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

프로그래밍의 정석

파이썬

도경구 지음

생능출판

CHAPTER 2

변수와 함수

CHAPTER 2

변수와 함수

✔ 2.1 변수

2.2 함수

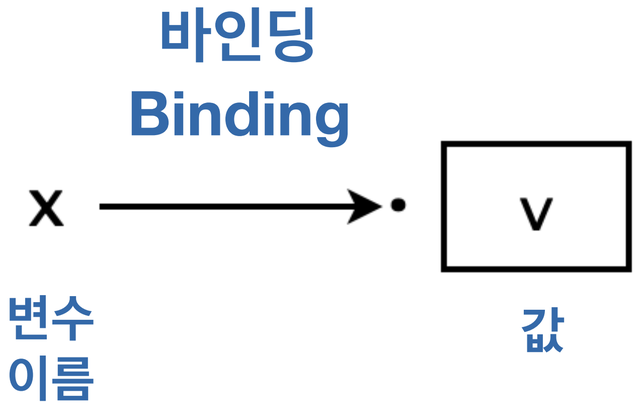
변수

Variable

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

네임스페이스

Namespace



지정문

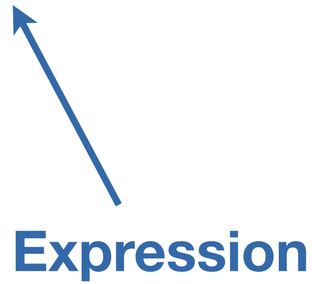
Assignment

<변수> = <식>

지정문

Assignment

<변수> = <식>



Expression

지정문

Assignment

<변수> = <식>

지정문

Assignment Statement

✓ `x = 3 + 4`

네임스페이스

지정문

Assignment Statement

✔ `x = 3 + 4`

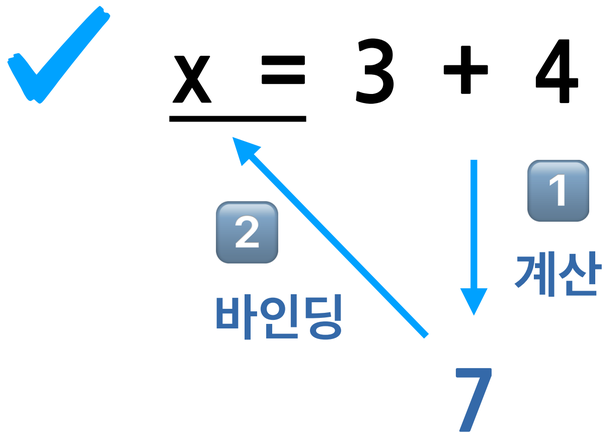
1️⃣ 계산

7

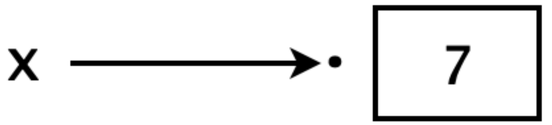
네임스페이스

지정문

Assignment Statement



네임스페이스



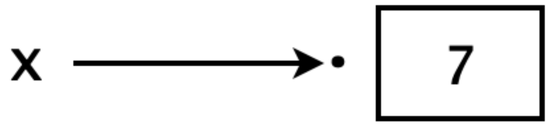
지정문

Assignment

x = 3 + 4

✓ x = x + 2

네임스페이스



지정문

Assignment

```
x = 3 + 4
```

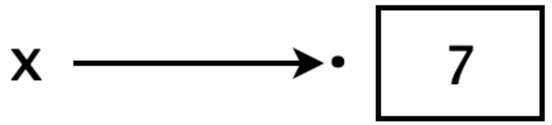
✔

```
x = x + 2
```

1️⃣ 계산

9

네임스페이스



지정문

Assignment

x = 3 + 4

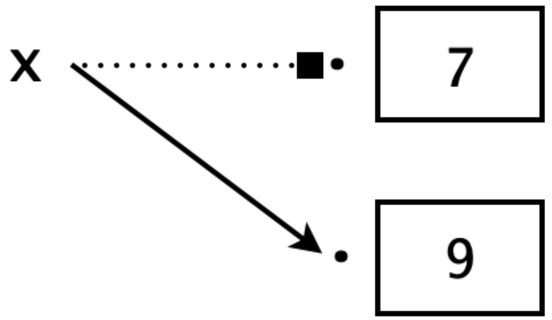
✔ x = x + 2

1 계산

2 바인딩

9

네임스페이스



지정문

Assignment

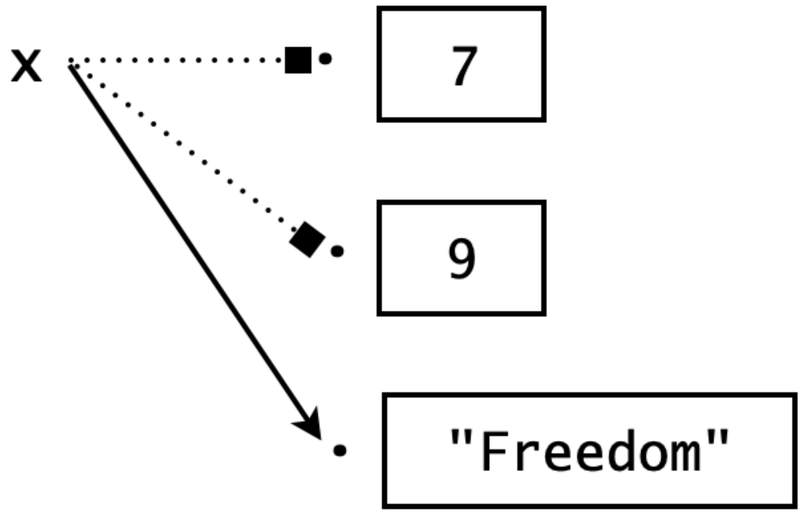
```
x = 3 + 4
```

```
x = x + 2
```

✓

```
x = "Freedom"
```

네임스페이스



Python	**Dynamic Binding** **동적 바인딩**
C **Java**	**Static Binding** **정적 바인딩**

Static Binding
정적 바인딩

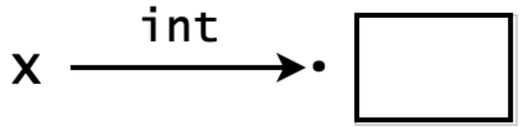
✓ `int x`

네임스페이스

Static Binding
정적 바인딩

✓ `int x`

네임스페이스



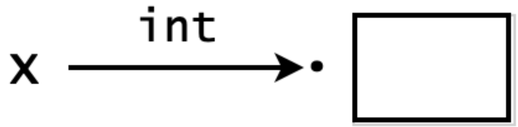
Static Binding
정적 바인딩

```
int x
```

✓

```
x = 3 + 4
```

네임스페이스



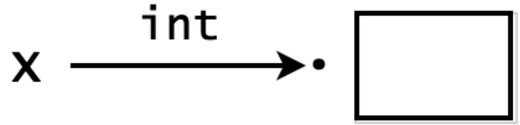
Static Binding
정적 바인딩

```
int x
x = 3 + 4
```

1️⃣ 계산

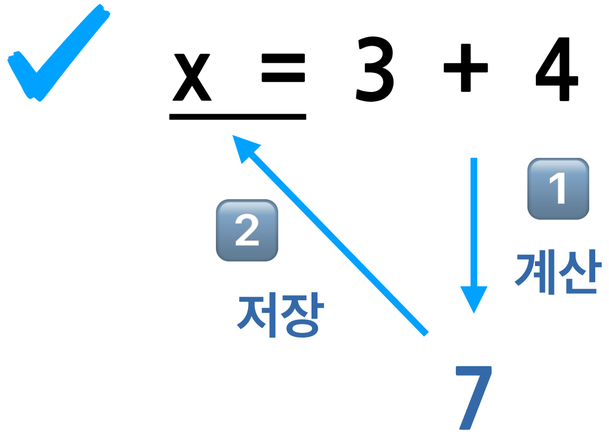
7

네임스페이스

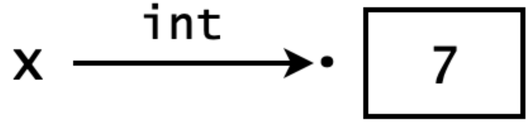


Static Binding
정적 바인딩

```
int x

x = 3 + 4
```



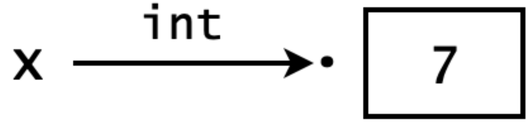
네임스페이스



Static Binding
정적 바인딩

```
int x

x = 3 + 4

x = x + 2
```

네임스페이스



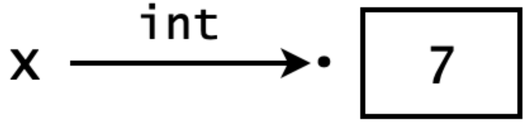
Static Binding
정적 바인딩

```
int x

x = 3 + 4

x = x + 2
```

1️⃣ 계산

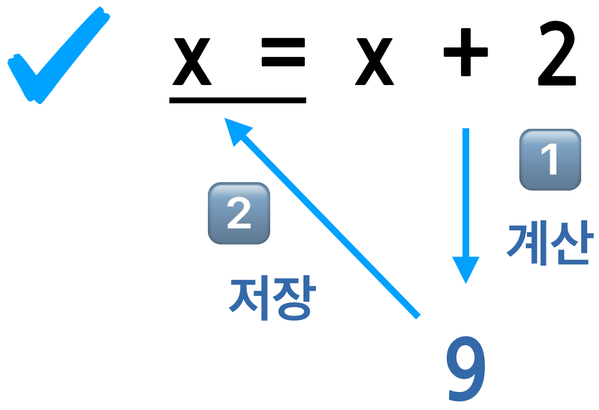
9

네임스페이스

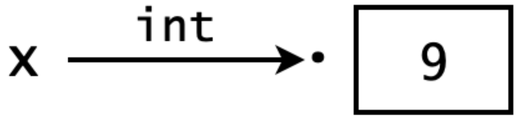


Static Binding
정적 바인딩

```
int x

x = 3 + 4

x = x + 2
```



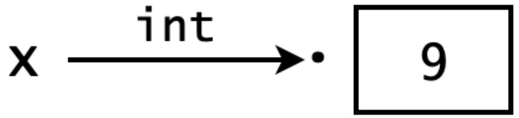
네임스페이스



Static Binding
정적 바인딩

```
int x

x = 3 + 4

x = x + 2

x = "Freedom"
```

네임스페이스

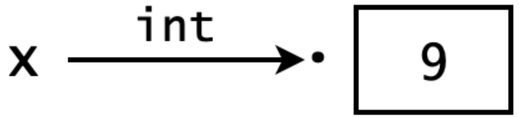


Static Binding
정적 바인딩

```
int x

x = 3 + 4

x = x + 2

x = "Freedom"
```

✓ ✗

네임스페이스

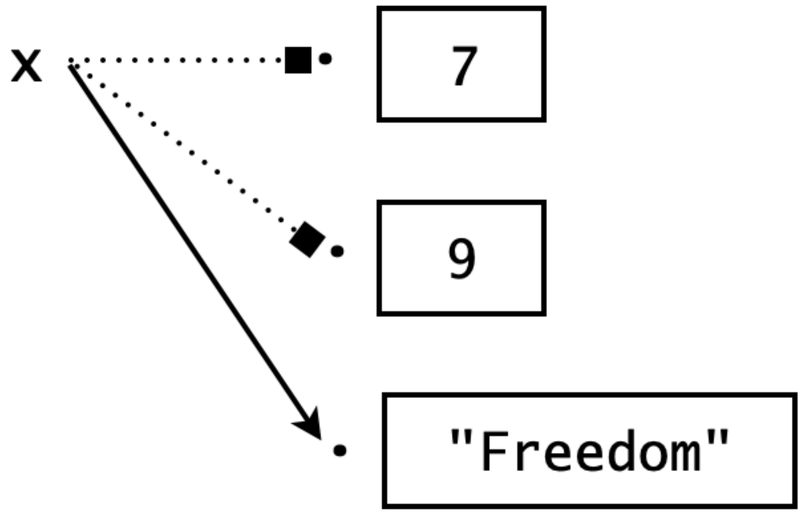


지정문

Assignment

```
x = 3 + 4
x = x + 2
x = "Freedom"
print(pooh)
```

네임스페이스



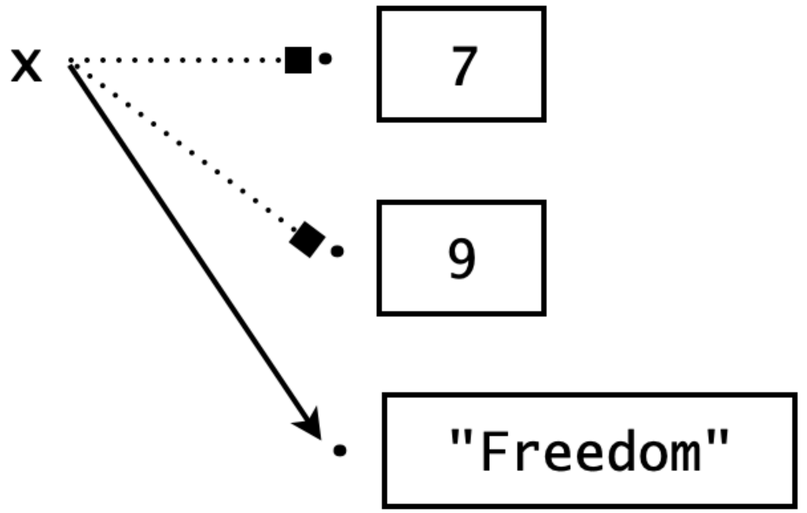
지정문

Assignment

```
x = 3 + 4
x = x + 2
x = "Freedom"
```

✓ `print(pooh)` ✗

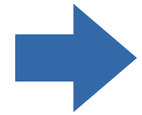
네임스페이스



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

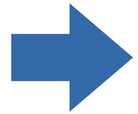
Python coding convention

원의 면적 구하기

$$\pi \times r^2$$

표준 라이브러리

math



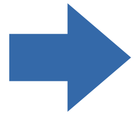
통합개발환경

IDE

Integrated Development Environment

IDLE의 편집기 활용

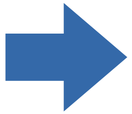
프로그램을 파일에 저장하여 한꺼번에 실행



표준 입력

Standard Input

```
x = input()
```



반올림

```
round()
```

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

프로그래밍의 정석

파이썬

도경구 지음

서니웨이 북스

pp.66~67

실습 2.1 키보드 입력과 반올림

실습 2.2 동전 합산 서비스

지정문의 실행 순서

```
x = 3

y = 7
```

지정문의 실행 순서

- ✓ x = 3
- ✓ y = 7

지정문의 실행 순서

x = 3

y = 7

✓ x = y

y = x

지정문의 실행 순서

x = 3

y = 7

✓ x = y

y = x

지정문의 실행 순서

x = 3

y = 7

~~x = y~~

✓ y = x

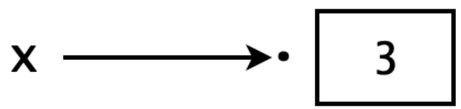
지정문의 실행 순서

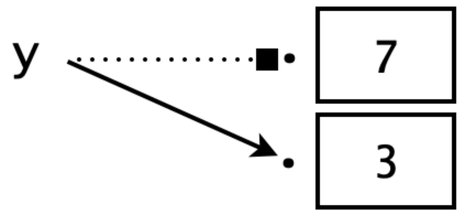
x = 3

y = 7

~~x = y~~

y = x





지정문의 실행 순서

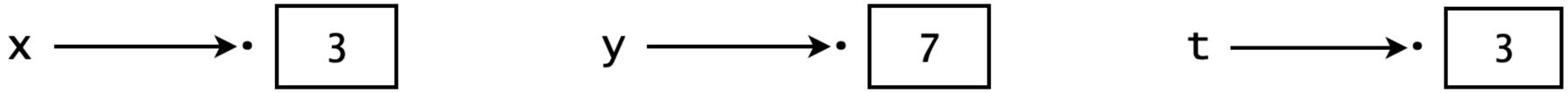
- [x] `x = 3`
- [x] `y = 7`
- [ ] `t = x`
- [ ] `x = y`
- [ ] `y = t`

지정문의 실행 순서

```
x = 3
y = 7
t = x
x = y
y = t
```

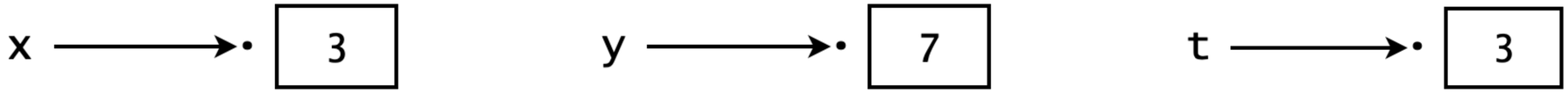
지정문의 실행 순서

```
x = 3
y = 7
t = x
x = y
y = t
```



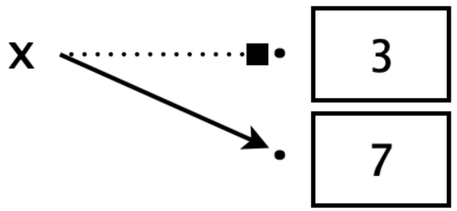
지정문의 실행 순서

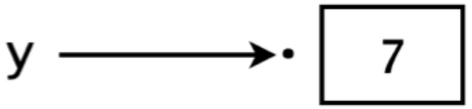
```
x = 3
y = 7
t = x
x = y
y = t
```

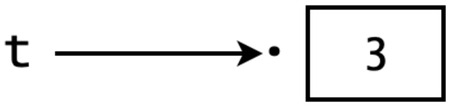


지정문의 실행 순서

```
x = 3
y = 7
t = x
x = y
y = t
```

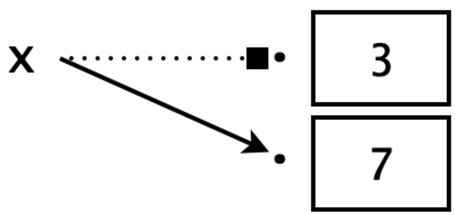


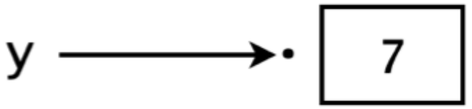


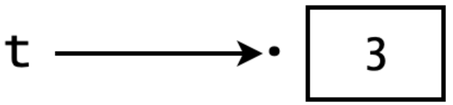


지정문의 실행 순서

```
x = 3
y = 7
t = x
x = y
y = t
```

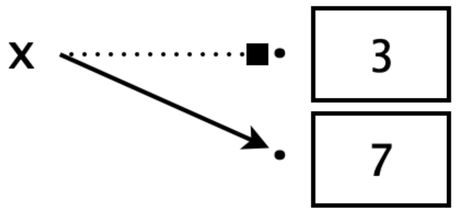


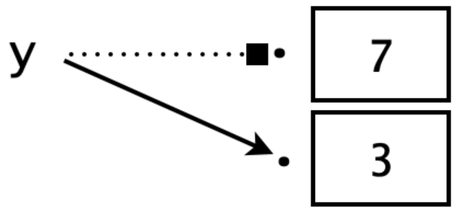


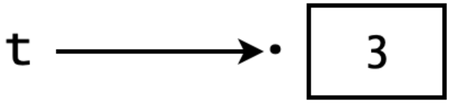


지정문의 실행 순서

```
x = 3
y = 7
t = x
x = y
y = t
```

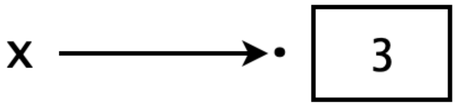


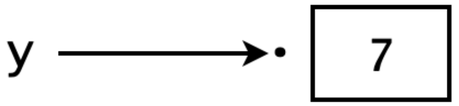




동시 지정

✔ `x, y = 3, 7`

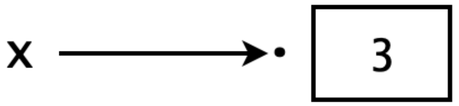


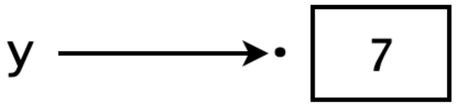


`x, y = y, x`

동시 지정

```
x, y = 3, 7
```





✓

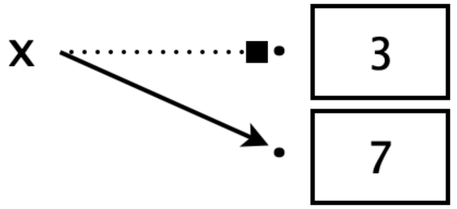
```
x, y = y, x
```

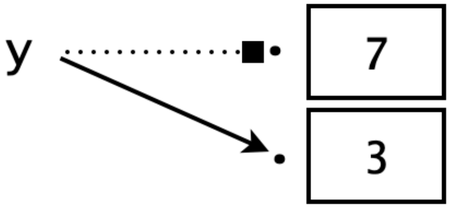
동시 지정

```
x, y = 3, 7
```

✓

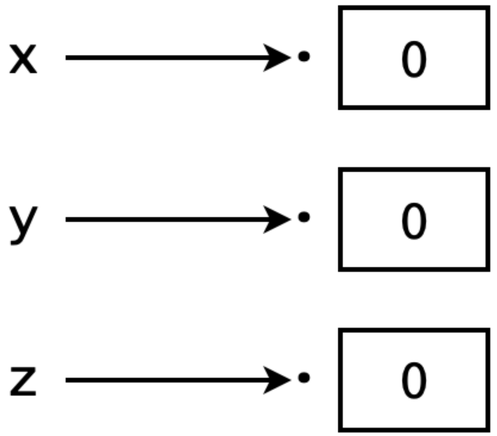
```
x, y = y, x
```





복수 지정

✔ `x = y = z = 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)
```

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

프로그래밍의 정석

파이썬

도경구 지음

생능출판

pp.72

실습 2.3 온도 변환 서비스

CHAPTER 2

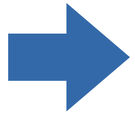
변수와 함수

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
```

람다 식

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

람다 식

lambda 〈변수〉 : 〈식〉

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

람다 식

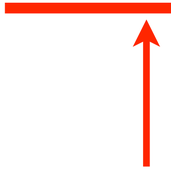
lambda 〈변수〉 : 〈식〉

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

람다 식

lambda 〈변수〉 : 〈식〉

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

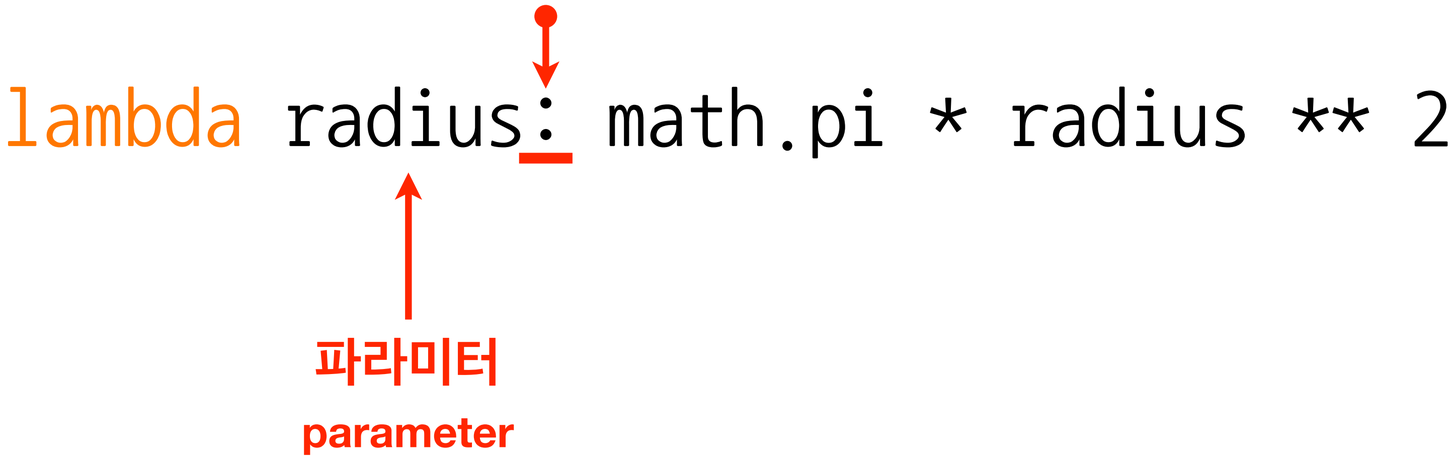


파라미터

parameter

람다 식

lambda 〈변수〉 : 〈식〉



람다 식

lambda 〈변수〉 : 〈식〉

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

람다 식에 대입

Application

(〈람다식〉)(〈식〉)

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

함수 정의 Function definition

함수 호출 Function call

함수 정의 Function definition

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

함수 정의 Function definition

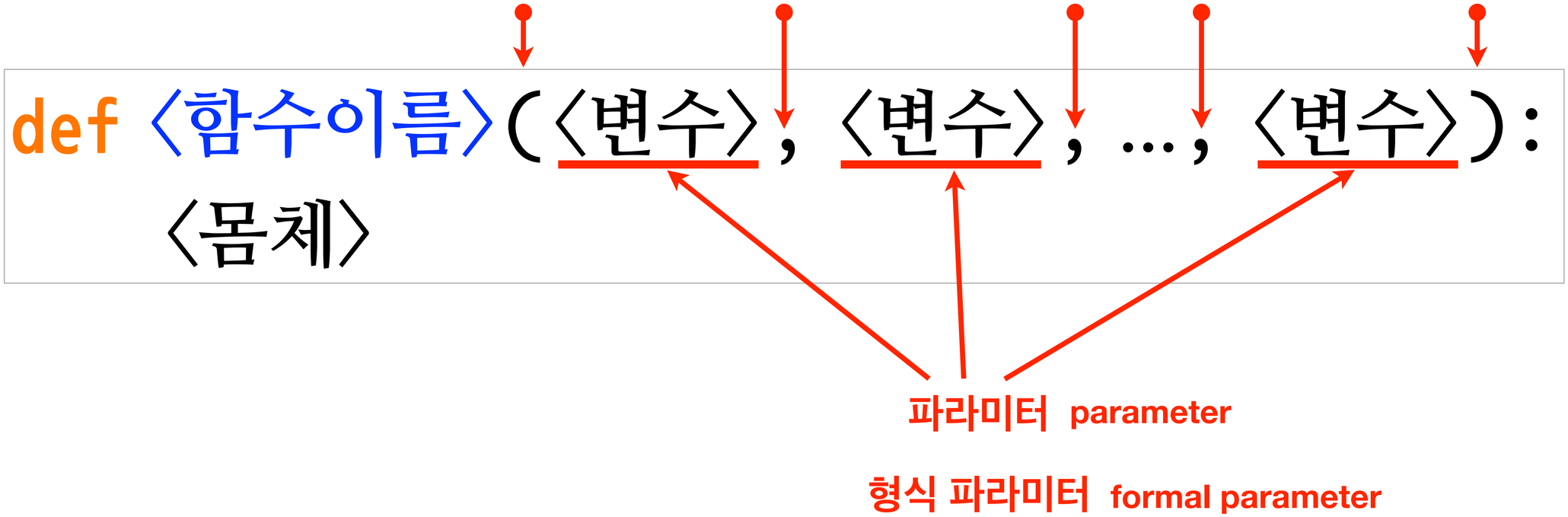
```
def 〈함수이름〉(〈변수〉, 〈변수〉, ..., 〈변수〉):
    〈몸체〉
```

예약어
keyword

함수 정의 Function definition

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

함수 정의 Function definition



함수 정의 Function definition

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

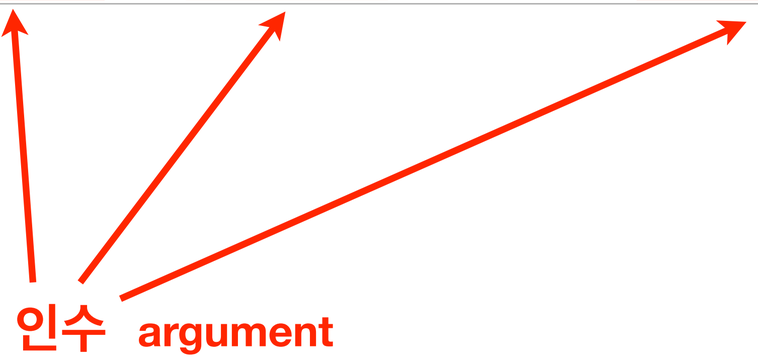
코드 블록
block

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〉)


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

함수 정의 Function definition

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

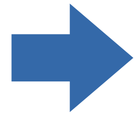
```
    return 〈식〉
```

프로시저

Procedure

`return` 문이 없는 함수

함수 만들기 실전



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

프로그래밍의 정석

파이썬

도경구 지음

생능출판

pp.83~86

실습 2.4 동전 합산 함수

실습 2.5 온도 변환 함수

실습 2.6 9의 보수 계산 함수

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

프로그래밍의 정석

파이썬

도경구 지음

생능출판

CHAPTER 2

변수와 함수