

스크립트 프로그래밍

02 프로그래밍 기초

2016 2학기 (02분반)

강승우

학습 목표

- 간단한 계산을 수행하는 프로그램을 작성할 수 있다. (2.2절)
- input 함수를 사용하여 프로그램 사용자의 입력을 얻을 수 있다. (2.3절)
- 변수와 함수와 같은 항목에 이름을 붙인 식별자를 사용할 수 있다. (2.4절)
- 데이터를 변수에 할당할 수 있다. (2.5절)
- 동시 할당을 수행할 수 있다. (2.6절)
- 이름 상수를 정의할 수 있다. (2.7절)
- +, -, *, /, //, %, ** 연산자를 사용할 수 있다. (2.8절)
- 수치 표현식을 작성하고 평가할 수 있다. (2.9절)
- 코딩을 단순화하기 위해 확장 할당 연산자를 사용할 수 있다. (2.10절)
- int와 round 함수를 이용하여 수치 타입 변환과 반올림을 수행할 수 있다. (2.11절)
- time.time()을 사용하여 현재 시스템 시간을 구할 수 있다. (2.12절)
- 소프트웨어 개발 절차를 이해하고 그 절차를 상환액 프로그램에 적용할 수 있다. (2.13절)

프로그램 작성하기

- 프로그램 작성

- 문제 해결을 위한 전략을 설계한 후에
그 전략을 구현하기 위해 프로그래밍 언어를 사용하는 것

- 알고리즘을 설계한 후에 그 알고리즘을 프로그래밍 명령문 혹은 코드로 바꾸는 것

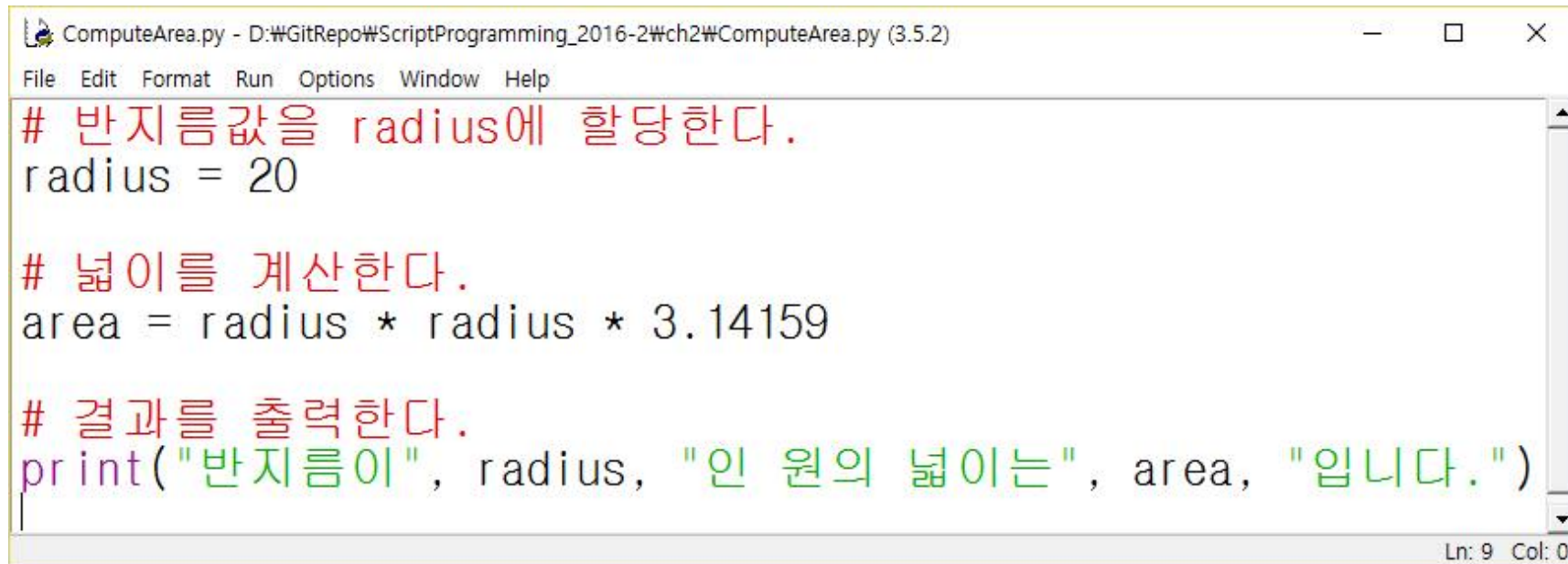
- 알고리즘 (자연어 혹은 의사코드로 표현)

- 실행할 행동들과 그 행동들의 수행 순서를 나열함으로써 문제를 어떻게 해결할지를 서술하는 것
 - 프로그래밍 언어로 프로그램을 작성하기 전에 프로그램을 설계하는데 도움을 줌

예: 원의 넓이를 계산하는 알고리즘

- 사용자로부터 원의 반지름을 얻는다
- 수식을 이용하여 원의 넓이를 계산한다
 - $\text{넓이} = \text{반지름} * \text{반지름} * 3.14$
- 결과를 화면에 출력한다

원의 면적을 계산하는 간단한 프로그램



```
ComputeArea.py - D:\GitRepo\ScriptProgramming_2016-2\ch2\ComputeArea.py (3.5.2)
File Edit Format Run Options Window Help
# 반지름값을 radius에 할당한다.
radius = 20

# 넓이를 계산한다.
area = radius * radius * 3.14159

# 결과를 출력한다.
print("반지름이", radius, "인 원의 넓이는", area, "입니다.")
Ln: 9 Col: 0
```

프로그램 실행 추적

변수 radius에 20을 할당 →
메모리 공간에 20을 저장하고 이를
radius라는 변수 이름으로 참조

radius → 20

변수 area에 계산 결과값을 할당

area → 1256.636

콘솔에 메시지를 출력

```
ComputeArea.py - D:\GitRepo\ScriptProgramming_2016-2\ch2\ComputeArea.py (3.5.2)
File Edit Format Run Options Window Help
# 반지름값을 radius에 할당한다.
radius = 20
# 넓이를 계산한다.
area = radius * radius * 3.14159
# 결과를 출력한다.
print("반지름이", radius, "인 원의 넓이는", area, "입니다.")
Ln: 9 Col: 0
```

```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [M
SC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more informat
ion.
>>>
RESTART: D:\GitRepo\ScriptProgramming_2016-2\ch2\ComputeAre
a.py
반지름이 20 인 원의 넓이는 1256.636 입니다.
>>> |
Ln: 6 Col: 4
```

콘솔에서 입력 받기

- input 함수 사용
 - variable = input("값을 입력하세요: ")
 - 문자열 값으로 입력된다
 - eval 함수 사용
 - 문자열 형태의 값을 숫자 값으로 변경
 - var = eval(stringVariable)
- eval("51 + (54 * (3 + 2))")의 결과는?

콘솔 입력 이용하는 프로그램

```
ComputeAreaWithConsoleInput.py - D:\GitRepo\ScriptProgramming_2016-2\ch2\ComputeAreaWithConsoleInput.py (3.5.2)
File Edit Format Run Options Window Help
# 사용자로부터 반지름 값을 입력받는다.
radius = eval(input("반지름을 입력하세요: "))

# 넓이를 계산한다.
area = radius * radius * 3.14159

# 결과를 출력한다.
print("반지름이", radius, "인 원의 넓이는", area, "입니다.")
```

```
ComputeAverage.py - D:\GitRepo\ScriptProgramming_2016-2\ch2\ComputeAverage.py (3.5.2)
File Edit Format Run Options Window Help
# 사용자로부터 세 개의 숫자를 입력받는다.
number1 = eval(input("첫 번째 숫자를 입력하세요: "))
number2 = eval(input("두 번째 숫자를 입력하세요: "))
number3 = eval(input("세 번째 숫자를 입력하세요: "))

# 평균을 계산한다.
average = (number1 + number2 + number3) / 3

# 결과를 출력한다.
print(number1, number2, number3,
      "의 평균은", average, "입니다.")
```

Ln: 9 Col: 10

식별자 (identifier)

- 변수와 함수 등의 항목들을 프로그램 내에서 서로 구별할 수 있도록 해주는 이름
 - 문자, 숫자, 밑줄(underscore)로 구성된 일련의 문자
- 식별자 작성 규칙
 - 문자 혹은 밑줄로 시작해야 하고 숫자로 시작해서는 안 된다
 - 식별자는 키워드가 될 수 없다
 - 예약어(reserved word)로 불리는 키워드(keyword)는 파이썬에서 특별한 의미를 가진다
 - 예를 들어, import는 키워드인데 임의의 모듈을 프로그램에 импорт 할지를 파이썬 해석기에 알려준다
 - 식별자에는 길이 제한이 없다
 - 대문자와 소문자를 구분한다
 - Area, area, AREA는 서로 다른 식별자이다

식별자

- 서술 식별자
 - 식별자로 축약어를 가급적 사용하지 않고 완전한 단어를 사용하는 것이 좋다
 - 프로그램을 읽고 이해하기 쉽게 해준다
 - 예: numberOfStudent vs. numStuds, numOfSt, numOfStudent
 - 카멜 표기법 (camelCase)

변수, 할당문

- 변수와 할당문

- 변수 (variable): 메모리에 저장된 값을 참조하는 이름

radius = 1.0

area = radius * radius * 3.14159

radius = 3.0

area = radius * radius * 3.14159

- 할당문 (assignment statement): 어떤 값을 1개의 변수에 할당하는 명령문

표현식

- 표현식 (expression)
 - 값, 변수, 연산자를 포함한 계산

$$x = 1$$

$$y = x + 5$$

$$x = 5 * (3 / 2) + 10$$

$$x = x + 1$$

$i = j = k = 1$ (하나의 값을 여러 변수에 할당)

$1 = x$ (잘못됨, 변수가 왼쪽에, 값이 오른쪽에)

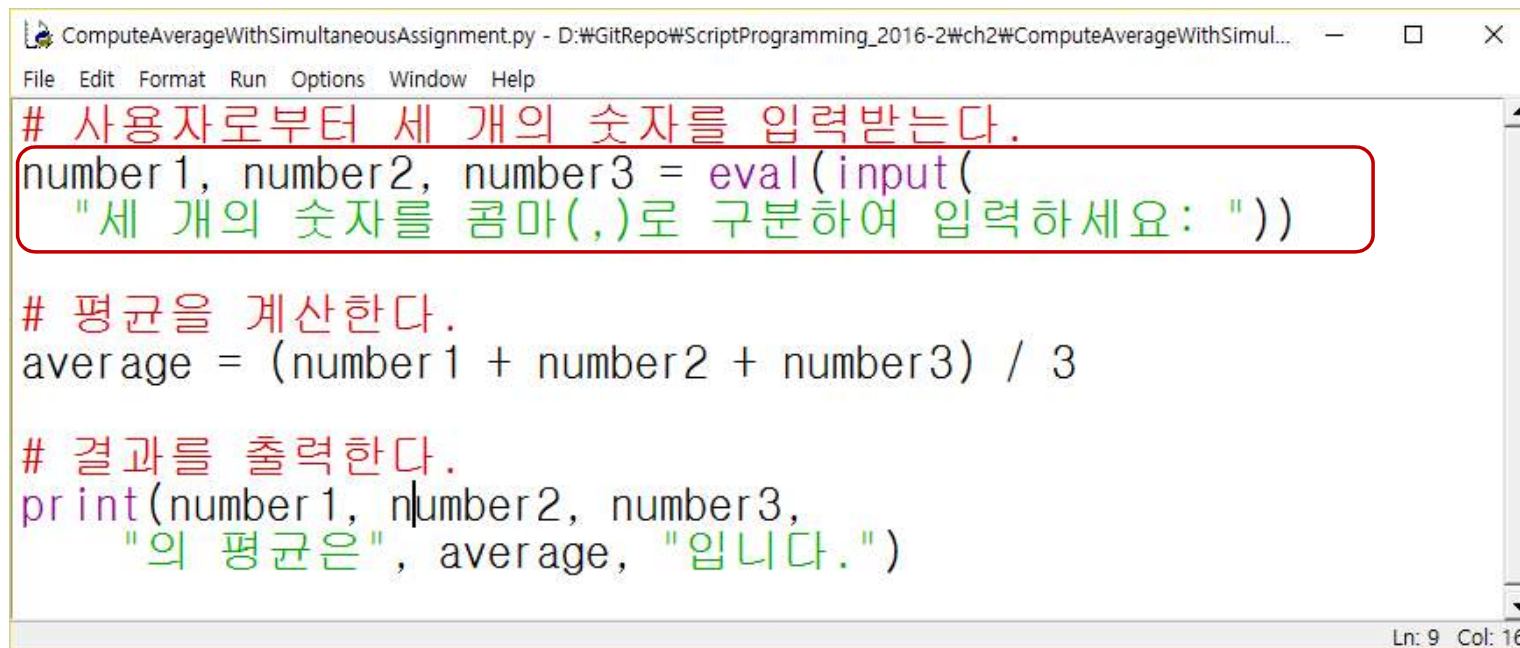
동시 할당문

- `var1, var2, var3 = exp1, exp2, exp3`
 - 오른쪽 부분에 있는 모든 표현식들을 계산하고, 계산된 값을 왼쪽 부분에 각각 대응되는 변수로 동시에 할당
- 변수 값 교환하기

```
x = 1
y = 2
temp = x
x = y
y = temp
-----
x, y = y, x
```

동시 할당문

- 하나의 명령문으로 여러 개의 입력을 얻기 위해서도 사용 가능



```
ComputeAverageWithSimultaneousAssignment.py - D:\GitRepo\ScriptProgramming_2016-2\ch2\ComputeAverageWithSimul...
File Edit Format Run Options Window Help
# 사용자로부터 세 개의 숫자를 입력받는다.
number1, number2, number3 = eval(input(
    "세 개의 숫자를 콤마(,)로 구분하여 입력하세요: "))

# 평균을 계산한다.
average = (number1 + number2 + number3) / 3

# 결과를 출력한다.
print(number1, number2, number3,
      "의 평균은", average, "입니다.")
```

Ln: 9 Col: 16

이름 상수 (named constant)

- 프로그램이 실행되는 동안 변수의 값은 바뀔 수 있지만, 이름 상수(named constant) 혹은 상수(constant)는 절대로 바뀌지 않는 영구적인 데이터를 가지고 있다
- 상수의 이름
 - 파이썬은 상수에 이름을 붙이는 것에 대해 특별한 문법을 가지고 있지는 않다
 - 상수를 나타내기 위해 단순히 변수를 생성하나, 상수와 변수를 서로 구별하기 위해 상수에는 **대문자를 사용**하여 이름을 붙인다
 - 예: $\text{PI} = 3.14159$
- 상수 사용의 이점
 - 동일한 값이 여러 번 사용될 때, 반복을 피할 수 있음
 - 상수 값을 변경해야 하는 경우, 코드 상에서 한번만 변경하면 됨
 - 서술 이름을 사용하여 프로그램을 읽기 쉬움

수치 데이터 타입과 연산자

- 수치 데이터 타입

- 정수 (int): 3, 4, 5
- 실수 (float): 3.0, 3.14159, 10.123

- 연산자

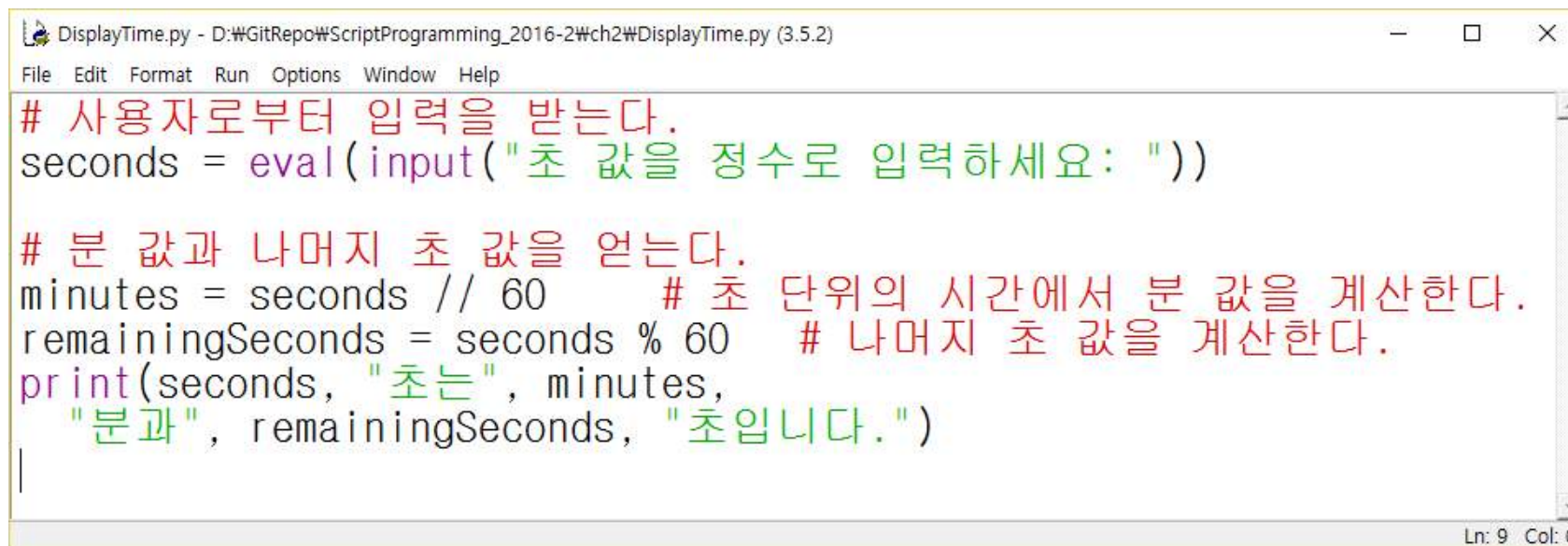
연산자	의미	예	결과
+	덧셈	$34 + 1$	35
-	뺄셈	$34.0 - 0.1$	33.9
*	곱셈	$300 * 30$	9000
/	실수 나눗셈	$1 / 2$	0.5
//	정수 나눗셈	$1 // 2$	0
**	누승	$4 ** 0.5$	2.0
%	나머지	$20 \% 3$	2

//, **, % 연산자

- //
- 정수 나눗셈: 결과로 정수 반환, 소수점 이하 버림
- **
 - $10 * 10 * 10 \rightarrow 10 ** 3$
 - 거듭제곱 표현
- %
 - 나머지 값을 결과로 산출
 - 사용 예
 - 짝수, 홀수 구분
 - 요일 구분: 오늘은 목요일, 10일 후는 무슨 요일일까?
 $(4 + 10) \% 7 = 0 \rightarrow$ 일요일

시간 출력하기

- 초 단위의 시간에서 분 값과 나머지 초 값을 얻어내는 프로그램



```
DisplayTime.py - D:\GitRepo\ScriptProgramming_2016-2\ch2\DisplayTime.py (3.5.2)
File Edit Format Run Options Window Help
# 사용자로부터 입력을 받는다.
seconds = eval(input("초 값을 정수로 입력하세요: "))

# 분 값과 나머지 초 값을 얻는다.
minutes = seconds // 60      # 초 단위의 시간에서 분 값을 계산한다.
remainingSeconds = seconds % 60  # 나머지 초 값을 계산한다.
print(seconds, "초는", minutes,
      "분과", remainingSeconds, "초입니다.")
|
Ln: 9 Col: 0
```

- 시, 분, 초 값을 얻어내는 것으로 변경하려면 어떻게 하면 될까?

과학적 표기법

- 부동소수점 표현은 과학적 표기법으로 표현될 수도 있다.
- 예
 - $1.23456e+2$ (또는 $1.23456e2$) = 123.456
 - $1.23456e-2$ = 0.0123456
- 지수를 나타내는 E(또는 e)는 소문자, 대문자 의미가 동일하다.

산술 연산식

$$\frac{3 + 4x}{5} - \frac{10(y - 5)(a + b + c)}{x} + 9\left(\frac{4}{x} + \frac{9 + x}{y}\right)$$

- 위 산술식은 아래와 같은 파이썬 표현식으로 변환할 수 있다

파이썬 표현식

```
(3 + 4 * x) / 5 - 10 * (y - 5) * (a + b + c) / x + 9 * (4 / x + (9 + x) / y)
```

연산자 우선순위와 표현식 평가



- 지수(**)
- 곱셈, 실수 나눗셈, 정수 나눗셈, 나머지 (왼쪽에서 오른쪽)
- 덧셈, 뺄셈 (왼쪽에서 오른쪽)

확장 할당 연산자

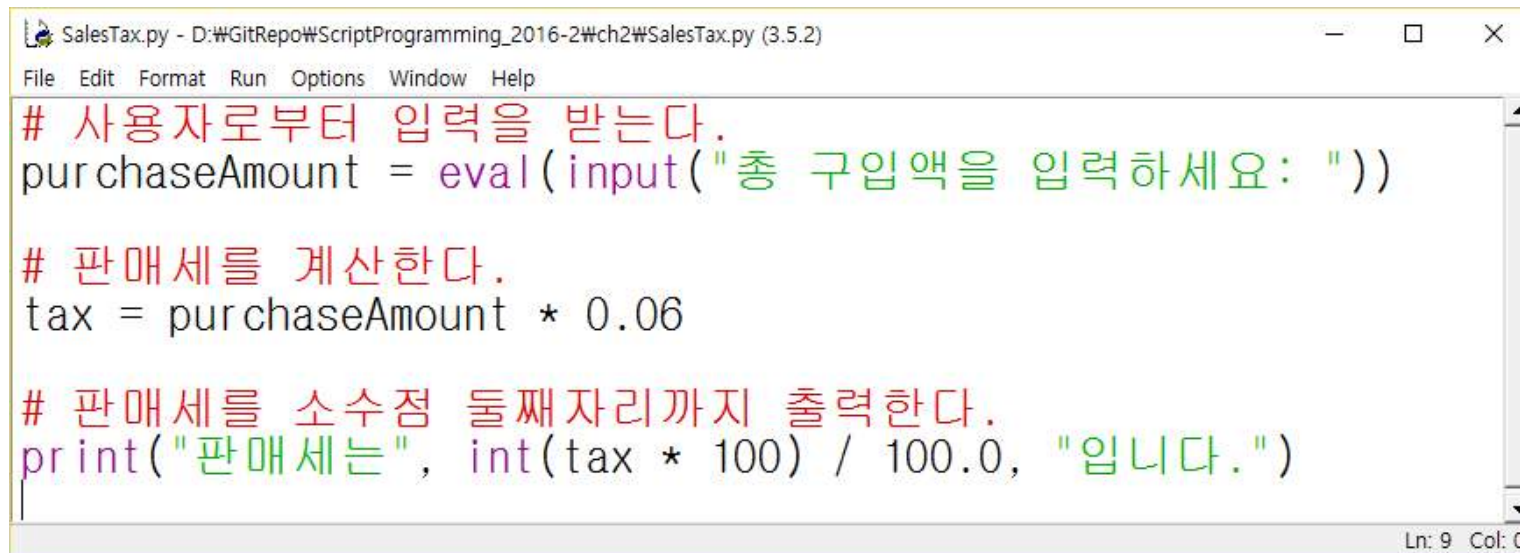
연산자	예	의미
<code>+=</code>	<code>i += 8</code>	<code>i = i + 8</code>
<code>-=</code>	<code>i -= 8</code>	<code>i = i - 8</code>
<code>*=</code>	<code>i *= 8</code>	<code>i = i * 8</code>
<code>/=</code>	<code>i /= 8</code>	<code>i = i / 8</code>
<code>/=</code>	<code>i /= 8</code>	<code>i = i / 8</code>
<code>%=</code>	<code>i %= 8</code>	<code>i = i % 8</code>
<code>**=</code>	<code>i **= 8</code>	<code>i = i ** 8</code>

타입 변환과 반올림

- 서로 다른 타입의 피연산자 2개를 가진 이항 연산을 수행하면?
 - 정수와 실수를 곱하면?
 - $3 * 4.5 \rightarrow 3.0 * 4.5$
 - 정수를 실수로 타입 변환
- `datatype(value)`
 - `int(4.5) → 4`
 - `float(4) → 4.0`
 - `str(4) → "4"`
- `round(4.6) → 5`
 - `round`: 어떤 숫자에 가장 가까운 값으로 반올림

소수점 이하 2자리 유지하기

- 소수점 이하 2자리까지 판매세를 출력하는 프로그램 작성
 - 판매세: 6%로 가정



```
SalesTax.py - D:\GitRepo\ScriptProgramming_2016-2\ch2\SalesTax.py (3.5.2)
File Edit Format Run Options Window Help
# 사용자로부터 입력을 받는다.
purchaseAmount = eval(input("총 구입액을 입력하세요: "))

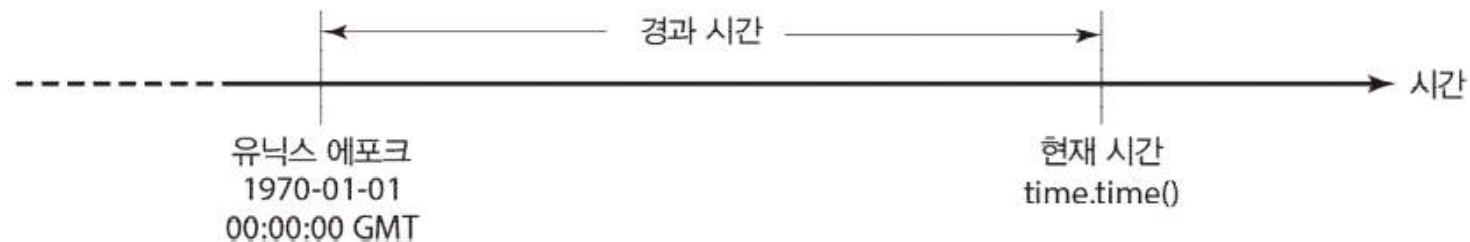
# 판매세를 계산한다.
tax = purchaseAmount * 0.06

# 판매세를 소수점 둘째자리까지 출력한다.
print("판매세는", int(tax * 100) / 100.0, "입니다.")
```

- **`int(tax * 100) / 100.0`**

사례 연구: 현재 시간 출력하기

- 그리니치 평균시(Greenwich Mean Time; GMT)인 현재 시간을 13:19:14와 같이 시:분:초 형태로 화면에 출력하는 프로그램
- 시간 관련 파이썬 모듈 이용
 - time
- time 모듈의 time() 함수
 - 1970년 1월 1일을 기준 00:00:00 시간으로 하여 현재 시간까지 경과된 시간을 밀리초 단위로 반환 (1970년: Unix 운영체제가 공식 도입된 연도)



예제 코드

```
ShowCurrentTime.py - D:\GitRepo\ScriptProgramming_2016-2\ch2\ShowCurrentTime.py (3.5.2)
File Edit Format Run Options Window Help

import time

currentTime = time.time() # 현재 시간을 얻어온다.

# 1970년 1월 1일 자정 이후로의 전체 초 값을 얻어온다.
totalSeconds = int(currentTime)

# 현재 시간의 초 값을 계산한다.
currentSecond = totalSeconds % 60

# 전체 분 값을 계산한다.
totalMinutes = totalSeconds // 60

# 현재 시간의 분 값을 계산한다.
currentMinute = totalMinutes % 60

# 전체 시 값을 계산한다.
totalHours = totalMinutes // 60

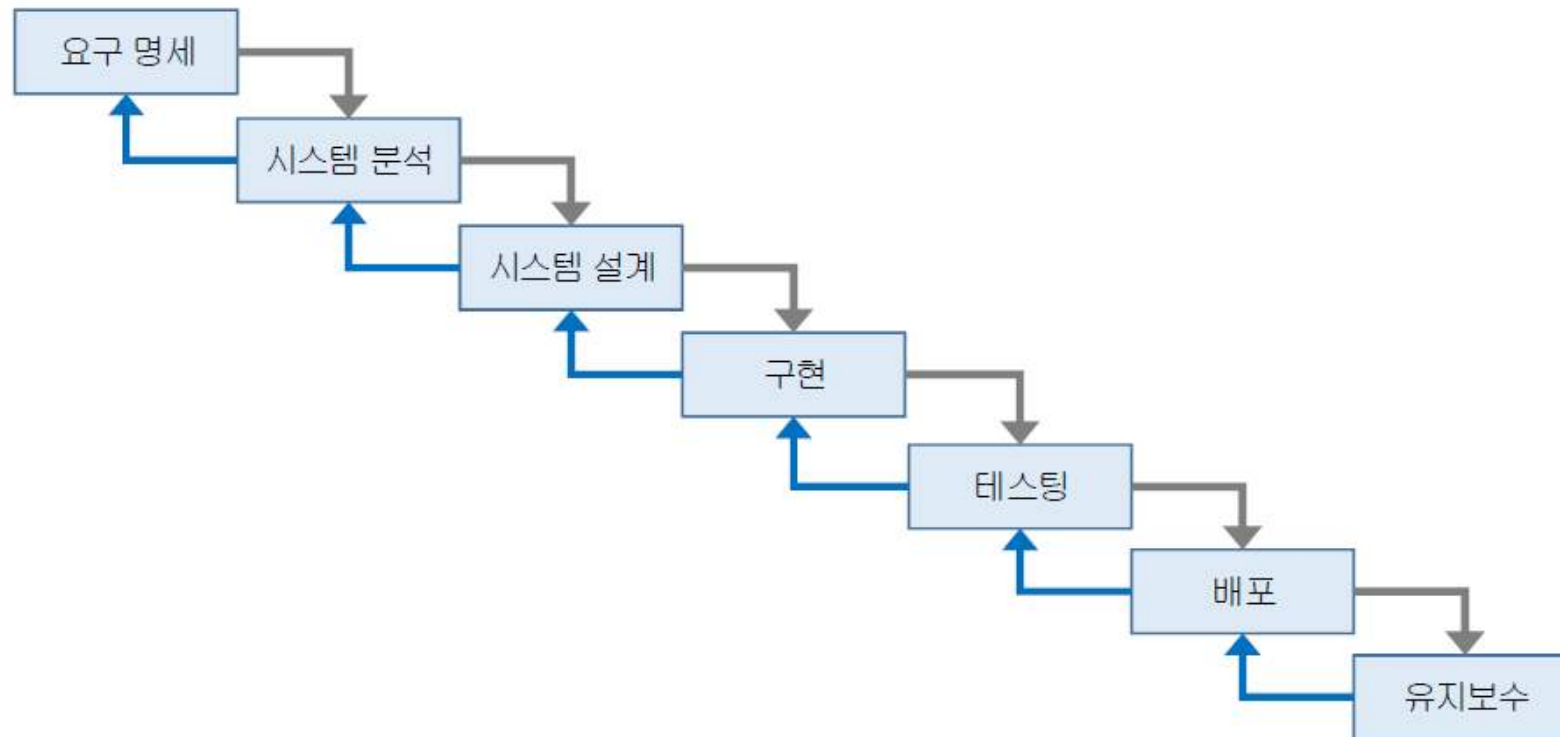
# 현재 시간의 시 값을 계산한다.
currentHour = totalHours % 24

# 결과를 출력한다.
print("현재 시간은" + str(currentHour) + ":"
      + str(currentMinute) + ":" + str(currentSecond) + " GMT")

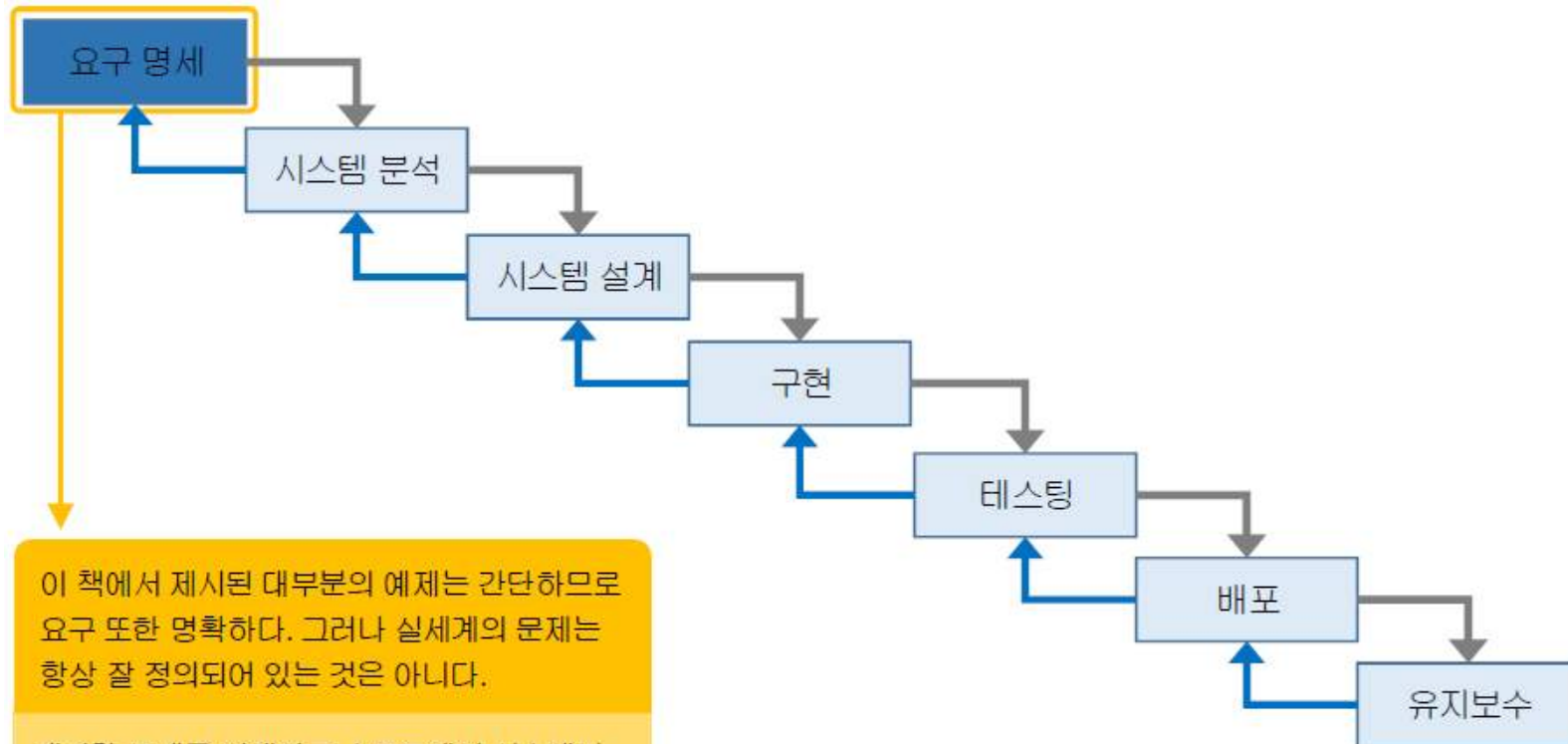
Ln: 12 Col: 18
```

소프트웨어 개발 과정

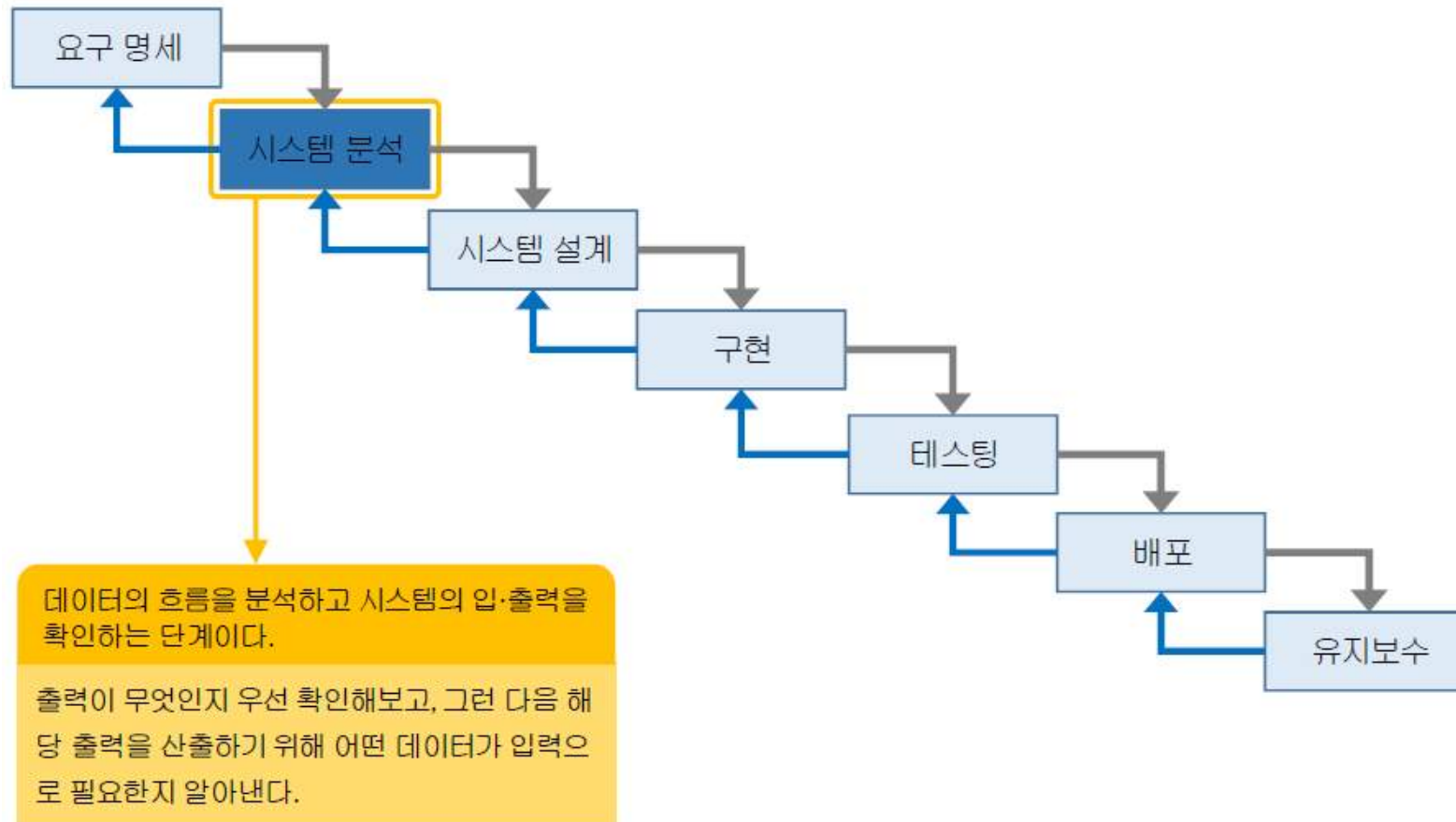
소프트웨어 개발 과정



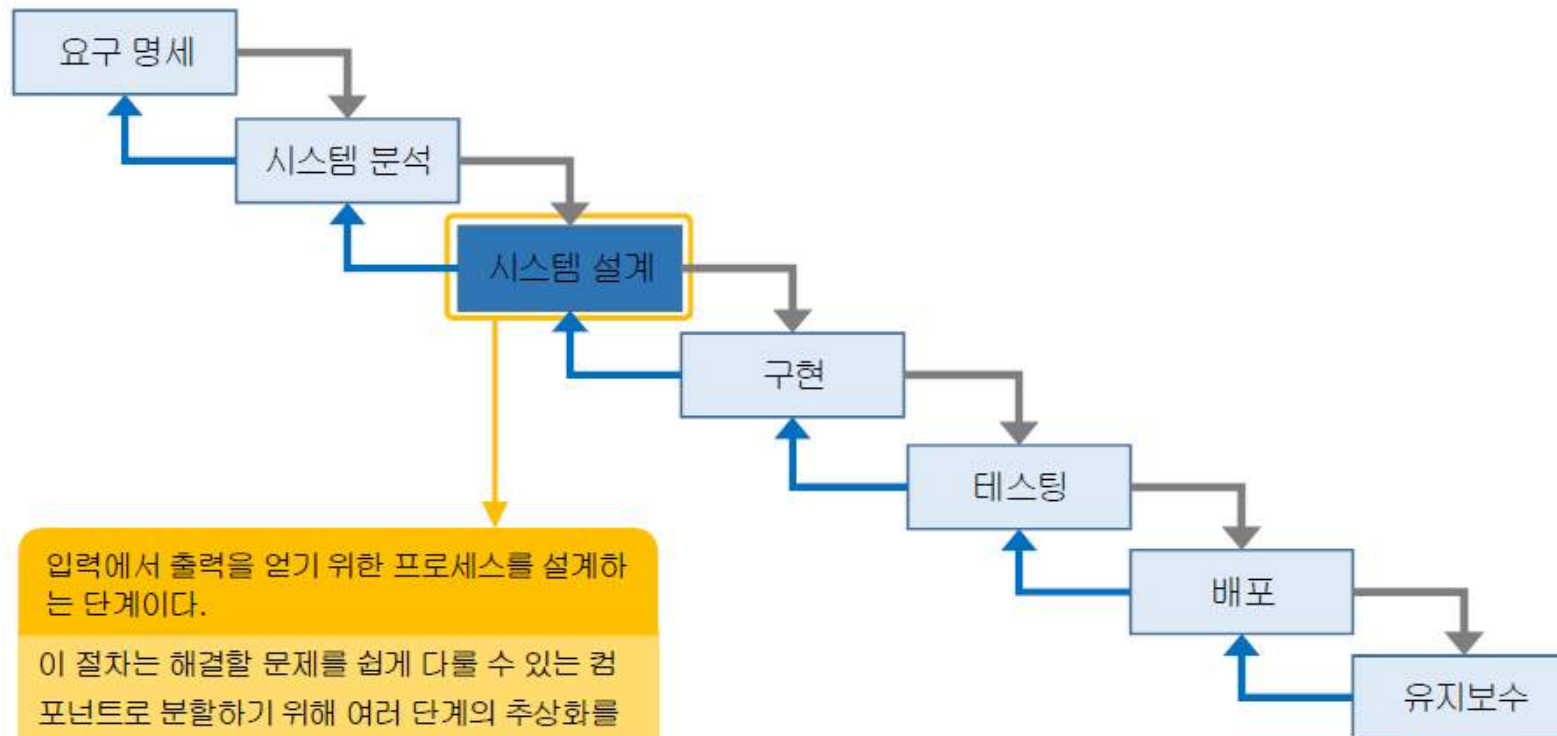
요구 명세



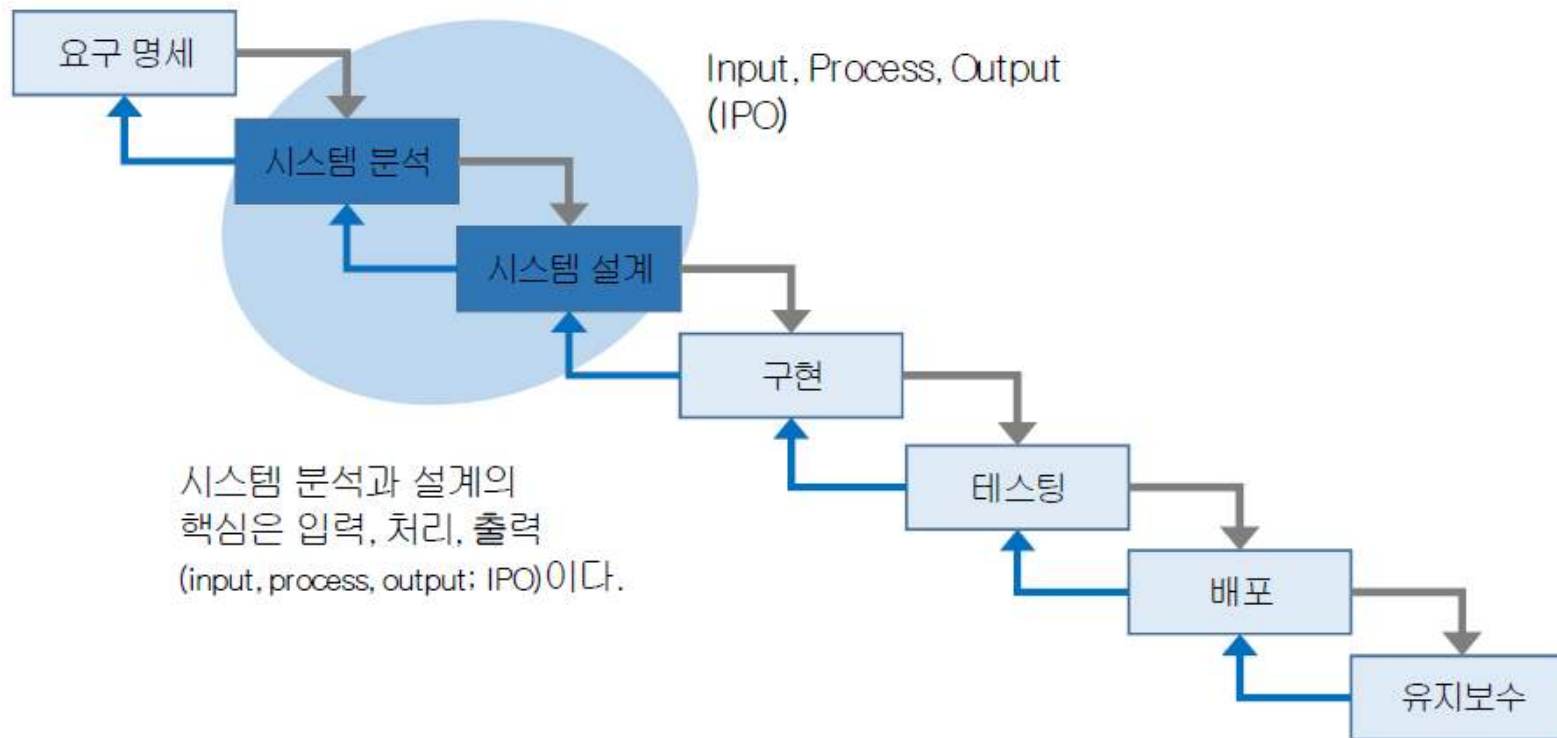
시스템 분석



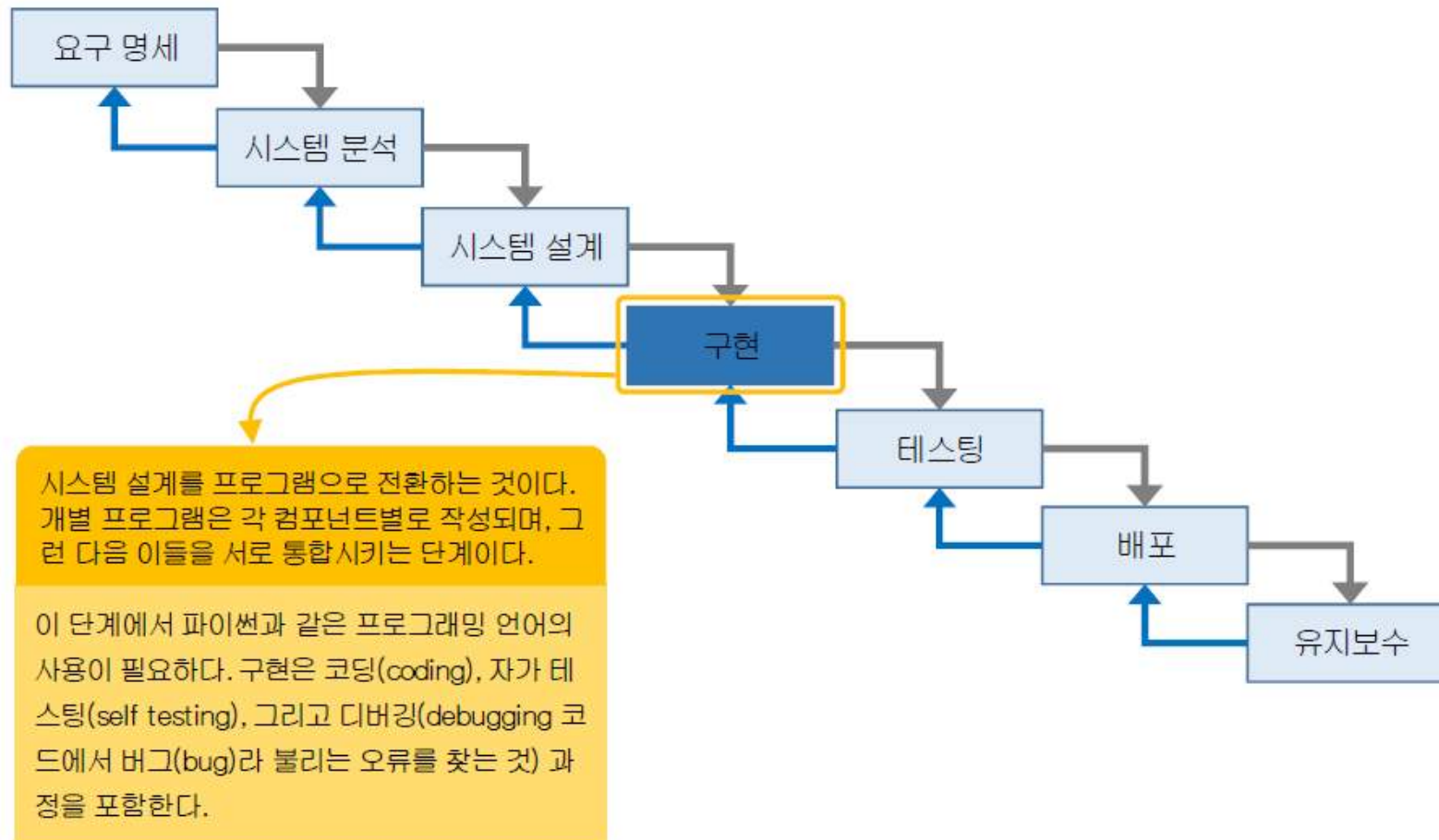
시스템 설계



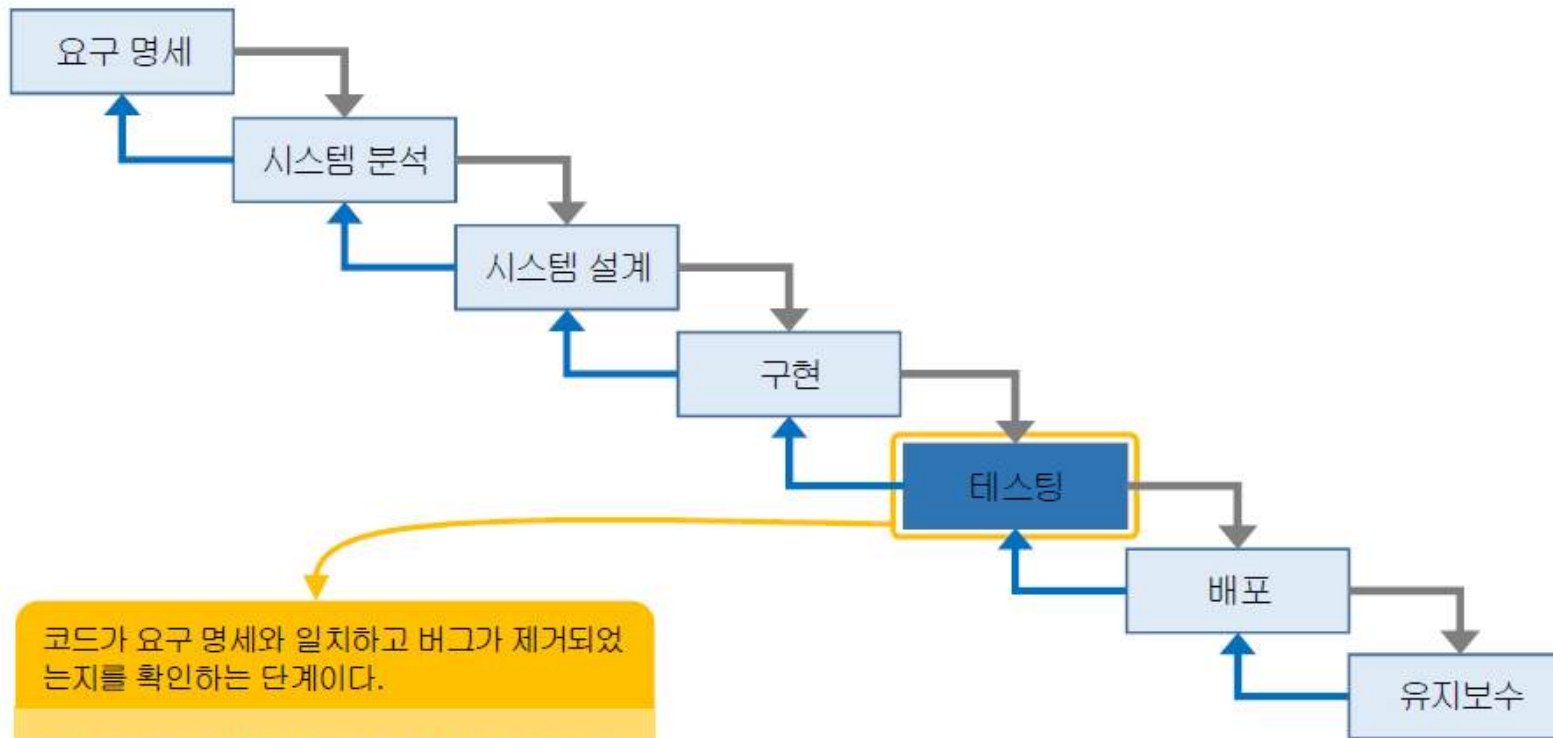
입력, 처리, 출력



구현



테스팅

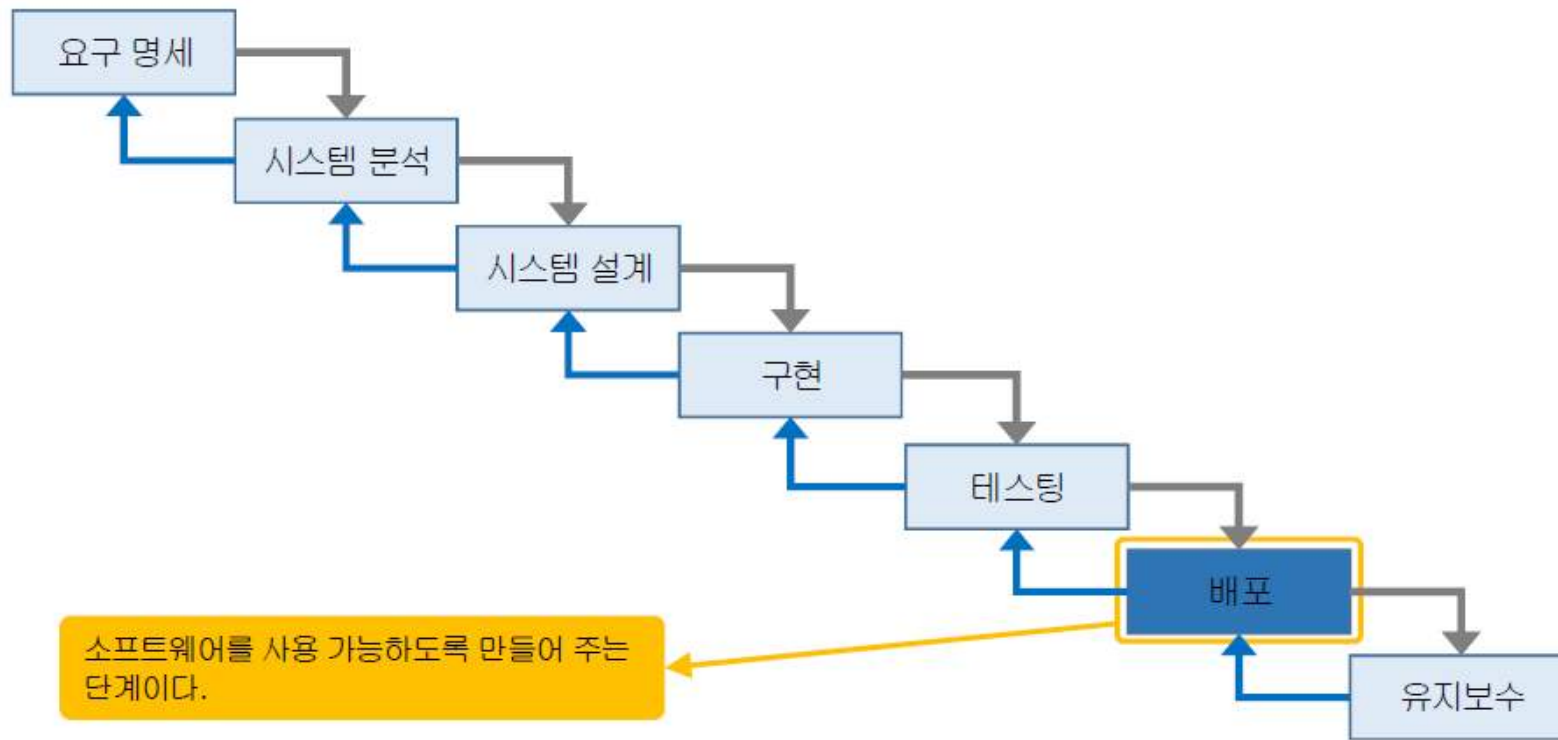


코드가 요구 명세와 일치하고 버그가 제거되었는지를 확인하는 단계이다.

일반적으로, 소프트웨어 제품의 설계와 구현에 관여하지 않은 소프트웨어 엔지니어들이 테스트를 수행한다.

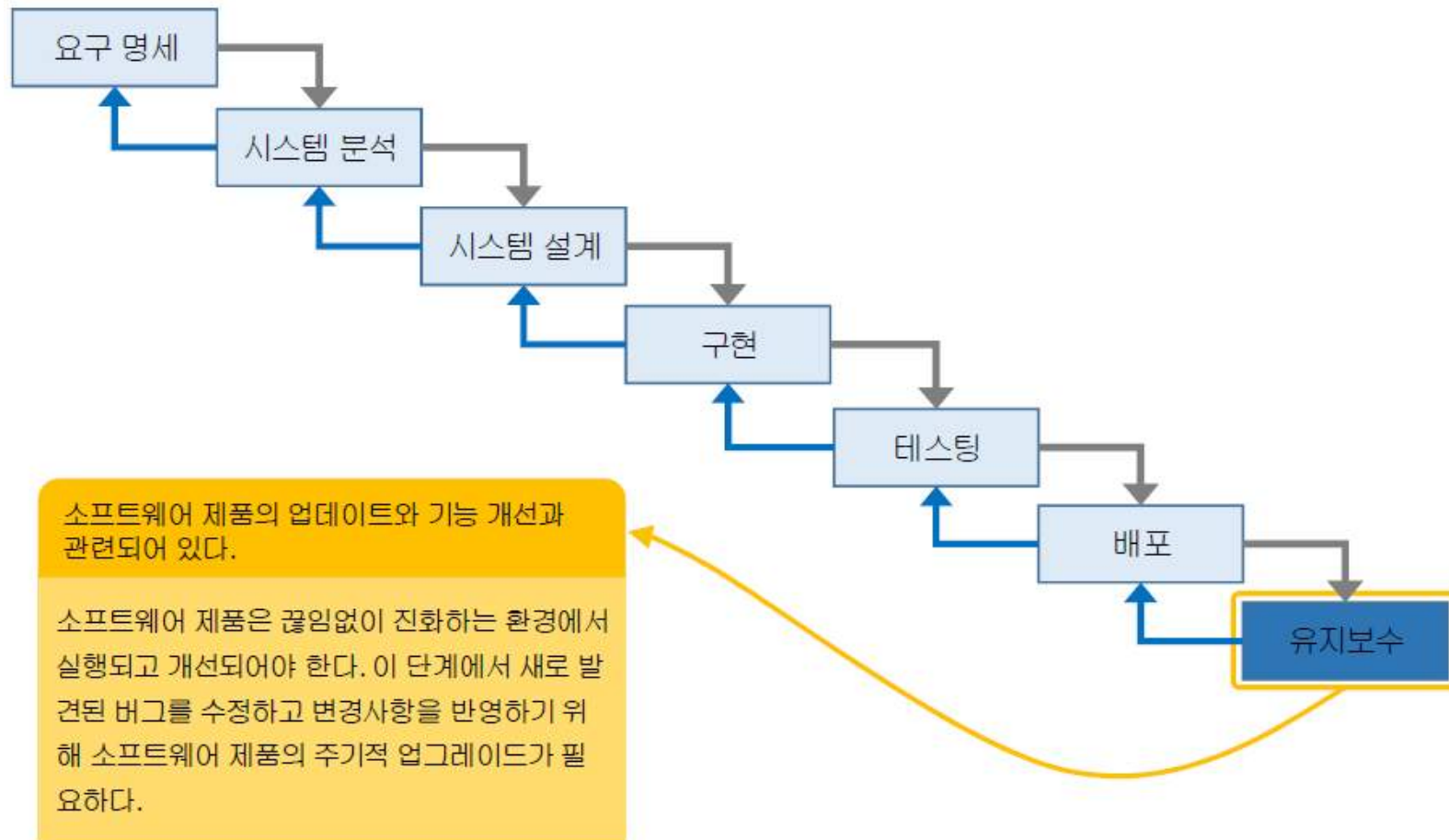
- QA (Quality Assurance) 팀

배포



- 배포 모델이 어떻게 달라져 왔을까?

유지 보수



예제 프로그램: 대출 상환금 계산

- 1. 요구 명세

- 사용자로부터 연이율, 대출금, 상환 년수를 입력 받도록 해야 한다
- 월상환액과 총상환액을 계산하고 화면에 출력해야 한다

- 2. 시스템 분석

- 출력은 월상환액과 총상환액
- 계산 수식

$$\text{월상환금} = \frac{\text{대출금} \times \text{월이율}}{1 - \frac{1}{(1 + \text{월이율})^{\text{상환년수} \times 12}}}$$

$$\text{총상환금} = \text{월상환금} \times \text{상환년수} \times 12$$

- 프로그램에 필요한 입력은 연이율, 대출금, 상환 년수

예제 프로그램: 대출 상환금 계산

- 3. 시스템 설계

- 단계 1: 사용자로부터 연이율, 상환 년수, 대출금을 입력 받는다
- 단계 2:
 - 연이율은 4.5%와 같은 퍼센트 서식의 숫자로 입력된다
 - 이 숫자를 100으로 나누어야 하고, 연이율로부터 월이율을 계산하기 위해 12로 다시 나누어야 한다
 - 그래서 월이율을 얻기 위해서는 퍼센트 서식인 연이율을 1200으로 나눈다
- 단계 3: 수식을 이용하여 월상환금을 계산한다
- 단계 4: 월상환금에 12를 곱하고 이것을 상환 년수와 곱하여 총상환금을 계산한다
- 단계 5: 월상환금과 총상환금을 화면에 출력한다

- 4. 구현

코드 예제

```
ComputeLoan.py - D:\GitRepo\ScriptProgramming_2016-2\ch2\ComputeLoan.py (3.5.2)
File Edit Format Run Options Window Help

# 연이율을 입력한다.
annualInterestRate = eval(input(
    "연이율을 입력하세요(예, 7.25): "))
monthlyInterestRate = annualInterestRate / 1200

# 상환년수를 입력한다.
numberOfYears = eval(input(
    "상환년수를 정수로 입력하세요(예, 5): "))

# 대출금을 입력한다.
loanAmount = eval(input("대출금을 입력하세요(예, 120000950: "))

# 총상환금을 계산한다.
monthlyPayment = loanAmount * monthlyInterestRate / (1
    - 1 / (1 + monthlyInterestRate) ** (numberOfYears * 12))
totalPayment = monthlyPayment * numberOfYears * 12

# 결과를 출력한다.
print(monthlyPayment)
print(int(monthlyPayment * 100))
print("월상환금은", int(monthlyPayment * 100) / 100, "입니다")
print("총상환금은", int(totalPayment * 100) / 100, "입니다.")
```

Ln: 8 Col: 8

예제 프로그램: 대출 상환금 계산

- 5. 테스트
 - 샘플 입력 데이터를 이용하여 프로그램 동작을 검사하고 출력이 정확한지 검증한다