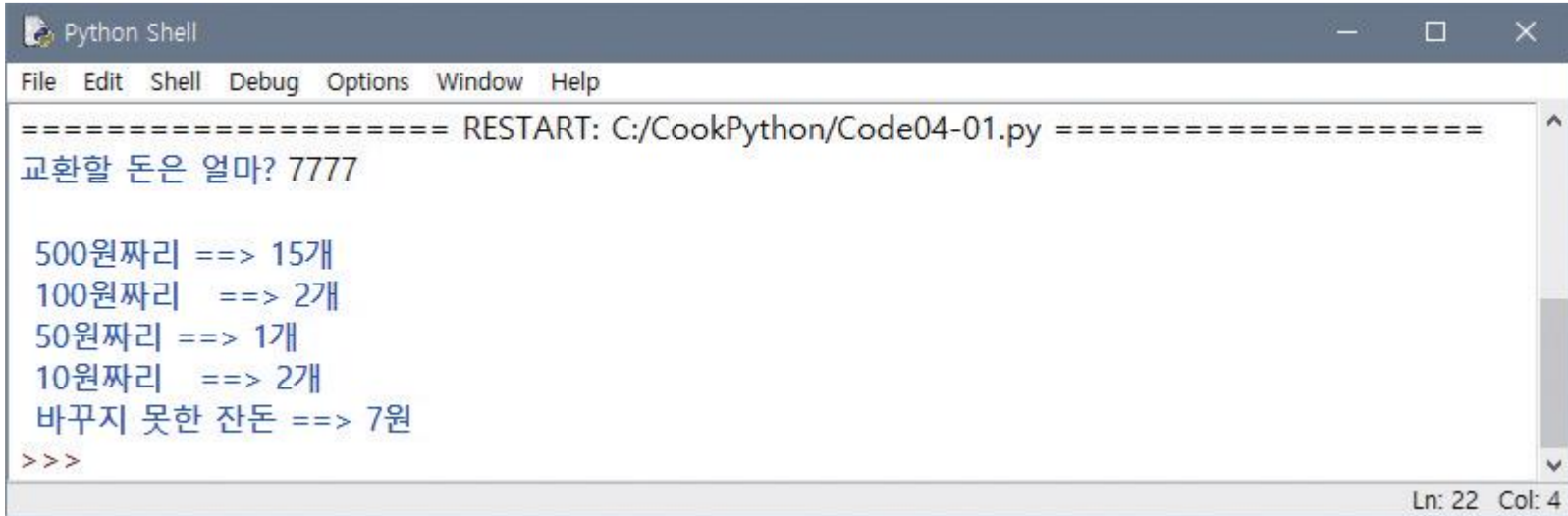


Section 01 이 장에서 만들 프로그램

■ [프로그램 1] 동전교환



The screenshot shows a Python Shell window with a menu bar (File, Edit, Shell, Debug, Options, Window, Help) and a text area containing the following text:

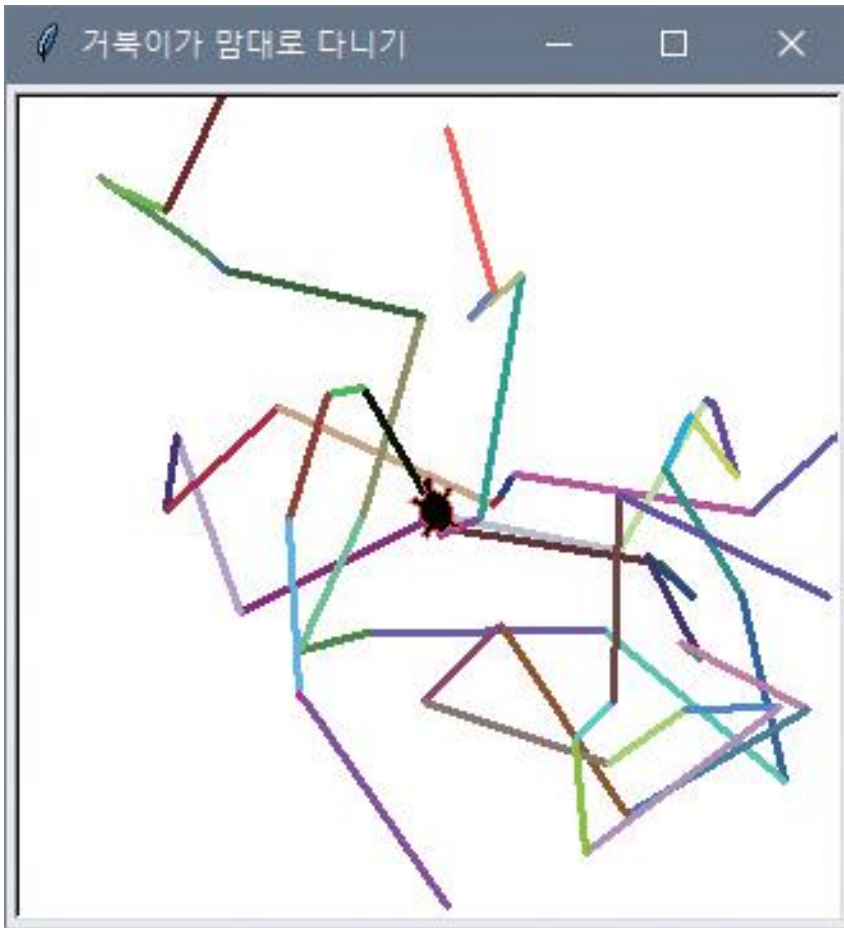
```
===== RESTART: C:/CookPython/Code04-01.py =====  
교환할 돈은 얼마? 7777  
  
500원짜리 ==> 15개  
100원짜리 ==> 2개  
50원짜리 ==> 1개  
10원짜리 ==> 2개  
바꾸지 못한 잔돈 ==> 7원  
>>>
```

The status bar at the bottom right indicates "Ln: 22 Col: 4".

Section 01 이 장에서 만들 프로그램

■ [프로그램 2] 마음대로 이동하는 거북이

- 거북이가 화 면 안에서 마음대로 이동하게 하는 프로그램.
- 단, 거북이가 화면을 벗어날 때는 다시 화면의 중앙으로 옮긴 후 마음대로 이동하도록 설정



■ 산술 연산자의 종류

표 4-1 산술 연산자의 종류

연산자	의미	사용 예	설명
=	대입 연산자	$a = 3$	정수 3을 a에 대입
+	더하기	$a = 5 + 3$	5와 3을 더한 값을 a에 대입
-	빼기	$a = 5 - 3$	5에서 3을 뺀 값을 a에 대입
*	곱하기	$a = 5 * 3$	5와 3을 곱한 값을 a에 대입
/	나누기	$a = 5 / 3$	5를 3으로 나눈 값을 a에 대입
//	나누기(몫)	$a = 5 // 3$	5를 3으로 나눈 후 소수점을 버리고 값을 a에 대입
%	나머지값	$a = 5 \% 3$	5를 3으로 나눈 후 나머지값을 a에 대입
**	제곱	$a = 5 ** 3$	5의 3제곱을 a에 대입

Section 02 산술 연산자

- $a//b$ 는 a 를 b 로 나눈 몫이고, $a\%b$ 는 a 를 b 로 나눈 나머지값

```
a = 5; b = 3  
print(a + b, a - b, a * b, a / b, a // b, a % b, a ** b)
```

출력 결과

```
8 2 15 1.6666666666666667 1 2 125
```

- Tip** • 세미콜론(;)은 앞뒤를 완전히 분리. 그러므로 $a=5; b=3$ 은 다음과 동일하다.
또 콤마(,)로 분리해서 값을 대입할 수도 있어 $a, b=5, 3$ 도 동일

```
a = 5  
b = 3
```

Section 02 산술 연산자

■ 산술 연산자의 우선순위

a, b, c = 2, 3, 4

```
print(a + b - c, a + b * c, a * b / c)
```

출력 결과

1 14 1.5

❶ $(a + b) - c$

❷ $a + (b - c)$

- 뿔셈에서는 계산되는 순서(연산자 우선순위)가 동일하므로 어떤 것을 먼저 계산하든 동일
- 특별히 괄호가 없을 때는 왼쪽에서 오른쪽 방향으로 계산

Section 02 산술 연산자

■ 산술 연산자의 우선순위

$$\textcircled{1} (a + b) * c \rightarrow (2 + 3) * 4 \rightarrow 5 * 4 \rightarrow 20$$

$$\textcircled{2} a + (b * c) \rightarrow 2 + (3 * 4) \rightarrow 2 + 12 \rightarrow 14$$

- 덧셈(또는 뺄셈)과 곱셈(또는 나눗셈)이 같이 있으면 곱셈(또는 나눗셈)이 먼저 계산된 후 덧셈(또는 뺄셈)이 계산
- 괄호가 없어도 $\textcircled{2}$ 처럼 계산
- 산술 연산자는 괄호가 가장 우선, 곱셈(또는 나눗셈)이 그 다음, 덧셈(또는 뺄셈)이 마지막
- 덧셈(또는 뺄셈)끼리 있거나 곱셈(또는 나눗셈)끼리 있으면 왼쪽에서 오른쪽으로

Tip • 덧셈, 뺄셈, 곱셈, 나눗셈이 함께 나오면 연산자 우선순위 때문에 종종 혼란스럽게 느껴진다. 이때는 괄호를 사용하면 된다. 괄호를 사용하면 무조건 괄호가 우선 계산, 두 번째 것이 더 나은 코딩

$$\textcircled{1} a = b + c * d;$$

$$\textcircled{2} a = b + (c * d);$$

■ 산술 연산을 하는 문자열과 숫자의 상호 변환

- 문자열이 숫자로 구성되어 있을 때, int() 또는 float() 함수 사용해서 정수나 실수로 변환
- 문자열을 int() 함수가 정수로, float() 함수가 실수로 변경

[illegible]

출력 결과

101 101.123 10000000000000000000000000000000

Section 02 산술 연산자

- 숫자를 문자열로 변환하려면 str() 함수 사용.
- a와 b가 문자열로 변경되어 100+1이 아닌 문자열의 연결인 '1001'과 '100.1231' 됨

```
a = 100; b = 100.123  
str(a) + '1'; str(b) + '1'
```

출력 결과

```
'1001'
```

```
'100.1231'
```

Tip • print() 함수는 출력 결과에 작은따옴표가 없어 문자열인지 구분하기가 어려워 사용하지 않음

■ 산술 연산자와 대입 연산자

- 대입 연산자 = 외에도 +=, -=, *=, /=, %=, //=, **= 사용
- 예) 첫 번째 대입 연산자 a+=3은 a에 3을 더해서 다시 a에 넣으라는 의미로 a=a+3과 같음

Tip • 파이썬에는 C/C++, 자바 등의 언어에 있는 증가 연산자 ++나 감소 연산자 --가 없음

표 4-2 대입 연산자의 종류

연산자	사용 예	설명
+=	a += 3	a = a + 3과 동일
-=	a -= 3	a = a - 3과 동일
*=	a *= 3	a = a * 3과 동일
/=	a /= 3	a = a / 3과 동일
//=	a //= 3	a = a // 3과 동일
%=	a %= 3	a = a % 3과 동일
**=	a **= 3	a = a ** 3과 동일

Section 02 산술 연산자

- a가 10에서 시작해 프로그램이 진행될수록 값이 누적

```
a = 10  
a += 5; print(a)  
a -= 5; print(a)  
a *= 5; print(a)  
a /= 5; print(a)  
a //= 5; print(a)  
a %= 5; print(a)  
a **= 5; print(a)
```

출력 결과

15 10 50 10.0 2.0 2.0 32.0

Section 02 산술 연산자

■ [프로그램 1]의 완성

- 학습한 연산자를 활용해서 동전 교환 프로그램 구현

Code04-01.py

```
1  ## 변수 선언 부분 ##
2  money, c500, c100, c50, c10 = 0, 0, 0, 0, 0
3
4  ## 메인 코드 부분 ##
5  money = int(input("교환할 돈은 얼마?"))
6
7  c500 = money // 500
8  money %= 500
9
10 c100 = money // 100
11 money %= 100
12
13 c50 = money // 50
14 money %= 50
```

2행 : 동전으로 교환할 돈(money)과 500원, 100원, 50원, 10원짜리 동전의 개수를 저장 할 변수 초기화
7행 : 500원짜리 동전의 개수를 구함
8행 : 다시 money를 500으로 나눈 후 나머지 값 저장
8행의 money%=500은 money=money%500과 동일.
10~11행 : 100원짜리 동전을, 13~14행에서 50원짜리 동전을, 16~17행에서 10원짜리 동전을 구함