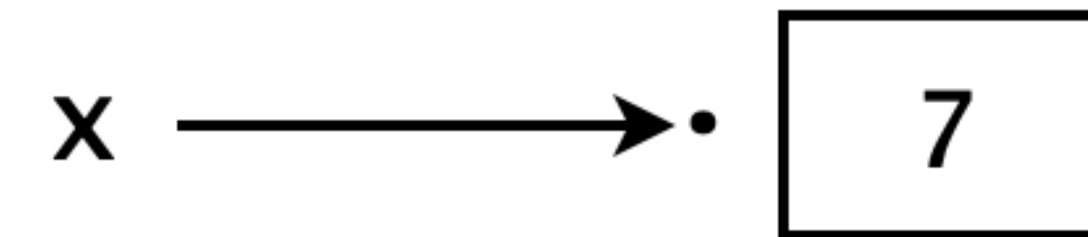


지정문

Assignment Statement



네임스페이스



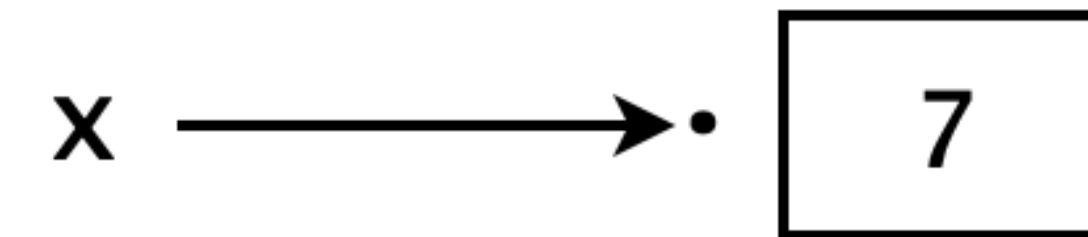
지정문

Assignment

$$x = 3 + 4$$

✓ $x = \underline{x + 2}$

네임스페이스



지정문

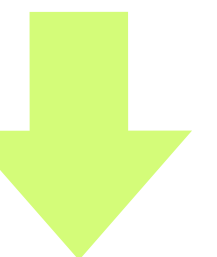
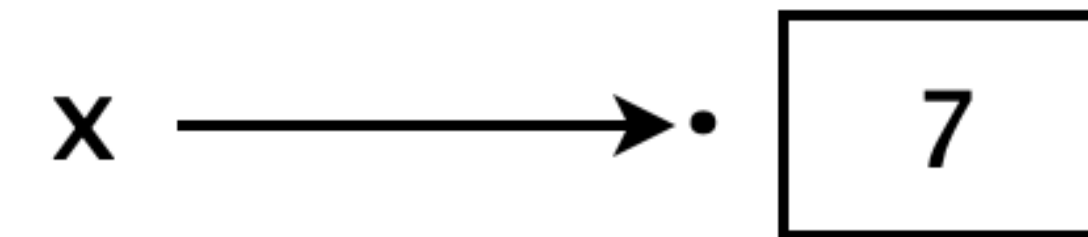
Assignment

$$x = 3 + 4$$

✓ $x = \underline{x + 2}$

↓ 1
계산
9

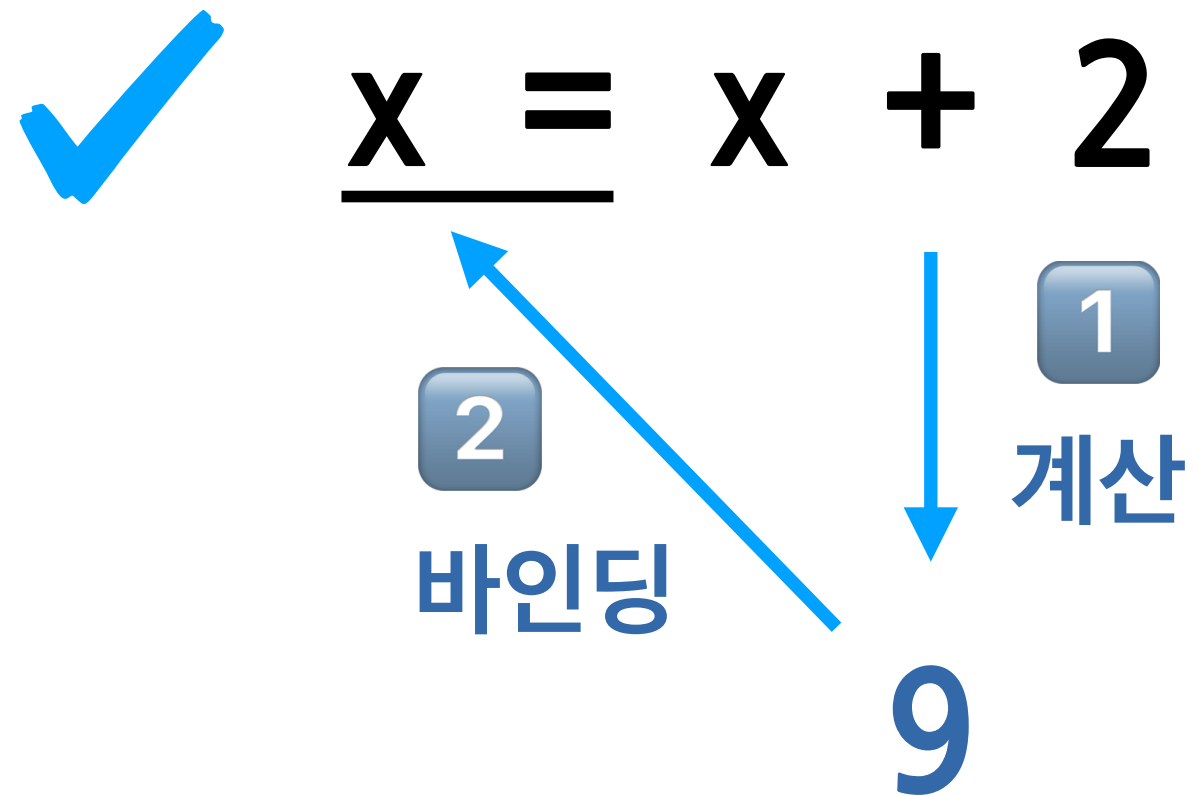
네임스페이스



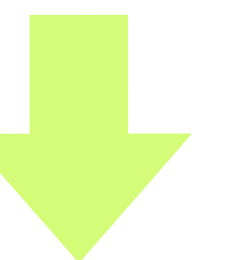
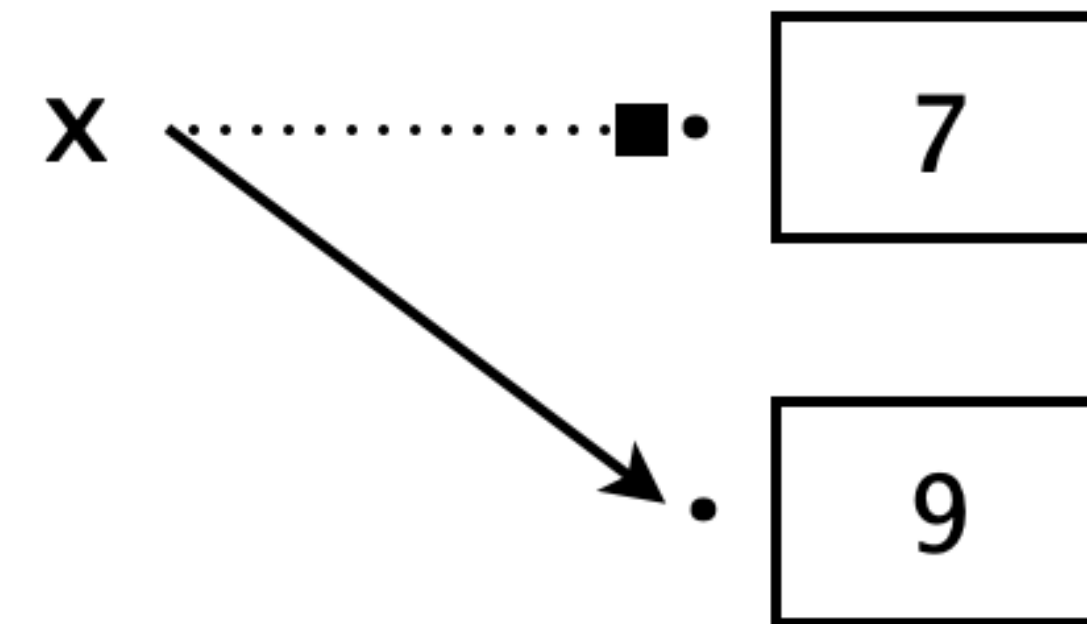
지정문

Assignment

$$x = 3 + 4$$



네임스페이스



지정문

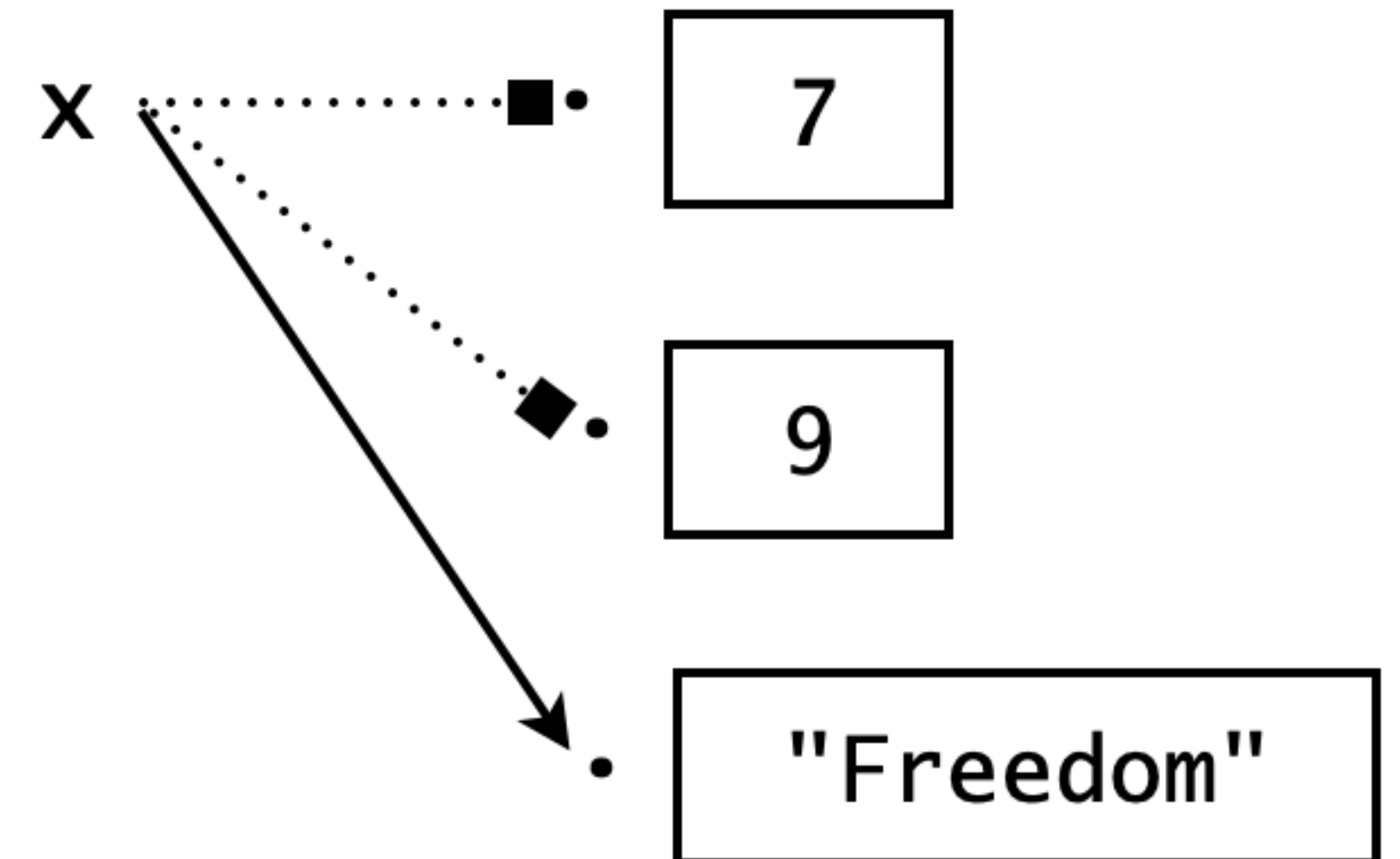
Assignment

$x = 3 + 4$

$x = x + 2$

✓ $x = \text{"Freedom"}$

네임스페이스



Python

Dynamic Binding
동적 바인딩

C

Java

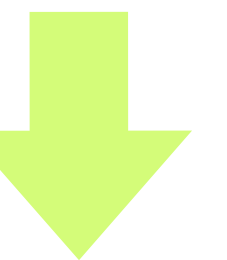
Static Binding
정적 바인딩

Static Binding

정적 바인딩

✓ `int x`

네임스페이스

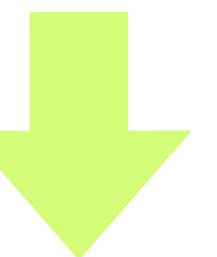
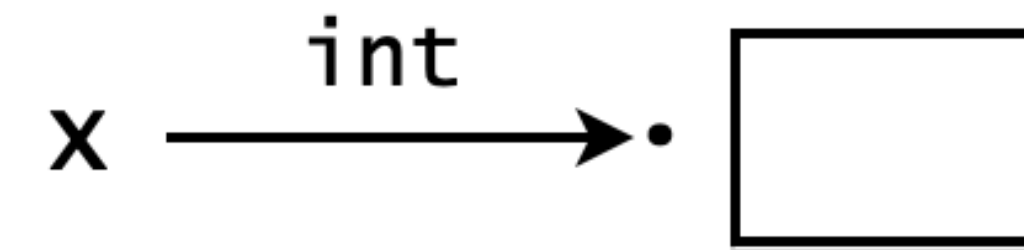


Static Binding

정적 바인딩

✓ `int x`

네임스페이스



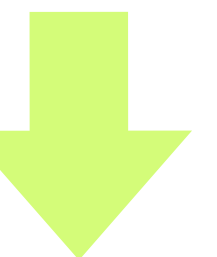
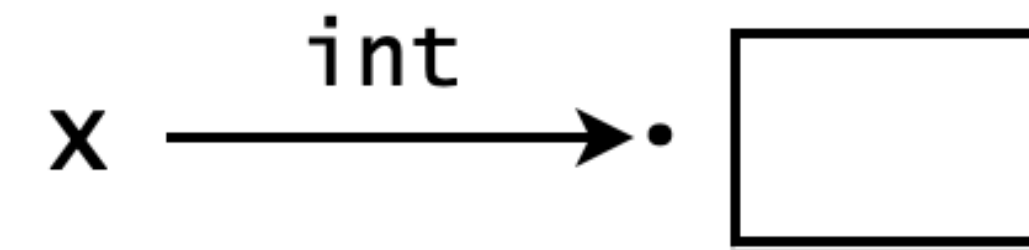
Static Binding

정적 바인딩

`int x`

✓ `x = 3 + 4`

네임스페이스



Static Binding

정적 바인딩

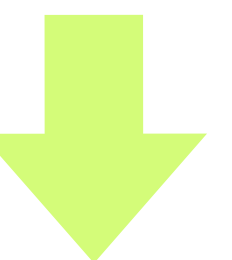
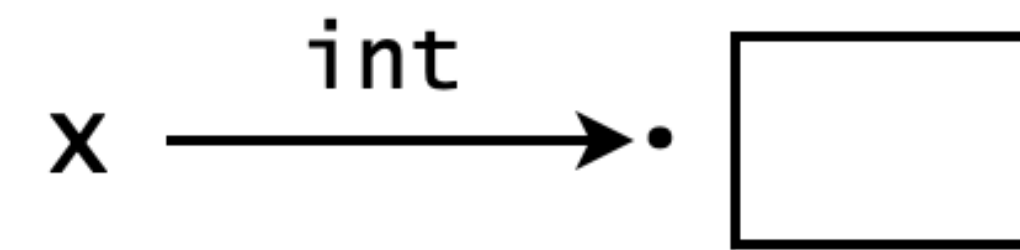
`int x`



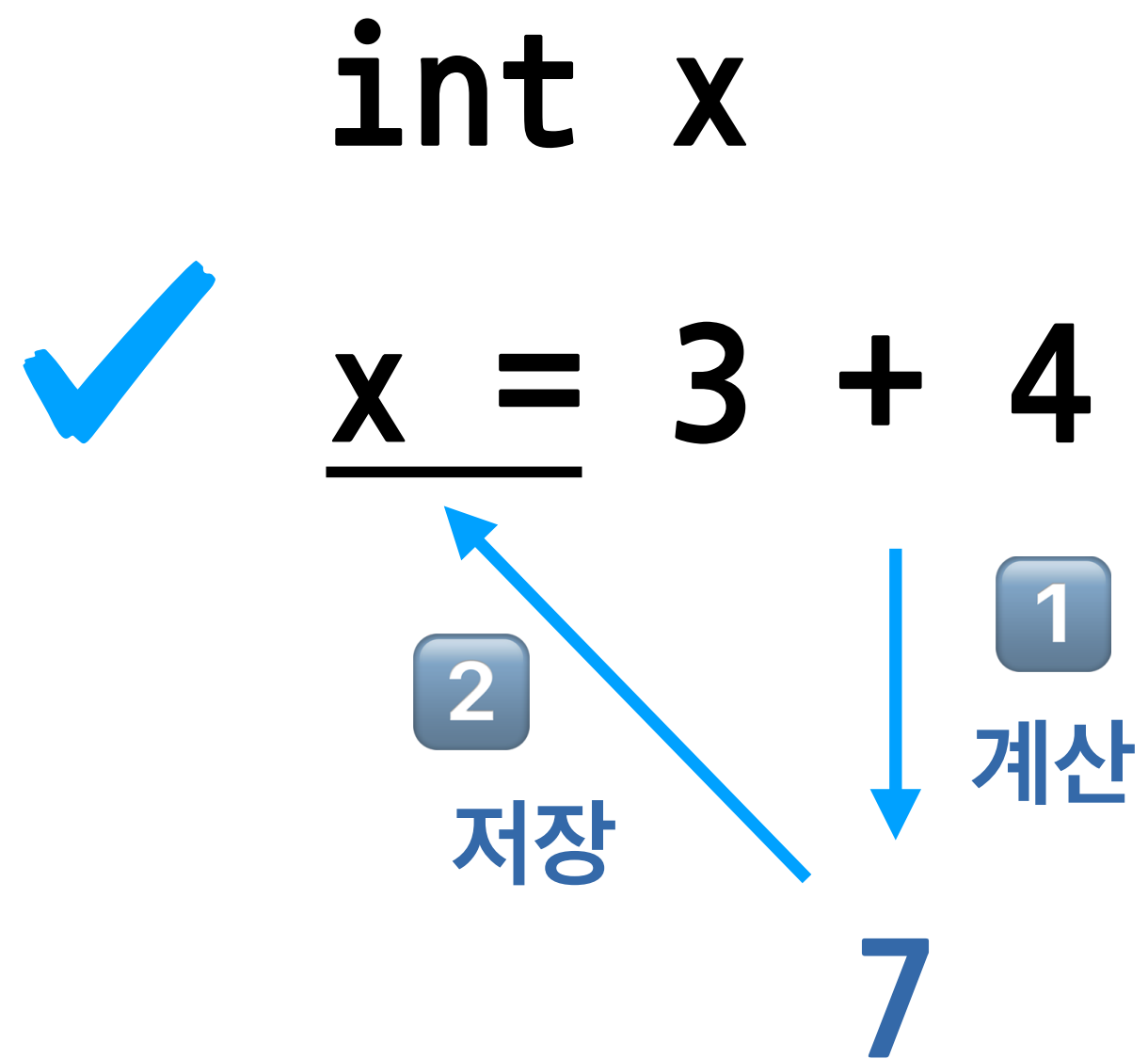
`x = 3 + 4`

↓ 1
계산
7

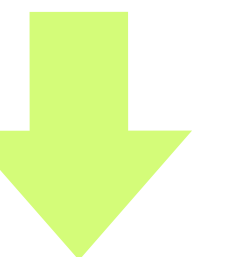
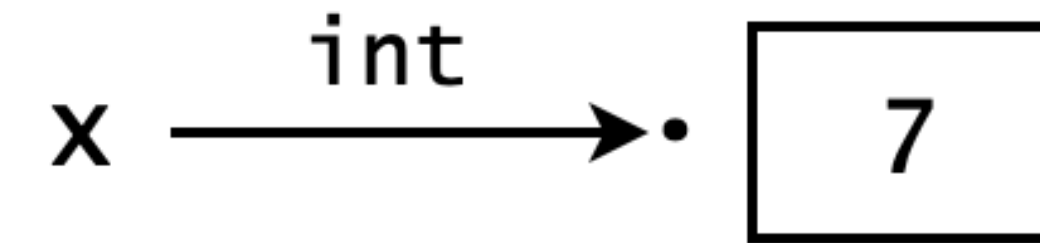
네임스페이스



Static Binding 정적 바인딩



네임스페이스



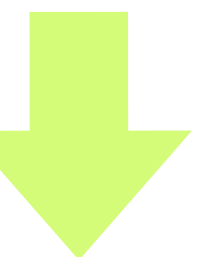
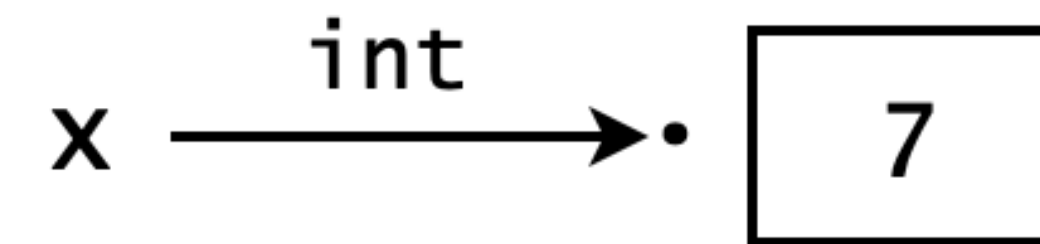
Static Binding 정적 바인딩

`int x`

`x = 3 + 4`

✓ `x = x + 2`

네임스페이스



Static Binding 정적 바인딩

`int x`

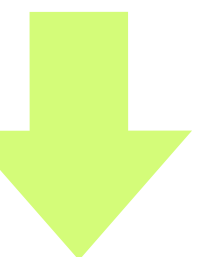
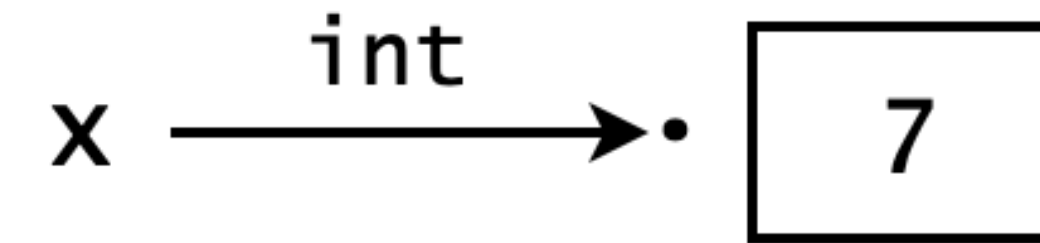
`x = 3 + 4`

✓ `x = x + 2`

↓ 1 계산

9

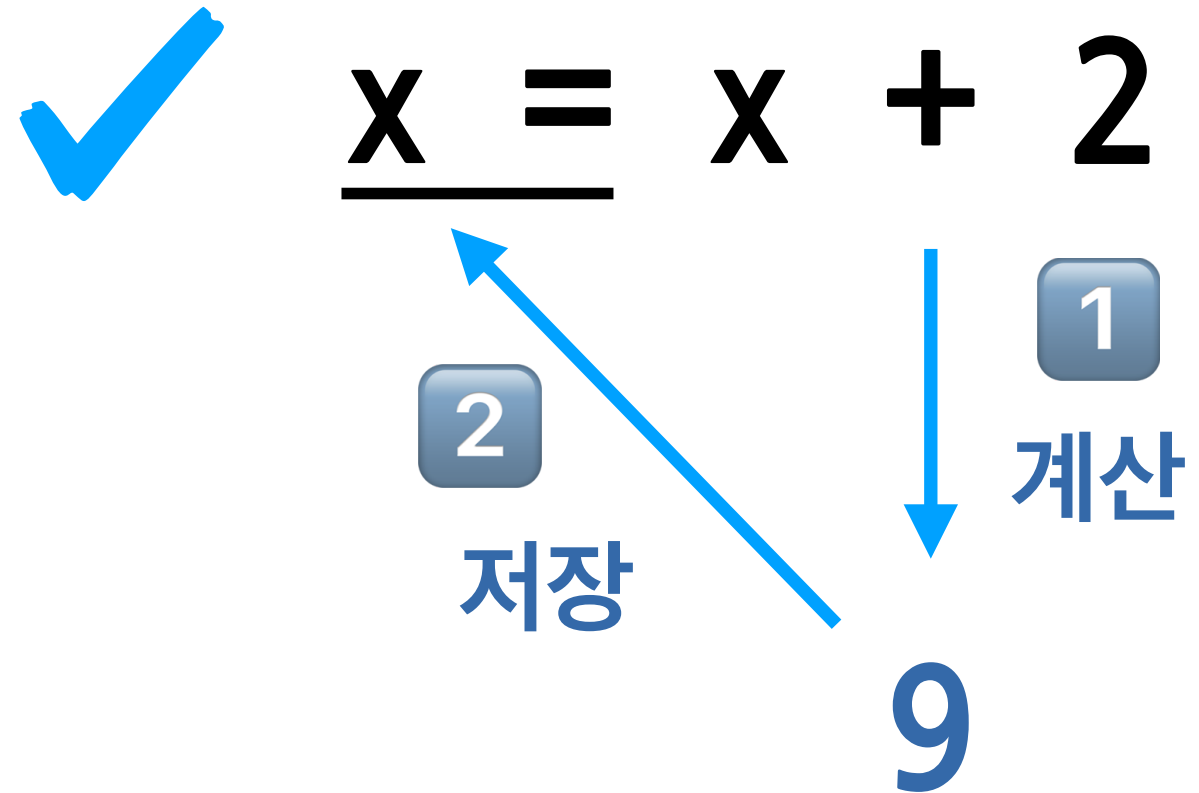
네임스페이스



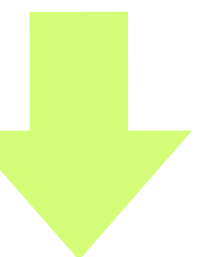
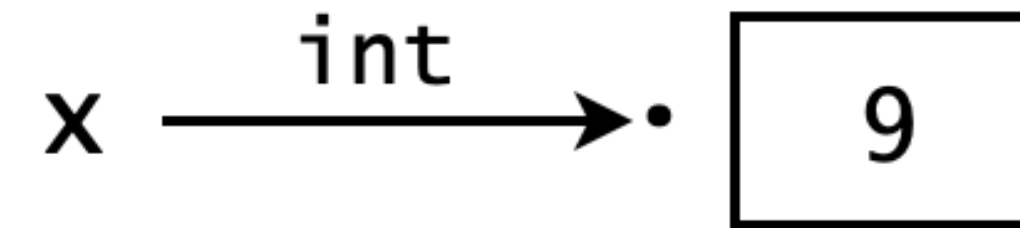
Static Binding 정적 바인딩

int x

x = 3 + 4



네임스페이스



Static Binding

정적 바인딩

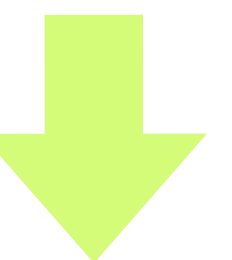
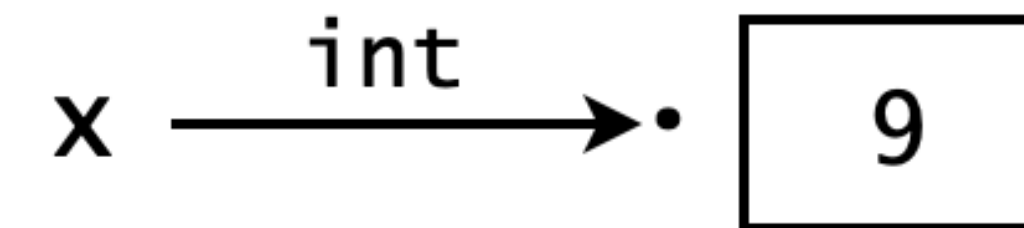
`int x`

`x = 3 + 4`

`x = x + 2`

✓ `x = "Freedom"`

네임스페이스



Static Binding 정적 바인딩

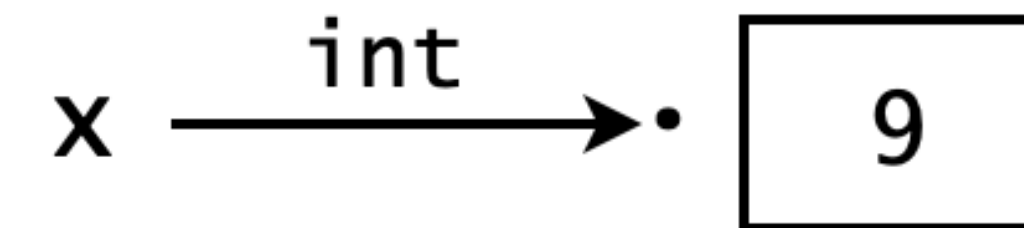
`int x`

`x = 3 + 4`

`x = x + 2`

✓ `x = "Freedom"` ✗

네임스페이스



지정문

Assignment

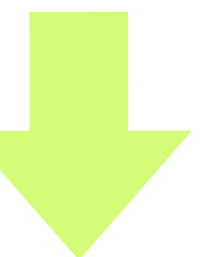
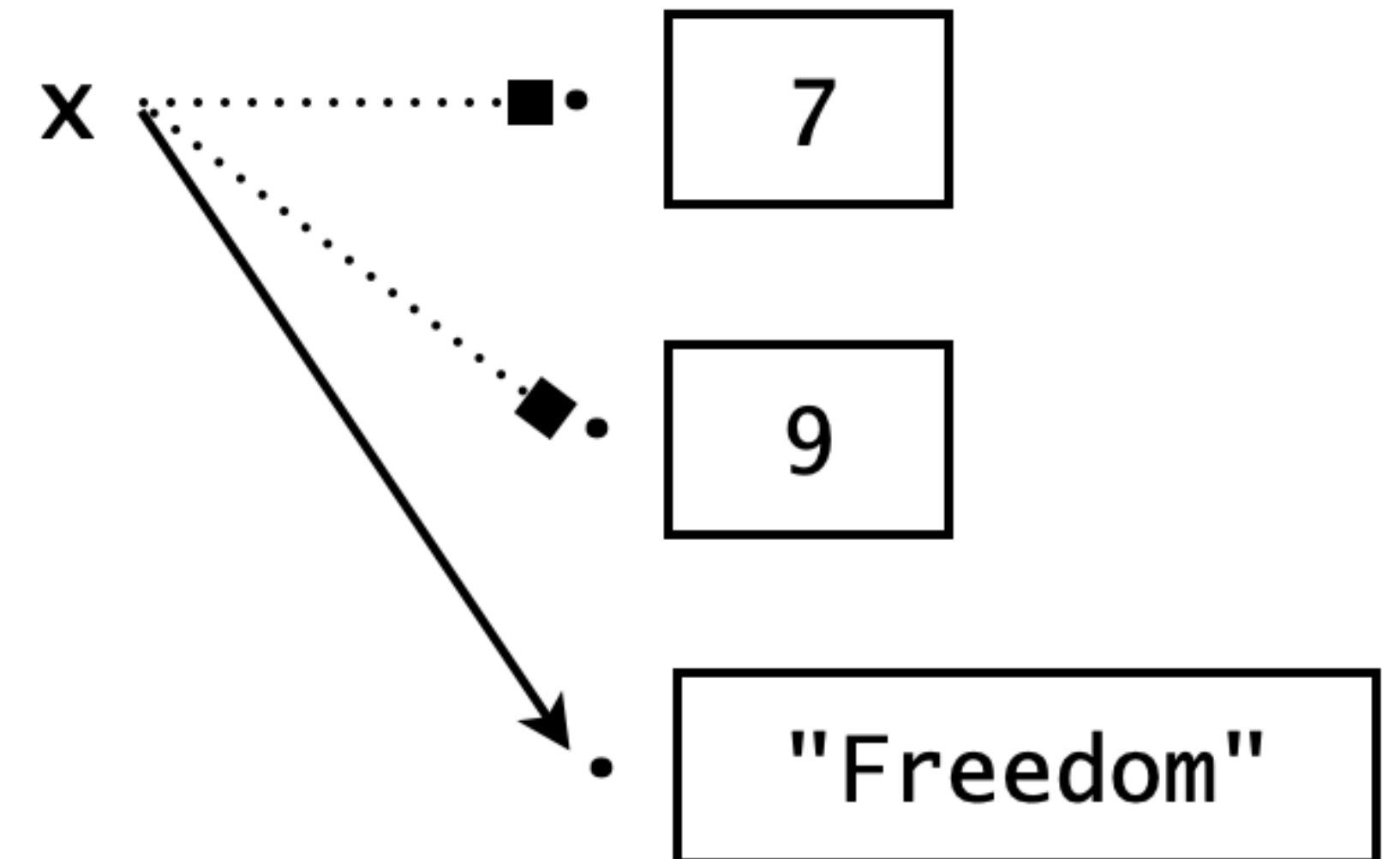
`x = 3 + 4`

`x = x + 2`

`x = "Freedom"`

✓ `print(x)`

네임스페이스



지정문

Assignment

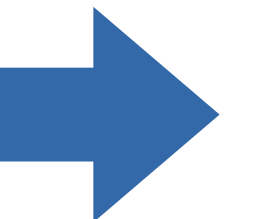
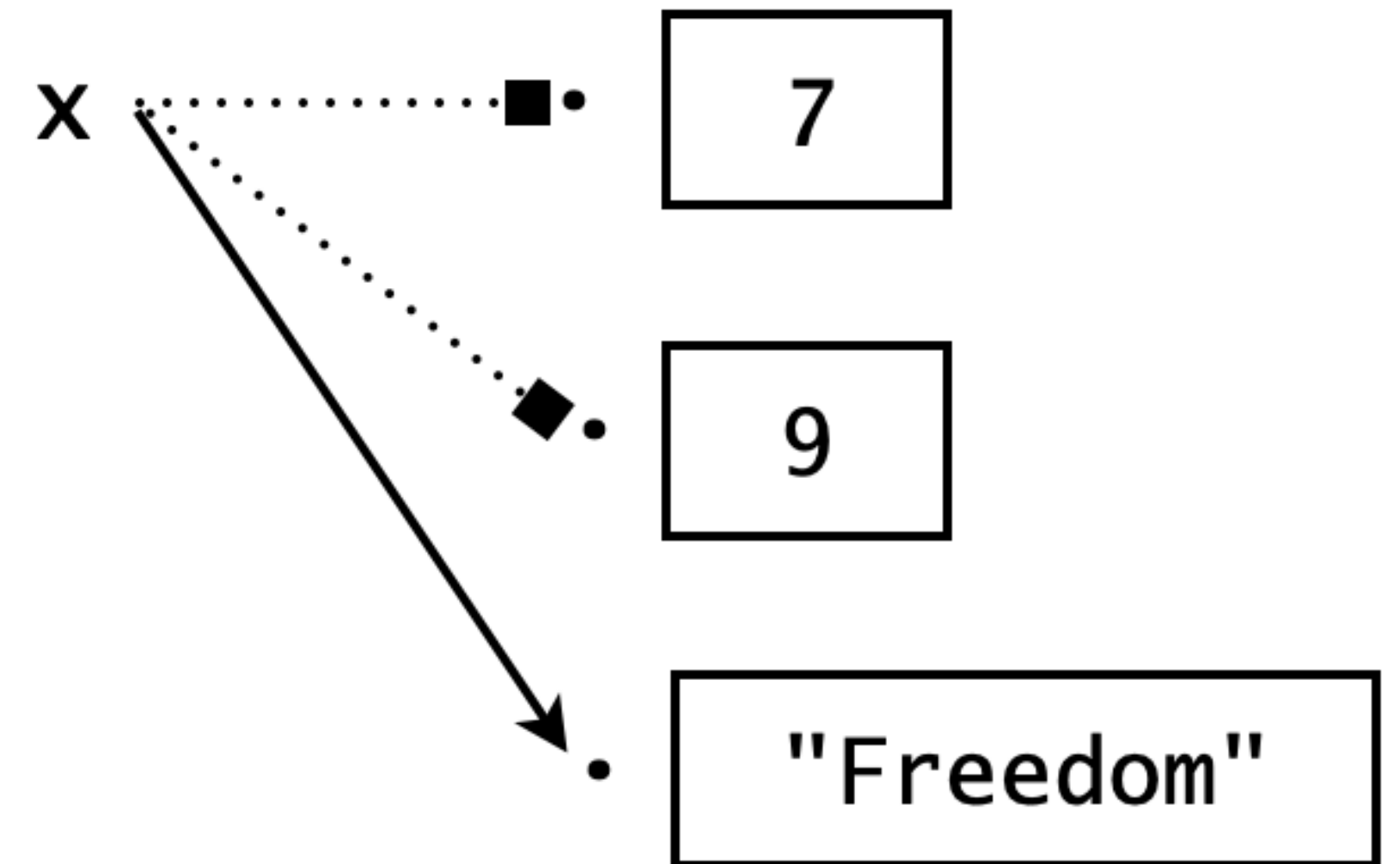
`x = 3 + 4`

`x = x + 2`

`x = "Freedom"`

✓ `print(pooh)` ✗

네임스페이스



변수 이름 짓기

아래 문자들의 조합

소문자

a-z

대문자

A-Z

숫자

0-9

밑줄문자

_

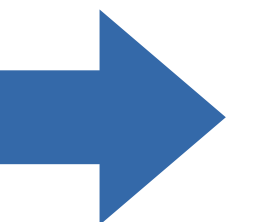
예외 : 숫자로 시작할 수 없음

susieQ

korea1st

python_programming

1stKorea



변수 이름 짓기

아래 문자들의 조합	
소문자	a-z
대문자	A-Z
숫자	0-9
밑줄문자	_
예외 : 숫자로 시작할 수 없음	

- 값의 특징을 잘 대변해주는 명사 또는 명사구를 고를 것
- 나름의 작명 규칙을 정하고 일관성을 유지할 것
- 관습을 따를 것 (일반 변수는 소문자로 시작)

susieQ

korea1st

python_programming

1stKorea

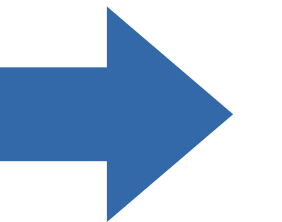
프로그램의 가독성.

파이썬 코딩 컨벤션

Python coding convention.

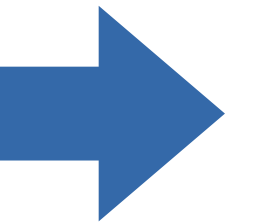
원의 면적 구하기

$$\pi \times r^2$$



표준 라이브러리

`math`



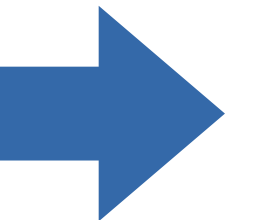
통합개발환경

IDE

Integrated Development Environment

IDLE의 편집기 활용

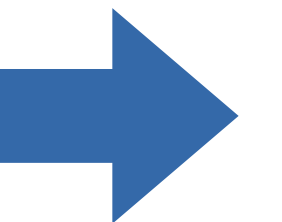
프로그램을 파일에 저장하여 한꺼번에 실행



표준 입력

Standard Input

```
x = input()
```



반올림

`round()`



실습 2.1 키보드 입력과 반올림

원의 면적 계산 프로그램의 계산 결과를 소수 첫째 자리 아래에서 반올림하여 프린트하도록 수정하자. 즉, 실행창에서 3을 입력하여 실행하면, 다음과 같이 실행창에 출력해야 한다.

```
The area of a circle with radius 3.0 is 28.3
```



실습 2.2 동전 합산 서비스

동전 지갑에 들어있는 동전의 총액을 계산하는 프로그램을 만들자. 액면가가 큰 동전부터 순서대로 보유 동전의 개수를 입력받도록 한다. 이 동전합산 서비스를 종료하면 실행 결과가 다음과 같은 형식으로 실행창에 남아 있어야 한다.

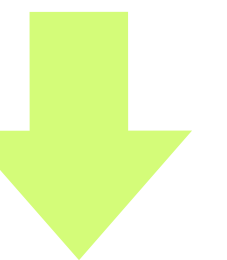
```
Enter the number of coins you have.  
500 won? 4  
100 won? 2  
50 won? 3  
10 won? 4  
You have 2390 won in total.
```

여기서 둘째 줄부터 다섯째 줄까지 있는 4, 2, 3, 4는 모두 키보드 입력이고, 나머지는 프로그램이 프린트한 것이다. 띄어쓰기도 똑같이 프린트하도록 해야 함에 주의하자.

지정문의 실행 순서

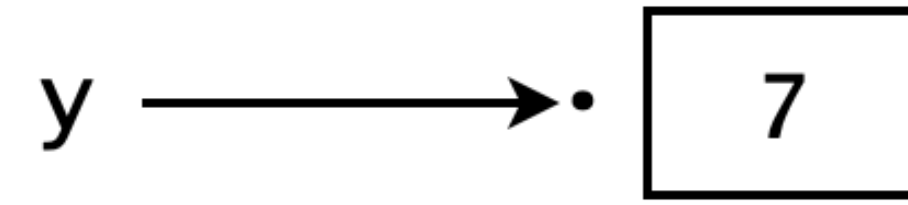
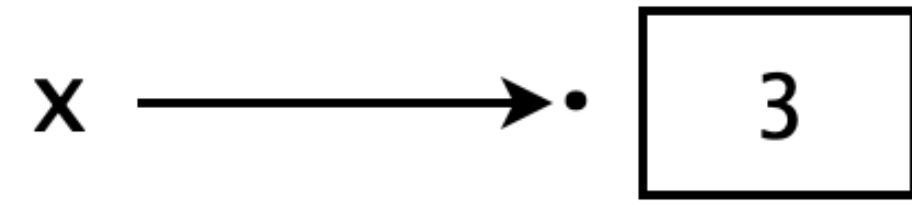
$x = 3$

$y = 7$

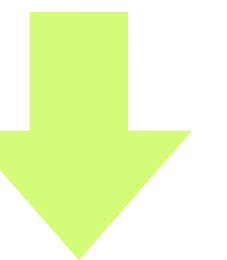


지정문의 실행 순서

✓ $x = 3$

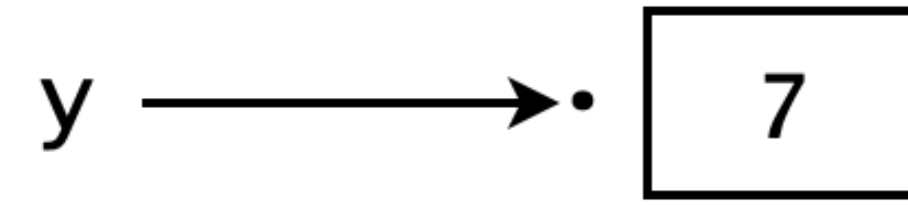
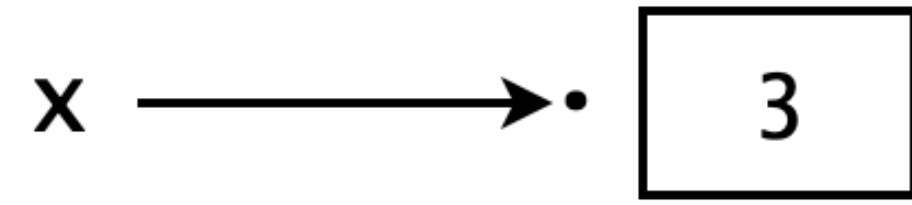


✓ $y = 7$



지정문의 실행 순서

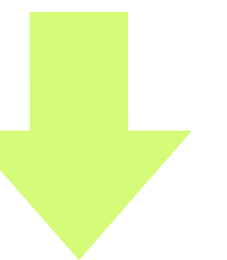
$$x = 3$$



$$y = 7$$


$$x = \underline{y}$$

$$y = x$$



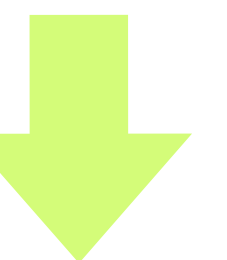
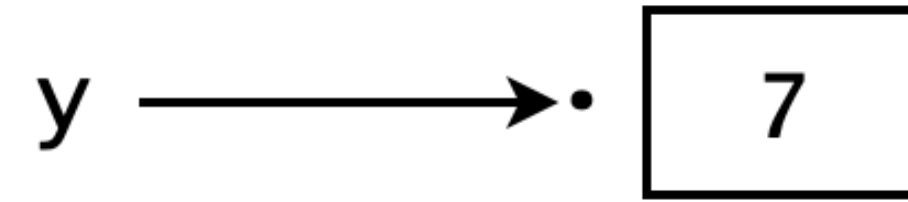
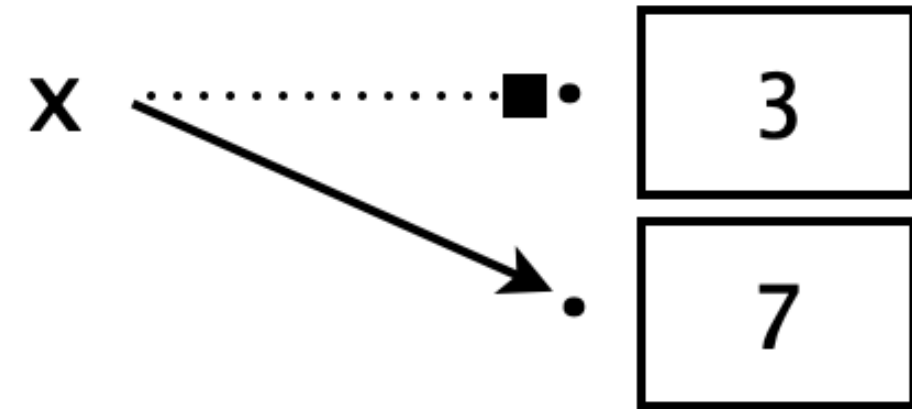
지정문의 실행 순서

$x = 3$

$y = 7$

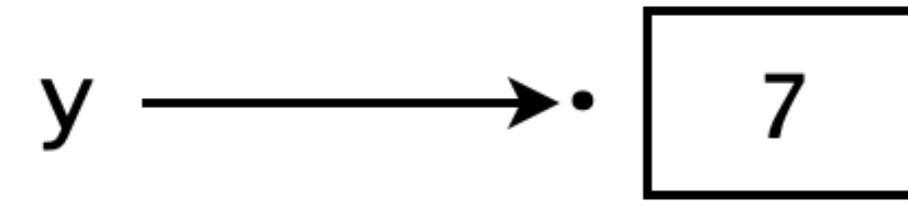
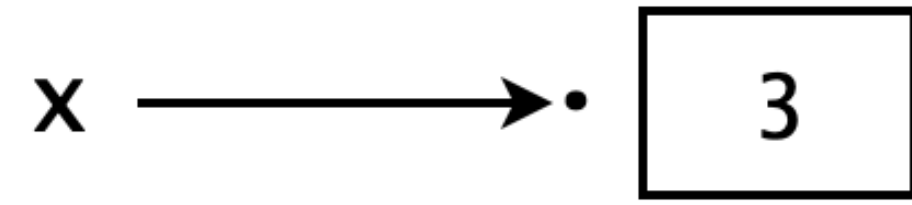
✓ $x = y$

$y = x$




지정문의 실행 순서

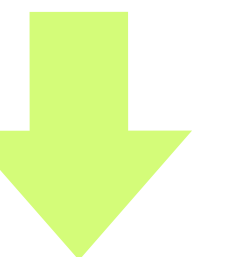
$$x = 3$$



$$y = 7$$

~~$$x = y$$~~


$$y = \underline{x}$$



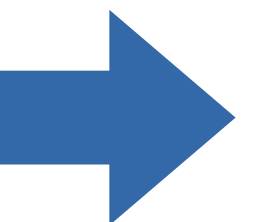
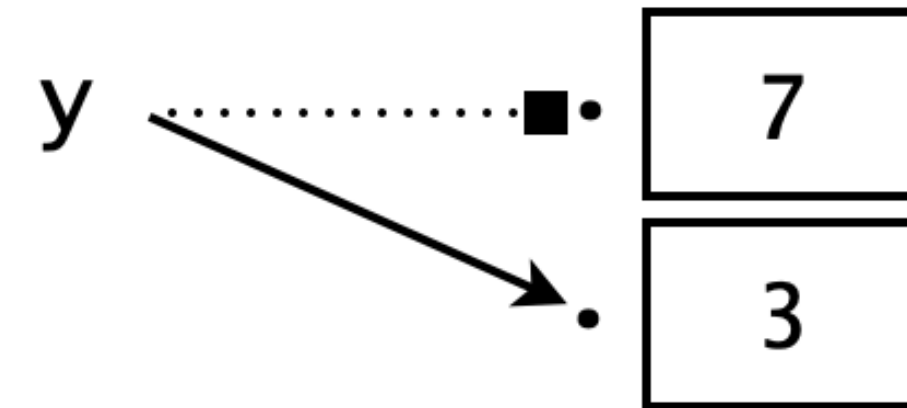
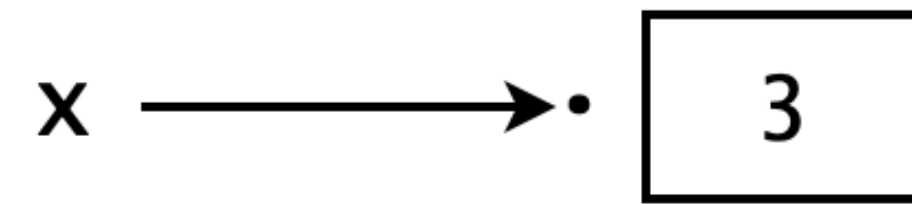
지정문의 실행 순서

$x = 3$

$y = 7$

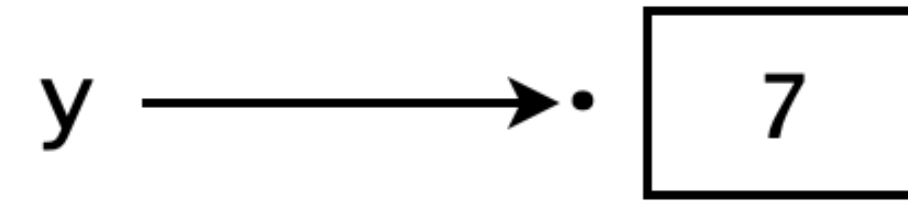
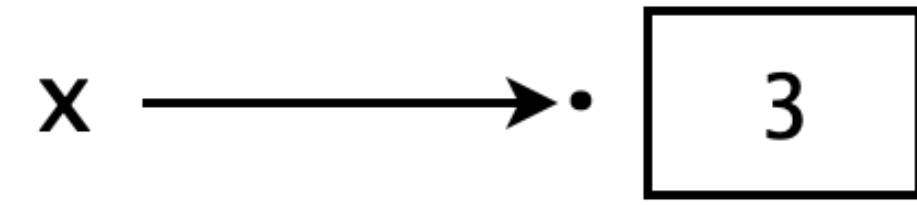
~~$x = y$~~

✓ $y = x$



지정문의 실행 순서

✓ $x = 3$

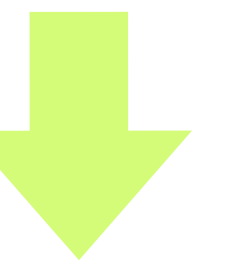


✓ $y = 7$

$t = x$

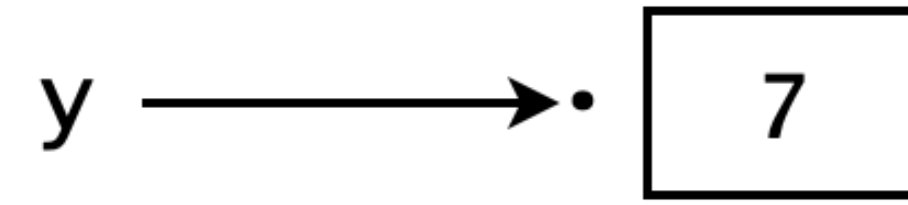
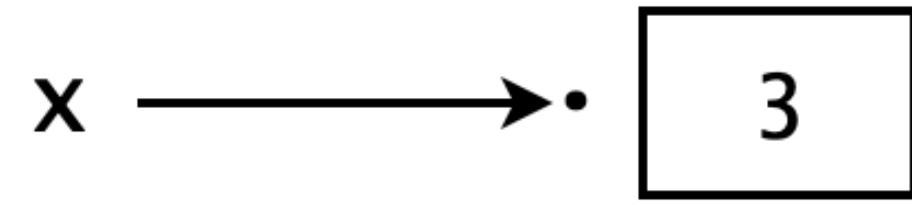
$x = y$

$y = t$




지정문의 실행 순서

$$x = 3$$

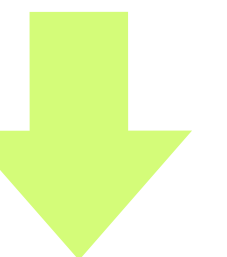


$$y = 7$$


$$t = x$$

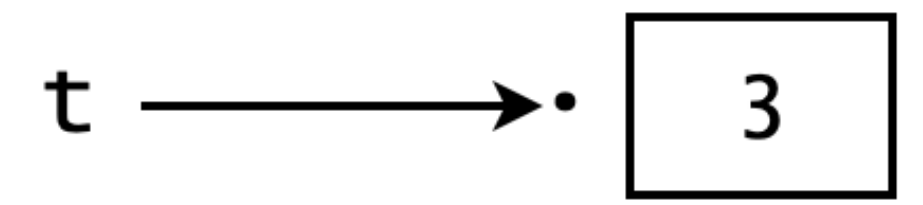
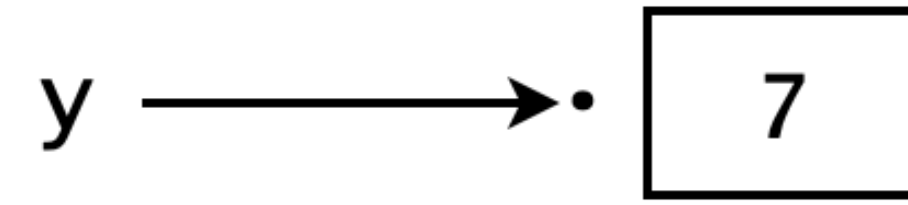
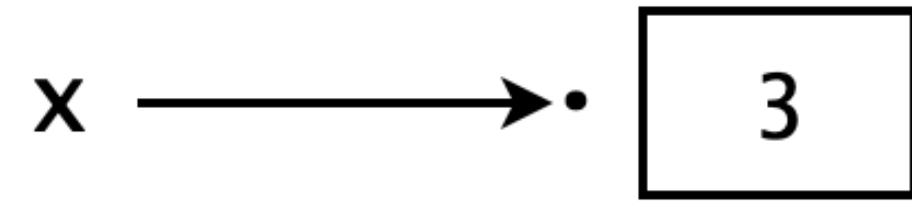
$$x = y$$

$$y = t$$



지정문의 실행 순서

$x = 3$

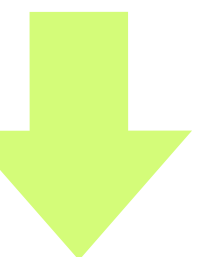


$y = 7$

✓ $t = x$

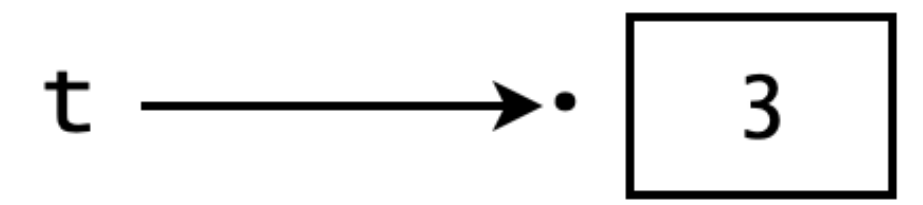
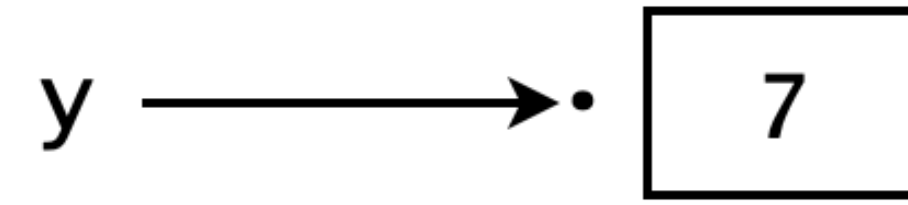
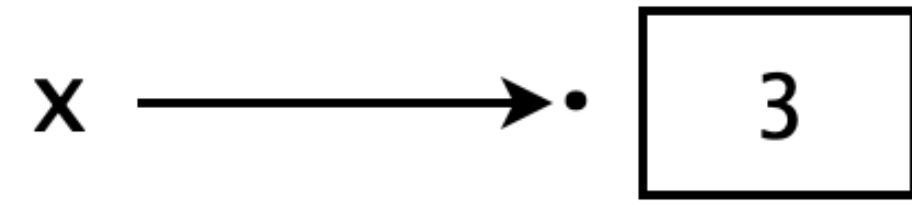
$x = y$

$y = t$




지정문의 실행 순서

$$x = 3$$

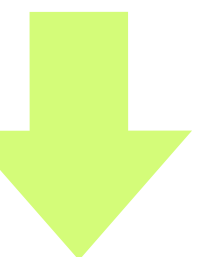


$$y = 7$$

$$t = x$$


$$x = y$$

$$y = t$$



지정문의 실행 순서

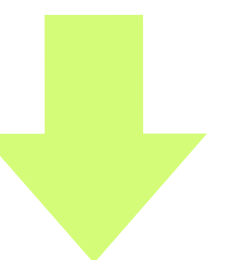
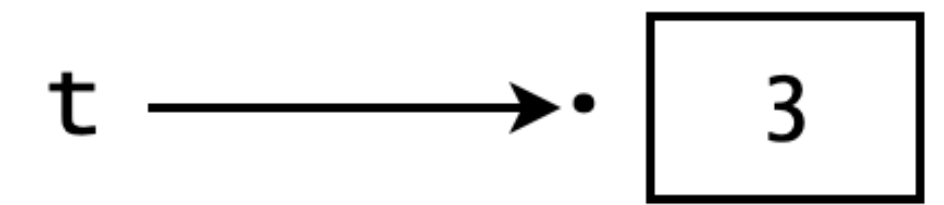
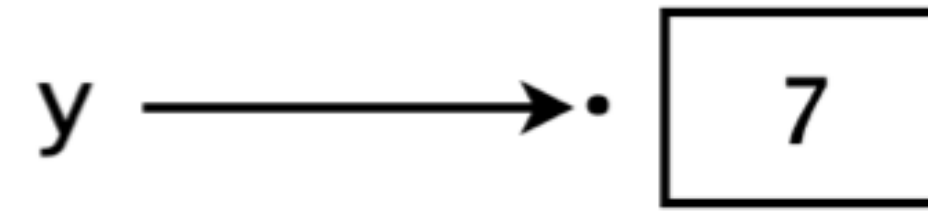
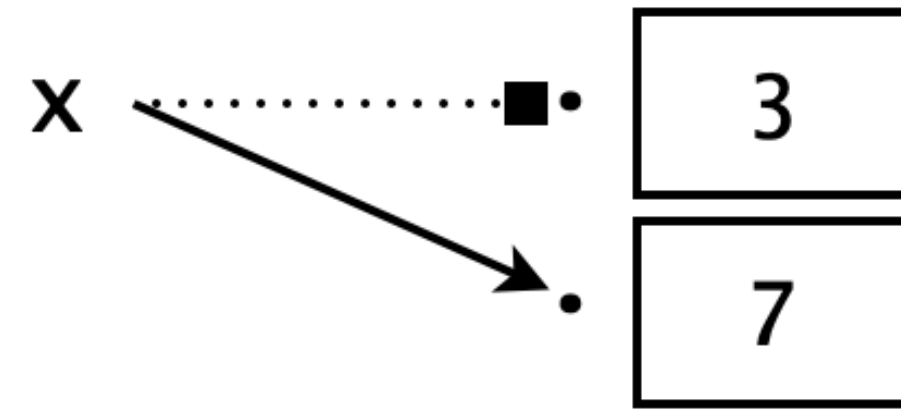
$x = 3$

$y = 7$

$t = x$

✓ $x = y$

$y = t$



지정문의 실행 순서

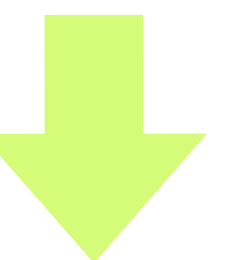
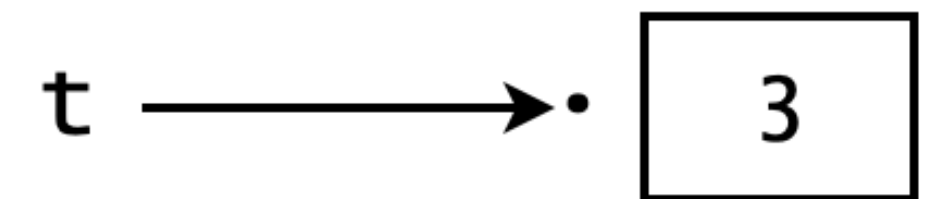
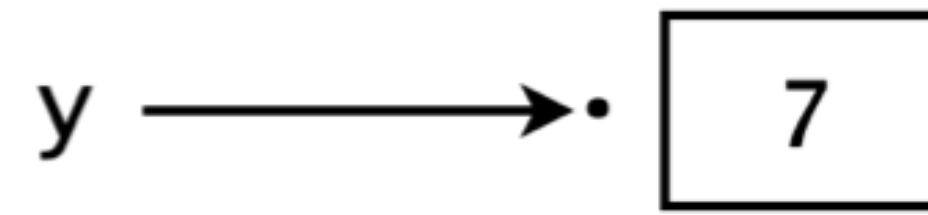
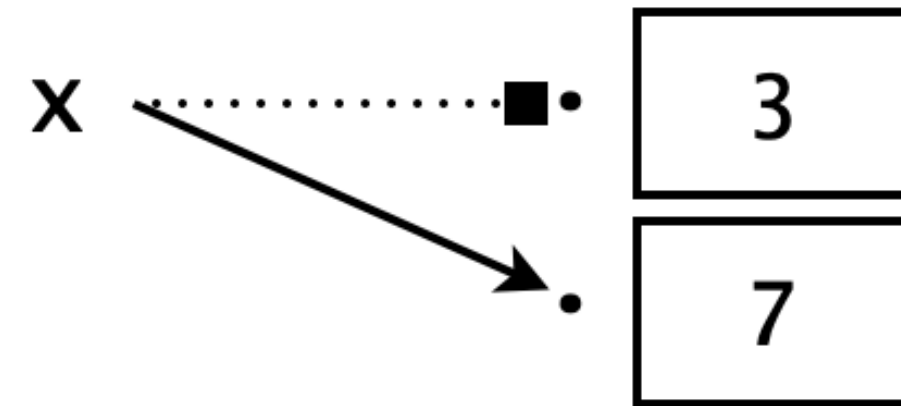
$x = 3$

$y = 7$

$t = x$

$x = y$

✓ $y = t$



지정문의 실행 순서

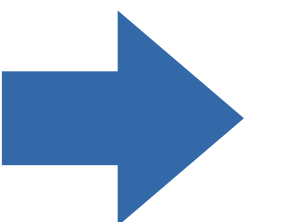
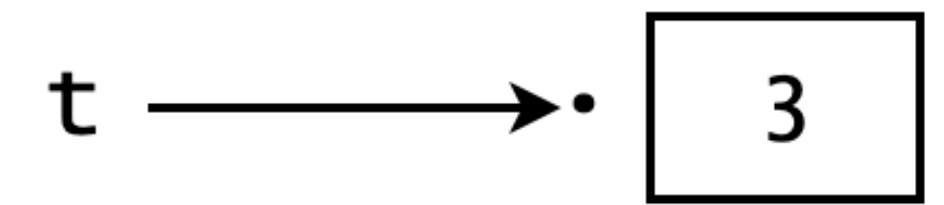
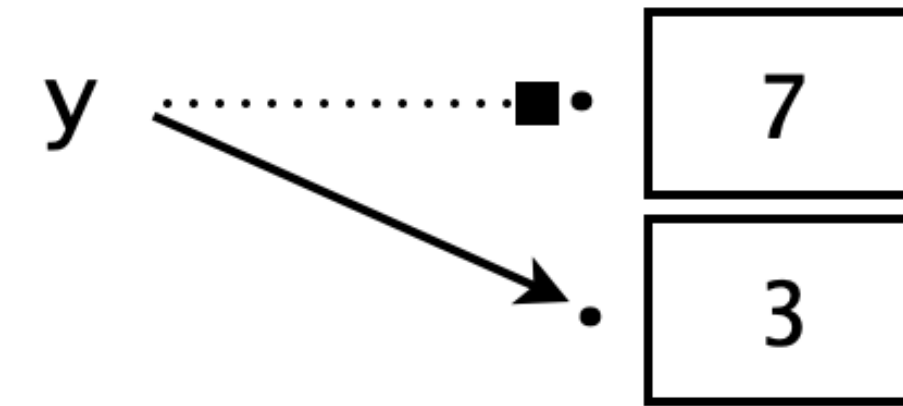
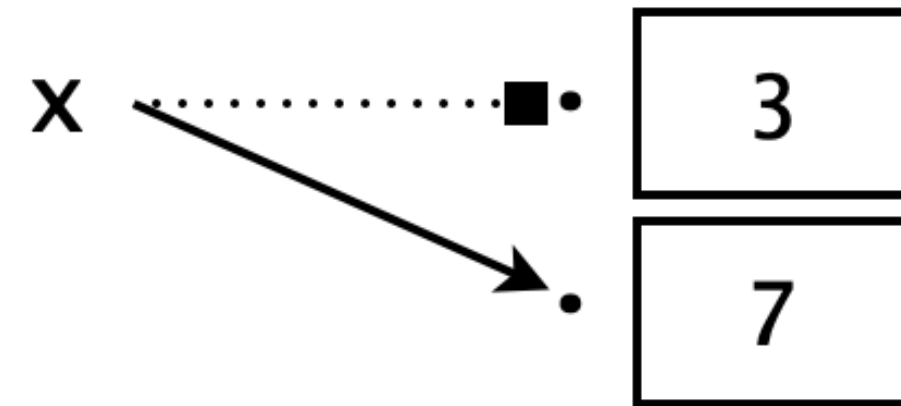
$x = 3$

$y = 7$

$t = x$

$x = y$

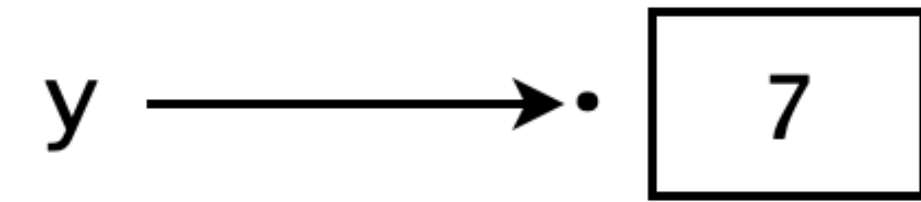
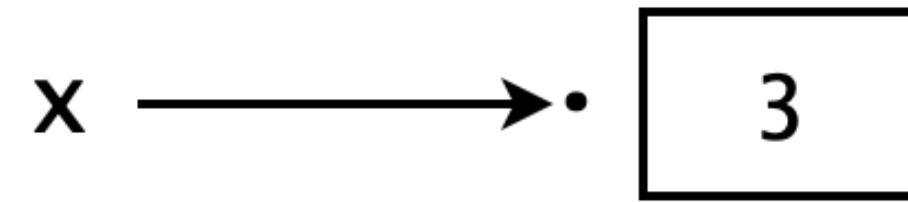
✓ $y = t$



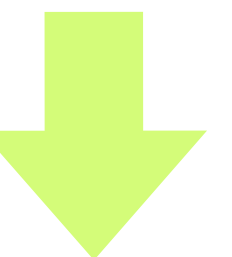
동시 지정



$x, y = 3, 7$

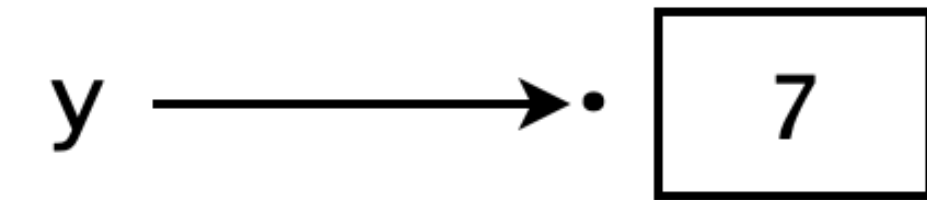
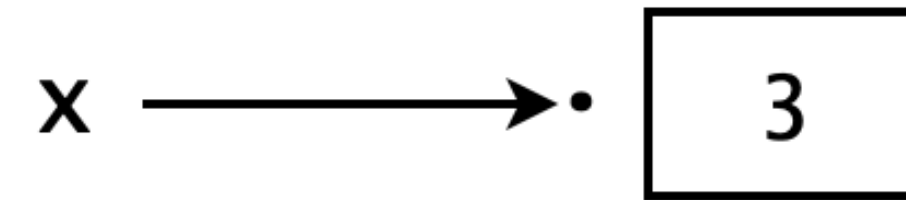


$x, y = y, x$

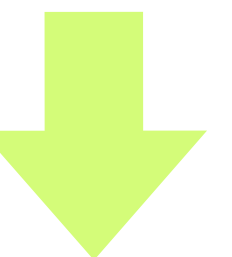


동시 지정

$x, y = 3, 7$

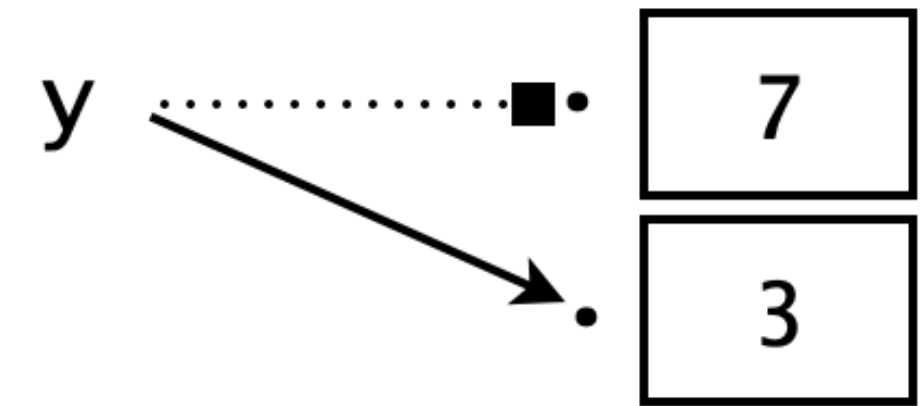
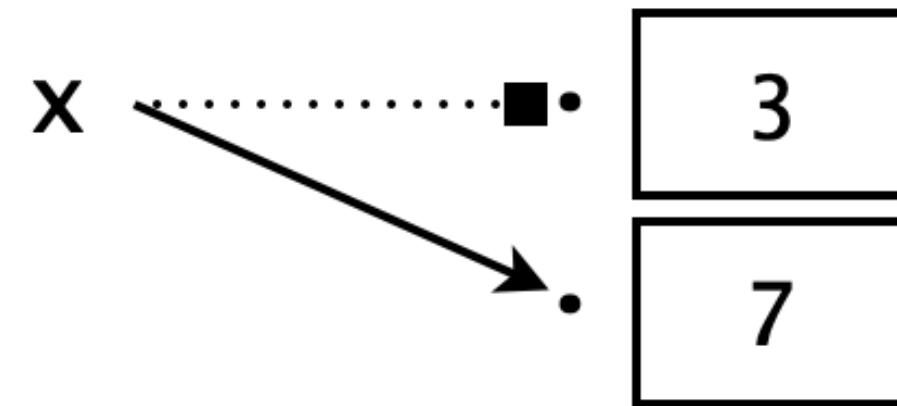


$x, y = y, x$

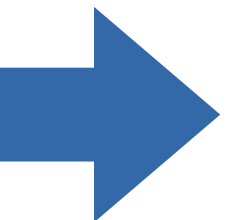


동시 지정

$x, y = 3, 7$



✓ $x, y = y, x$



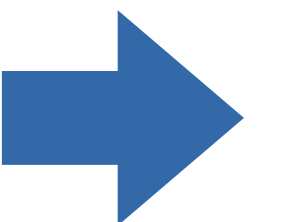
복수 지정

✓ $x = y = z = \underline{0}$

$x \longrightarrow \cdot$ 0

$y \longrightarrow \cdot$ 0

$z \longrightarrow \cdot$ 0



예약어

Keywords

Reserved Words

False	await	else	import	pass	None	break
except	in	raise	True	class	finally	is
return	and	continue	for	lambda	try	as
def	from	nonlocal	while	assert	del	global
not	with	async	elif	if	or	yield

주석

Comments

code : 2-10.py

```
1 # Calculate the area of circle
2 # in: radius from standard input
3 # out: area of circle to standard output
4 radius = float(input("Enter the radius: "))
5 from math import pi
6 area = pi * radius ** 2 # calculate the area of circle
7 print("The area of a circle with radius", radius, "is", area)
```



실습 2.3 온도 변환 서비스

온도를 표시하는 방식은 섭씨Celsius와 화씨Fahrenheit가 있다. 우리는 섭씨를 쓰지만, 미국과 같이 화씨를 쓰는 나라도 있다. 화씨온도는 우리에게 익숙하지 않으므로 섭씨로 몇도인지 환산해주면 좋겠다. 다음은 화씨온도를 섭씨로 바꾸는 공식이다.

$$C = (F - 32) \times \frac{5}{9}$$

화씨온도를 정수로 키보드 입력받아서, 섭씨로 변환하여 프린트하는 프로그램을 작성하자. 출력 형식은 아래와 동일한 형식이어야 한다.

```
Fahrenheit to Celsius conversion  
Degrees in Fahrenheit? 67  
19.4 degrees in Celsius
```

여기서 둘째 줄의 끝에 있는 67은 키보드 입력이고, 나머지는 프로그램이 프린트한 것이다. 특히 셋째 줄의 섭씨온도는 소수점 둘째 자리에서 반올림하여 첫째 자리 수까지만 표시해야 한다.

프로그래밍의 정석
파이썬

2

변수와 함수

2.1 변수 · 2.2 함수

CHAPTER 2

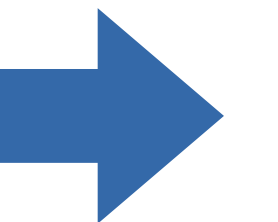
변수와 함수

2.1 변수

✓ 2.2 함수

함수

Function



내장 함수

Built-in Function

`print()`, `input()`, `int()`, `float()`, `str()`, `round()`

abs()	delattr()	hash()	memoryview()	set()
all()	dict()	help()	min()	setattr()
any()	dir()	hex()	next()	slice()
ascii()	divmod()	id()	object()	sorted()
bin()	enumerate()	input()	oct()	staticmethod()
bool()	eval()	int()	open()	str()
breakpoint()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	

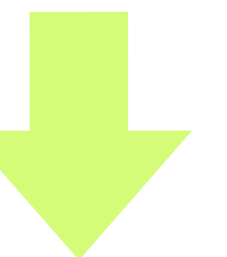
람다 요약

Lambda Abstraction

람다 식

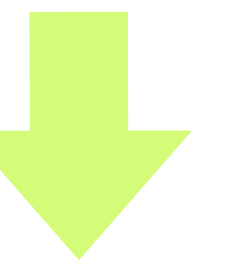
math.pi * radius ** 2

↑
원의 면적 계산 공식



람다 식

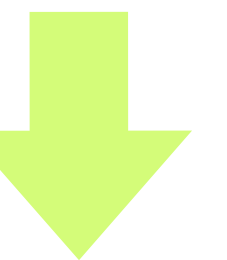
```
radius: math.pi * radius ** 2
```



람다 식

`lambda <변수> : <식>`

```
lambda radius: math.pi * radius ** 2
```

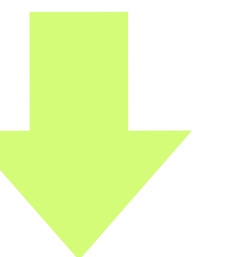


람다 식

Lambda <변수> : <식>

lambda radius: math.pi * radius ** 2

예약어
keyword

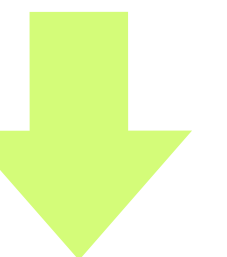


람다 식

lambda <변수> : <식>

```
lambda radius: math.pi * radius ** 2
```

↑
파라미터
parameter

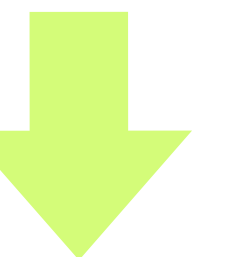


람다 식

lambda <변수> : <식>

```
lambda radius: math.pi * radius ** 2
```

↑
파라미터
parameter



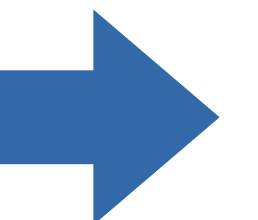
람다 식

lambda <변수> : <식>

```
lambda radius: math.pi * radius ** 2
```

↑
파라미터
parameter

↑
몸체
body



람다 식에 대입

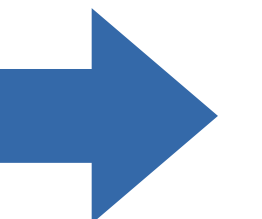
Application

(⟨람다식⟩)(⟨식⟩)

(`lambda radius: math.pi * radius ** 2`)(3)

인수

argument



함수 정의

Function definition

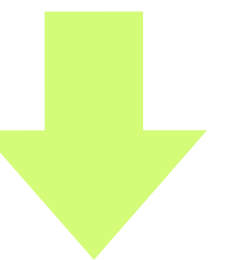
함수 호출

Function call

함수 정의

Function definition

```
def <함수이름>( <변수>, <변수>, ..., <변수> ):  
    <몸체>
```

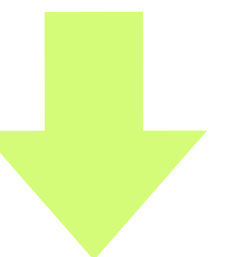


함수 정의

Function definition

```
def <함수이름>(<변수>, <변수>, ..., <변수>):  
    <몸체>
```

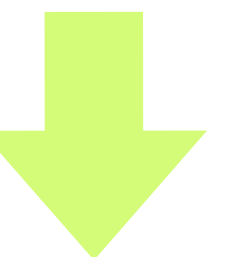
예약어
keyword



함수 정의

Function definition

```
def 함수이름(〈변수〉, 〈변수〉, ..., 〈변수〉):  
    〈몸체〉
```



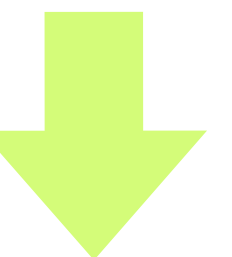
함수 정의

Function definition

```
def <함수이름> (<변수>, <변수>, ..., <변수>):  
    <몸체>
```

파라미터 parameter

형식 파라미터 formal parameter



함수 정의

Function definition

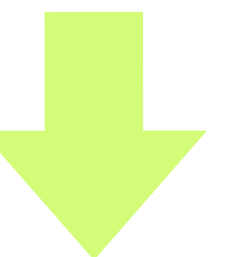
```
def <함수이름> (<변수>, <변수>, ..., <변수>):  
    <몸체>
```

코드 블록
block

함수 호출

Function call

〈함수이름〉(〈식〉, 〈식〉, ..., 〈식〉)



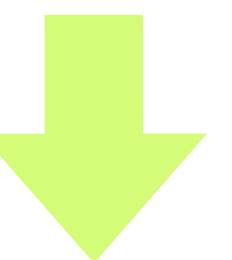
함수 호출

Function call

〈함수이름〉 (〈식〉, 〈식〉, ..., 〈식〉)

인수 argument

실제 파라미터 actual parameter



함수 호출

Function call

〈함수이름〉(〈식〉, 〈식〉, ..., 〈식〉)

지정

지정

지정

def 〈함수이름〉(〈변수〉, 〈변수〉, ..., 〈변수〉):
 〈몸체〉

함수 정의

Function definition

```
def <함수이름>( <변수>, <변수>, ..., <변수> ):  
    <몸체>
```

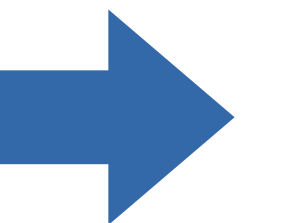
```
    return <식>
```

프로시저

Procedure

return 문이 없는 함수

함수 만들기 실전





실습 2.4 동전 합산 함수

〈실습 2.2〉에서 작성한 코드를 함수를 만들어 개선해 보자. 키보드 입력 대신 각 동전의 개수를 인수로 받아서 동전의 총액을 계산하여 리턴해 주는 함수 `coin_in_total` 을 다음 틀에 맞추어 작성하자.

code : 2-17.py

```
1 def coin_in_total(c500, c100, c50, c10):
2     return None # Write your expression here.
3
4 # Test code
5 print(coin_in_total(4,2,3,4)) # 2390
```

이 함수를 완성하여 잘 작동하는지 확인한 다음, 〈실습 2.2〉에서 작성한 코드를 이 함수를 호출하는 방식으로 재작성하자.



실습 2.5 온도 변환 함수

〈실습 2.3〉에서 작성한 코드를 함수를 만들어 개선해 보자. 화씨온도를 인수로 받아서 섭씨온도로 변환해주는 함수 `fahrenheit2celsius`를 다음 틀에 맞추어 작성하자.

code : 2-18.py

```
1 def fahrenheit2celsius(f):
2     return None # Write your expression here.
3
4 # Test code
5 print(fahrenheit2celsius(67))           # 19.4444444444444443
6 print(round(fahrenheit2celsius(67),1))  # 19.4
```

화씨온도를 섭씨온도로 변환하는 공식은 다음과 같다.

$$C = (F - 32) \times \frac{5}{9}$$



실습 2.6 9의 보수 계산 함수

9의 보수`complement`는 다음 표에서 관찰할 수 있듯이 수의 각 자리 수에서 9와의 차이로 얻는 수이다.

수	9의 보수
0	9
9	0
4	5
18	81
40	59
307	692
9142	857
9965	34
9999	0

자연수 n 의 자리 수를 k 라 하면, n 에 대한 9의 보수는 다음 공식으로 구할 수 있다.

$$10^k - 1 - n$$

자연수를 인수로 받아서 그 수의 9의 보수를 계산하여 리턴해 주는 함수 `complement_nine`을 다음 코드 틀에 맞추어 작성하자. 인수는 항상 자연수로 제공된다고 가정한다.

```
1 def complement_nine(n):
2     return None # Write your expression here.
3
4 # Test code
5 print(complement_nine(0)) # 9
6 print(complement_nine(9)) # 0
7 print(complement_nine(4)) # 5
8 print(complement_nine(18)) # 81
9 print(complement_nine(40)) # 59
10 print(complement_nine(307)) # 692
11 print(complement_nine(9142)) # 857
12 print(complement_nine(9965)) # 34
13 print(complement_nine(9999)) # 0
```

코딩 가이드

자연수 인수 n 의 자리 수 k 를 알아야 한다. 그런데 문자열의 길이는 내장 함수 `len`으로 다음과 같이 구할 수 있다.

```
>>> len("365")
3
```

`len`을 활용하여 n 의 자리 수를 어떻게 구할지 고민해 보자.

