Chapter 04
데이터와 변수

목차

학습목표

- 데이터에 대하여 이해합니다.
- 변수를 선언하고 초기화하는 방법을 알아봅니다.
- 변수명을 지을 때 따라야 할 규칙을 알아봅니다.
- 자료형의 종류와 자료형 변환 방법을 알아봅니다.
- 입력한 데이터를 변수에 저장하여 활용하는 방법을 알아봅니다.

Section 01

이 장에서 만들 프로그램

이 장에서 만들 프로그램

1. 비밀번호 발송 메일 프로그램

▪비밀번호를 잊어버린 고객에게 웹 사이트 관리자가 다음과 같은 형식으로 아이디와 비밀번호를 알려줌

```
To. gildong@abc.com
▶ 아이디 및 비밀번호 확인
  홍길동 고객님 안녕하세요.
  홍길동 고객님의 아이디는 비밀번호는 아래와 같습니다.
  아이디 : gildong
  비밀번호 : 1234
```

이 장에서 만들 프로그램

2. 날씨 예보 프로그램

▪사용자가 날짜, 아침 기온, 낮 기온, 비올 확률, 미세먼지 수치, 일출 시간, 일몰 시간 등의 정보를 입력하면 내일 날씨 예보를 해줌

```
내일 날씨 예보입니다.
월요일인 3월 30일의 아침 최저 기온은 -1도, 낮 최고 기온은 10도로 예보됐습니다.
비올 확률은 45%이고, 미세먼지는 좋음 수준일 것으로 예상됩니다.
일출 시간은 오전 6시 30분이고, 일몰 시간은 오후 7시 20분입니다.
바다의 물결은 남해 앞바다 0.5m, 동해 앞바다 1.5m, 서해 앞바다 0.5m 높이로 일겠습니다.
지금까지 3월 30일 월요일 날씨 예보였습니다.
```

Section 02
데이터란?

데이터와 정보

- 데이터
  - 어떤 물체나 현상을 측정하거나 관찰하여 얻어진 값
  - 알람 시계의 '시간' 데이터, 스마트폰 날씨 앱의 '기상청의 날씨' 데이터 등

- 정보
  - 특정 데이터를 모아서 의미 있고 관리하기 쉽게 정리해 놓은 것
  - 즉 데이터란 정보를 이루고 있는 구성 요소를 의미함

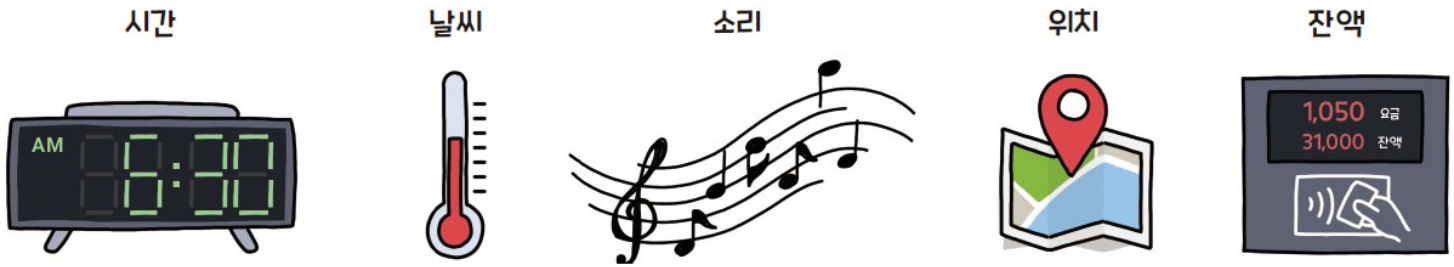


그림 4-1 일상생활에서 사용하는 다양한 데이터

데이터는 어디에 저장될까?

- 컴퓨터 프로그램 속 데이터
  - 숫자, 문자 같은 단순 데이터부터 사진, 음원 등의 복합 데이터까지 다양함
  - 데이터는 정확해야 하고 필요할 때 언제든지 수정할 수 있어야 함
  - 또한 사용 목적에 따라 다른 형태로 가공할 수도 있어야 함

그림 4-2 컴퓨터를 이용한 데이터 처리 과정

데이터는 어디에 저장될까?

- 메모리
  - 사용자가 입력한 데이터, 인터넷을 통해 전송 받은 데이터 등 컴퓨터에서 다루는 모든 데이터는 메모리에 저장됨
  - 컴퓨터의 메모리는 수많은 방들로 이루어져 있으며 각 방마다 하나의 데이터를 저장함
  - 이때 데이터는 숫자, 문자, 사진, 소리 등으로 사용자는 언제든지 데이터를 생성, 수정, 복사, 삭제할 수 있음

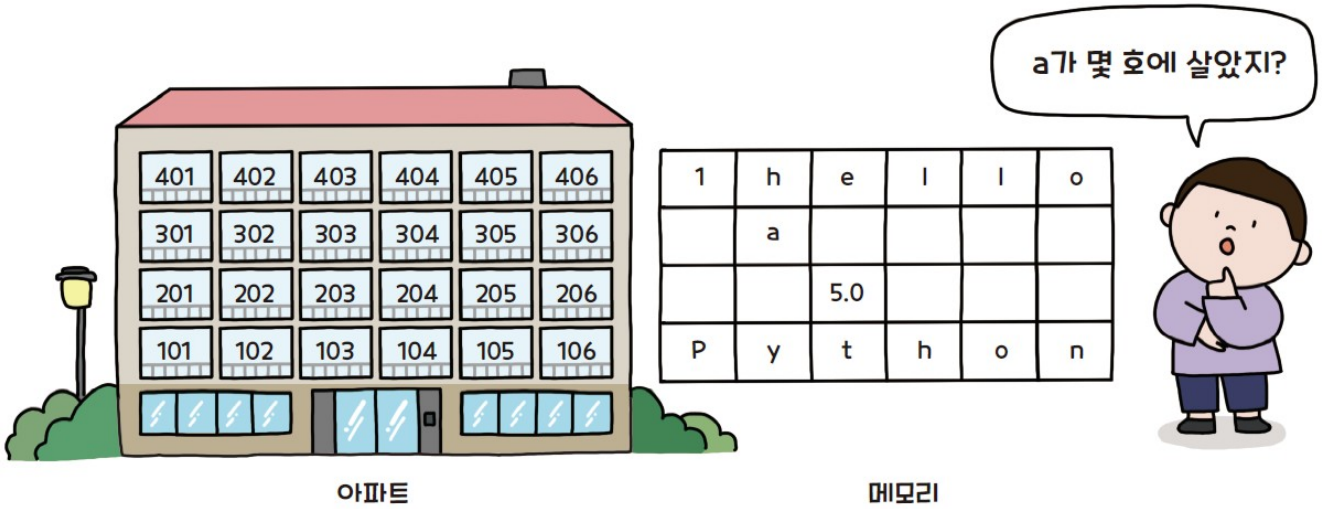


그림 4-3 메모리의 구조

확인문제

확인문제

다음 빈 칸에 공통적으로 들어갈 단어를 쓰시오.

- 정보를 이루고 있는 요소로 어떤 물체나 현상을 측정하거나 관찰하여 얻어진 값을 [ ](이)라고 한다.
- 컴퓨터에서 연산 처리를 위해서는 기본적으로 [ ]이/가 있어야 한다.

정답

데이터

Section 03
변수란?

변수와 메모리

- 메모리 주소
  - 아파트에 동, 호수가 있듯이 메모리에도 각 방마다 주소가 있음
  - 실제로 프로그래밍할 때는 메모리 주소 대신 이름을 붙여 사용함

- 변수
  - 메모리 방마다 부여한 이름을 변수라고 함
  - 즉 변수는 메모리 방의 이름을 의미함

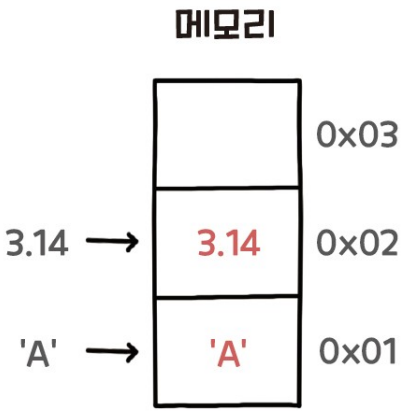


그림 4-4 메모리 내 데이터 저장 방법

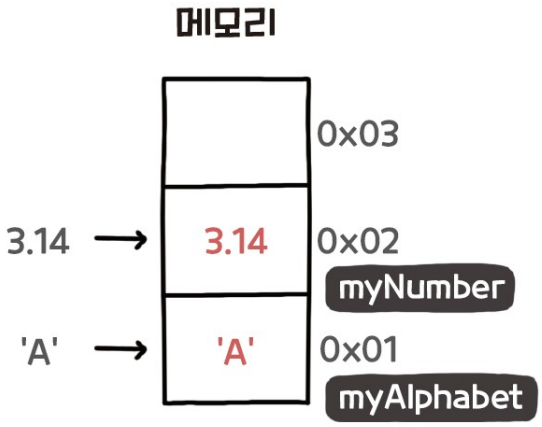


그림 4-5 변수의 개념

변수 저장

변수 저장하고 사용하기

- 1. 파이썬 셸을 실행하고 다음과 같이 입력함

```
>>> myAlphabet = 'A'
```

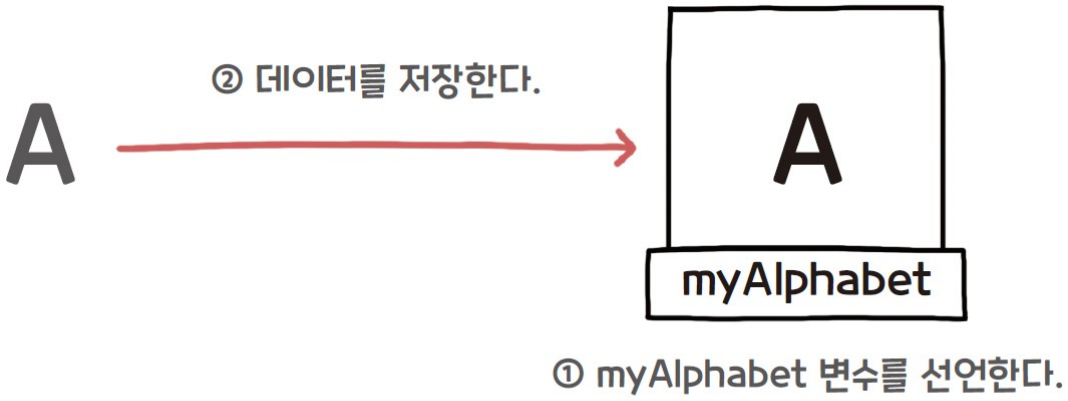


그림 4-6 변수에 데이터를 저장하는 방식

변수 저장

변수 저장하고 사용하기

- 2. [Enter] 클릭하여 출력 결과 확인

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:20:19) [MSC v.1925 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> myAlphabet = 'A'
>>>
```

그림 4-7 변수에 데이터를 저장한 후 출력 결과

- 3. 저장된 데이터 'A'를 출력하기 위해 'print(변수명)'을 입력

```
>>> print(myAlphabet)
A
```

그림 4-8 print(myAlphabet) 명령 실행 과정

문제 해결

문제 해결 4-1 자신의 이름과 전공 출력하기 ch04_sol_01.py

변수 myName과 myMajor에 자신의 이름과 전공을 저장하고 출력해봅시다.

```
01   myName = 'Hong gil dong'
02   myMajor = 'Computer'
03   print(myName)
04   print(myMajor)
```

```
Hong gil dong
Computer
```

Section 04
변수 사용법

변수 선언과 초기화

- 변수 선언
  - 메모리에 비어 있는 방을 깨끗하게 청소하고 이름표를 붙여 데이터를 저장할 준비를 하는 단계

- 변수 초기화
  - 만들어진 변수(메모리 방)에 데이터를 저장하는 단계
  - 데이터를 저장하기 위해서 할당 연산자(=)를 사용함
  - 할당 연산자 '='는 우측의 데이터를 좌측의 변수에 대입하라는 뜻

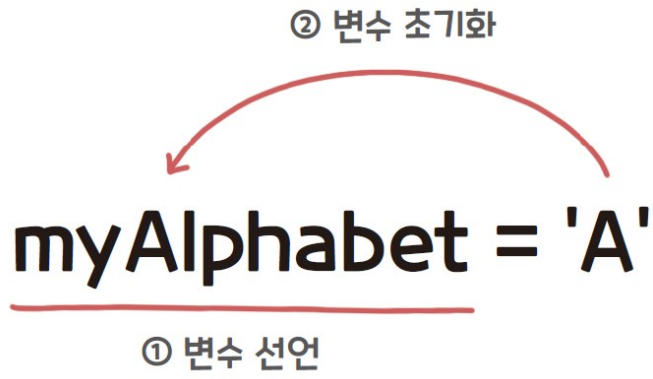


그림 4-9 변수 선언과 초기화

변수 선언과 초기화

- 변수를 사용하지 않고 'Hello Python' 출력하기

```
>>> print('Hello Python')
Hello Python
```

- 변수 intro를 선언하고 'Hello Python'으로 초기화한 후 출력하기

```
>>> intro = 'Hello Python'
>>> print(intro)
Hello Python
```

변수 선언과 초기화

- 변수의 특징
  - 변수에 저장된 데이터는 언제든지 바뀔 수 있으나 새로운 데이터가 저장되면 과거의 데이터는 사라짐
  - 변수를 초기화할 때와 마찬가지로 할당 연산자 '='를 이용해서 새로운 데이터를 저장함

- 변수 myAlphabet에 저장된 'A'를 'ABC'로 변경한 후 출력하기

```
>>> myAlphabet = 'A'
>>> print(myAlphabet)
A
>>> myAlphabet = 'ABC'
>>> print(myAlphabet)
ABC
```

문제 해결

문제 해결 4-2 **변수 값 변경하기** ch04_sol_02.py

다음 순서에 맞추어 코드를 작성해봅시다. 아래 코드를 가리고 작성한 후 자신이 작성한 코드와 비교해봅니다.

① intro 변수를 선언하고 'Hello'로 초기화한다.

② intro 변수에 저장된 값을 화면에 출력한다.

③ intro 변수의 데이터를 '안녕하세요.'로 변경한다.

④ 변경된 값을 화면에 출력한다.

```
01   intro = 'Hello'
02   print(intro)
03   intro = '안녕하세요.'
04   print(intro)
```

```
Hello
안녕하세요.
```

변수를 초기화하지 않아도 될까?

- 변수를 선언만 하고 초기화하지 않을 경우
  - 파이썬 셸에 nickname 변수만 입력하고 [Enter]를 누름 → 에러 발생
  - nickname 변수의 선언은 되지만 초기화가 이루어지지 않았기 때문임

```
>>> nickname
Traceback (most recent call last):
  File "<pyshell#18>", line 1, in <module>
    nickname
NameError: name 'nickname' is not defined
```

변수를 초기화하지 않아도 될까?

- 변수를 정의할 때는 변수 선언과 초기화를 한 세트로 해주어야 함

```
>>> nickname = 'Mr. Hong'
>>> print(nickname)
Mr. Hong
```

- 초기화한 변수에 다시 아무런 데이터도 할당하지 않을 경우
  - 정상 실행

```
>>> nickname
'Mr. Hong'
```

Section 05
변수명 규칙

변수명 규칙

영문자를 사용함

```
>>> hongAge = 20
>>> print(hongAge)
20
```

권장

```
>>> 홍길동나이 = 20
>>> print(홍길동나이)
20
```

권장하지 않음

소문자로 시작함

- 클래스의 이름을 대문자로 시작하기 때문에 클래스명과 구분짓기 위해서 소문자로 시작하는 것이 좋음

```
>>> money = 100
>>> print(money)
100
```

권장

```
>>> Money = 100
>>> print(Money)
100
```

권장하지 않음

변수명 규칙

- 데이터의 의미를 쉽게 파악할 수 있도록 지음
  - 해당 변수에 어떤 데이터가 저장되는지 쉽게 알 수 있도록 짓는 것이 좋음
  - 예를 들어 게임 프로그램에서 점수는 score, 플레이어는 player 등

- 두 개 이상의 단어가 조합될 경우 낙타표기법을 따름
  - 낙타표기법 : 첫 번째 단어는 소문자로 시작하고, 두 번째 단어부터는 첫 글자만 대문자로 표기하는 방법

그림 4-10 낙타표기법에 따른 변수 이름 짓기

변수명 규칙

- 예약어는 변수명으로 사용할 수 없음
  - 파이썬에 이미 예약된 단어들은 변수명으로 사용할 수 없음

True	False	None	if	elif
continue	def	finally	else	for
pass	while	with	try	except
break	class	return	import	as

그림 4-11 파이썬의 예약어

  - 더 많은 예약어 확인하기

```
>>> import keyword
>>> print(keyword.kwlist)
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class',
'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from',
'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass',
'raise', 'return', 'try', 'while', 'with', 'yield']
```

변수명 규칙

- 언더바(_)를 제외한 특수문자는 사용할 수 없음

```
>>> !score = 10
>>> sc!ore = 10
>>> $score = 10
>>> sc$ore = 10
```

특수문자를 사용한 변수명

Enter

```
SyntaxError: invalid syntax
```

구문 에러 발생

- 특수문자 언더바(_)는 변수명의 위치에 상관없이 자유롭게 사용할 수 있음

```
>>> _var = 10
>>> print(_var)
10
```

변수명 앞에 사용한 경우

```
>>> v_a_r = 10
>>> print(v_a_r)
10
```

변수명 중간에 사용한 경우

```
>>> var_ = 10
>>> print(var_)
10
```

변수명 뒤에 사용한 경우

변수명 규칙

- 공백문자(' ')는 사용할 수 없음
  - 공백문자를 사용하면 구문 에러(SyntaxError)가 발생함

```
>>> v ar = 10
SyntaxError: invalid syntax
```

(공백)

  - 공백문자는 변수명의 앞이나 뒤에 와서도 안 됨

```
>>>  var = 10
SyntaxError: unexpected indent
```

(공백)

공백문자가 변수명 앞에 온 경우

```
>>> var  = 10
>>> print(var)
10
```

(공백)

공백문자가 변수명 뒤에 온 경우

변수명 규칙

- 숫자는 첫 글자를 제외한 나머지 자리에서만 사용함

```
>>> 1player = 'Mr. Hong'
SyntaxError: invalid syntax
```

처음에 사용한 경우

```
>>> p1layer = 'Mr. Hong'
>>> print(p1layer)
Mr. Hong
```

중간에 사용한 경우

```
>>> player1 = 'Mr. Hong'
>>> print(player1)
Mr. Hong
```

뒤에 사용한 경우

확인 문제

확인문제

1. 현재 시간을 저장하기 위한 변수명로 가장 적합한 것은 무엇인가?

① `CurrentTime`　② `CT`
③ `currentTime`　④ `CTime`

2. 다음 중 가독성이 가장 좋은 변수명은 무엇인가?

① `rentalbook`　② `rentalBicycle`
③ `Rental Car`　④ `RentalService`

3. 다음 중 사용 가능한 변수명은 무엇인가?

① `$myScore`　② `*score!`
③ `_score_`　④ `↑highScore`

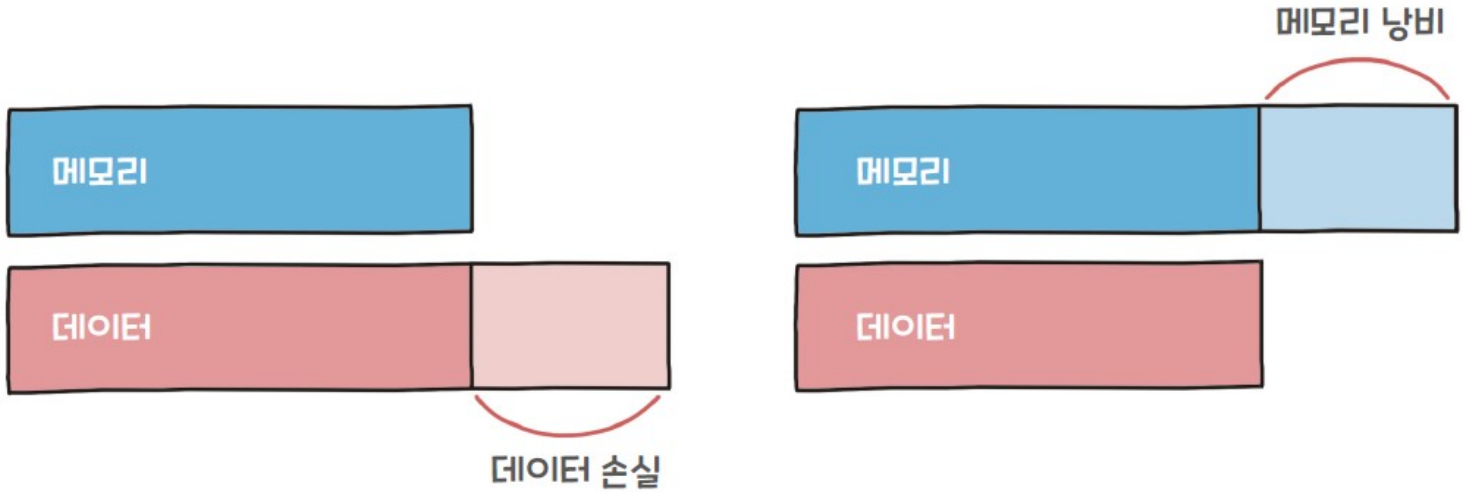
정답

1. ③　2. ②　3. ③

Section 06
자료형이란?

데이터 크기와 메모리

- 데이터 손실
  - 작은 방에 큰 데이터를 저장하여 데이터가 넘쳐 일부가 손실되는 것
- 메모리 낭비
  - 크기가 작은 데이터를 큰 방에 저장하면 프로그램은 정상적으로 실행됨
  - 하지만 이런 경우 메모리가 낭비되며 최악의 경우 전체 메모리가 부족해서 시스템이 멈출 수 있음



(a) 데이터가 메모리보다 큰 경우–데이터 손실

(b) 데이터가 메모리보다 작은 경우–메모리 낭비

그림 4-12 데이터 손실과 메모리 낭비

데이터 크기와 메모리

- 메모리 효율적 관리
  - 데이터의 크기에 따라서 메모리의 크기를 적절하게 부여하기
  - 이를 위해 대부분의 프로그래밍 언어에서는 데이터를 자료형으로 분류하고 그에 따른 적절한 메모리 크기를 자동으로 할당함

- 자료형(data type)
  - 데이터를 정수형(integer), 실수형(float), 문자형(string), 논리형(bool) 등으로 분류하는 것을 의미함

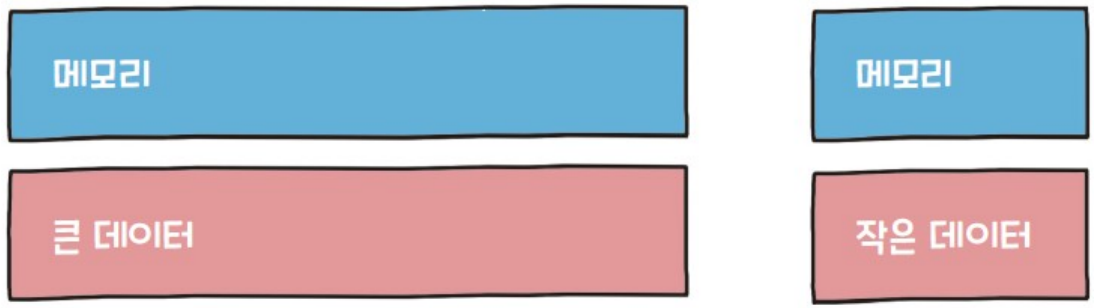


그림 4-13 적절한 메모리 할당

자료형에 따른 변수 초기화 방법

문자형

- 파이썬에서는 한 글자만 나타내는 문자형과 여러 글자가 모인 문자열형을 동일한 자료형으로 간주함
- 문자형 : 작은따옴표(' ')를 이용하여 변수에 저장함
- 문자열의 시작과 끝에 작은따옴표가 하나라도 없으면 에러가 발생함

```
>>> myUniversity = 'Hanguk University'
>>> print(myUniversity)
Hanguk University
```

하나 더 알기 **큰따옴표(" ")와 작은따옴표(' ')의 차이**

문자열을 저장할 때 큰따옴표(“ ”)를 사용해도 됩니다. 다른 프로그램 언어(C, C++, Java 등)에서는 작은따옴표와 큰따옴표를 엄격하게 구분해서 사용하지만 파이썬에서는 어떤 것을 사용해도 문제없습니다. 하지만 작은따옴표와 큰따옴표를 혼용하면 안 됩니다. 예를 들어 열 때는 작은따옴표, 닫을 때는 큰따옴표(‘ ”)를 사용하면 에러가 발생합니다.

자료형에 따른 변수 초기화 방법

숫자(정수, 실수)형

- 변수에 숫자를 저장할 때는 단순하게 숫자만 입력함

```
>>> myNumber = 10
>>> print(myNumber)
10
>>> myNumber = 3.14
>>> print(myNumber)
3.14
```

- 정수형 : 소수점이 없는 숫자 데이터

```
>>> myNum1 = 123
>>> print(myNum1)
123
```

```
>>> myNum2 = 12345678901234567890123456789012345678901234567890
>>> print(myNum2)
12345678901234567890123456789012345678901234567890
```

자료형에 따른 변수 초기화 방법

- 숫자(정수, 실수)형
  - 실수형 : 소수점이 있는 숫자 데이터를 의미함
    - 따라서 정수형을 의도했더라도 3.0을 입력하면 실수형으로 인식함
    - 실수형은 메모리의 크기에 제약을 받기 때문에 소수점 16번째 자리까지만 메모리에 저장하고 나머지는 손실됨

```
>>> myNum1 = 3.12
>>> print(myNum1)
3.12
```

```
>>> myNum2 = 3.12345678901234567890123456789012345678901234567890
>>> print(myNum2)
3.1234567890123457
```

자료형에 따른 변수 초기화 방법

- 숫자(정수, 실수)형
  - 숫자와 문자의 구분
    - var1 : **정수 777**이 저장됨
    - var2 : **문자열 '777'**이 저장됨

```
>>> var1 = 777
>>> var2 = '777'
```

```
>>> var1 = var1 + 1          ← 정수 + 1
>>> print(var1)
778
>>> var2 = var2 + 1          ← 문자열 + 1
Traceback (most recent call last):
  File "<pyshell#24>", line 1, in <module>
    var2 = var2 + 1
TypeError: can only concatenate str (not "int") to str
```

자료형에 따른 변수 초기화 방법

논리형

- '참'과 '거짓'을 나타내는 논리 데이터인 True와 False가 있음
- True와 False도 데이터이므로 변수에 저장할 수 있음

```
>>> flag = True
>>> print(flag)
True
>>> flag = False
>>> print(flag)
False
```

논리형 자료형의 첫 글자는 반드시 대문자(True, False)로 써야 합니다. 소문자 'true' 또는 'false'로 쓰면 에러가 발생합니다.

확인 문제

확인문제

1. 다음 중 올바르게 코딩된 코드는 무엇인가?

   ① `varStr = 'Hello Python"`
   ② `varInt = "123'`
   ③ `varFlt = 3.14`
   ④ `varBool = 'False"`

2. 다음은 'I am a boy.' 문자열을 wordVar 변수에 저장하고 출력하는 프로그램이다. 잘못된 부분을 찾아 고치시오.

```
>>> wordVar = I am a boy.
>>> print(wordVar)
I am a boy.
```

정답

1. ③　　2. `>>> wordVar = 'I am a boy.'`

Section 07
자료형 조회와 변환

자료형 조회

type( ) 함수

- 변수에 저장된 데이터가 어떤 자료형인지 알기 위해 사용하는 함수

```
>>> type(123)
<class 'int'>
```
정수형

```
>>> type('123')
<class 'str'>
```
문자형

```
>>> myVar = 3.14
>>> type(myVar)
<class 'float'>
>>> myVar = True
>>> type(myVar)
<class 'bool'>
```
실수형

논리형

확인 문제

확인문제

다음 코드를 보고 빈 칸에 자료형을 채우시오.

```
>>> type('Hello Python')
<class '          '>
>>> type(123)
<class '          '>
>>> type(3.14)
<class '          '>
>>> type(True)
<class '          '>
```

정답

str, int, float, bool

자료형 변환

자료형 변환

- 자료형은 서로 간에 변환이 가능하며 '자료형 변환' 또는 줄여서 '형 변환'이라고도 함
- 자료형을 변환하기 위해서는 함수를 사용함
- 각 함수의 괄호 안에 데이터를 넣으면 데이터의 자료형이 바뀜

표 4-1 자료형 변환 함수

함수	내용	사용 예
str()	괄호 안의 데이터를 문자로 변환	str(10), str(3.14), str(True)
int()	괄호 안의 데이터를 정수로 변환	int('10'), int(True)
float()	괄호 안의 데이터를 실수로 변환	float('3.14')
bool()	괄호 안의 데이터를 논리형으로 변환	bool('True'), bool(0)

자료형 변환

자료형 변환

정수형 10을 문자로 변환하기

```
>>> myVar = 10
>>> type(myVar)
<class 'int'>
>>> myVar = str(myVar)
```

```
>>> type(myVar)
<class 'str'>
```

문자 '123'을 정수와 실수로 변환하기

```
>>> myVar = '123'
>>> type(myVar)
<class 'str'>
>>> myVar = int(myVar)
>>> type(myVar)
<class 'int'>
```

정수형으로 변환

```
>>> myVar = '123'
>>> type(myVar)
<class 'str'>
>>> myVar = float(myVar)
>>> type(myVar)
<class 'float'>
```

실수형으로 변환

자료형 변환

자료형 변환

문자형을 논리형으로 변환하기

- 빈 문자('')를 논리형으로 바꾸면 False로 변환되고, 그 외 나머지 문자는 모두 True로 변환됨
- 공백문자(' ')도 True로 변환됨

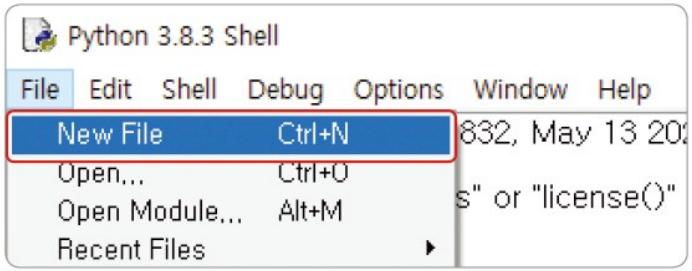
```
>>> myVar = ''
>>> type(myVar)
<class 'str'>
>>> myVar = bool(myVar)
>>> type(myVar)
<class 'bool'>
>>> print(myVar)
False
```

```
>>> myVar = 'Hello'
>>> type(myVar)
<class 'str'>
>>> myVar = bool(myVar)
>>> type(myVar)
<class 'bool'>
>>> print(myVar)
True
```

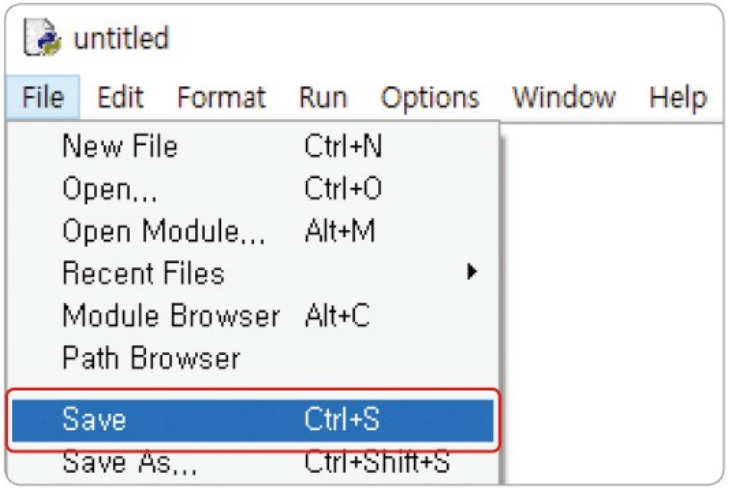
Section 08
변수의 활용

데이터 재활용

- ‘Hello Python!’을 5번 출력하는 프로그램 만들기
  - 1. 파이썬 셸에서 [File]-[New File] 메뉴를 선택함

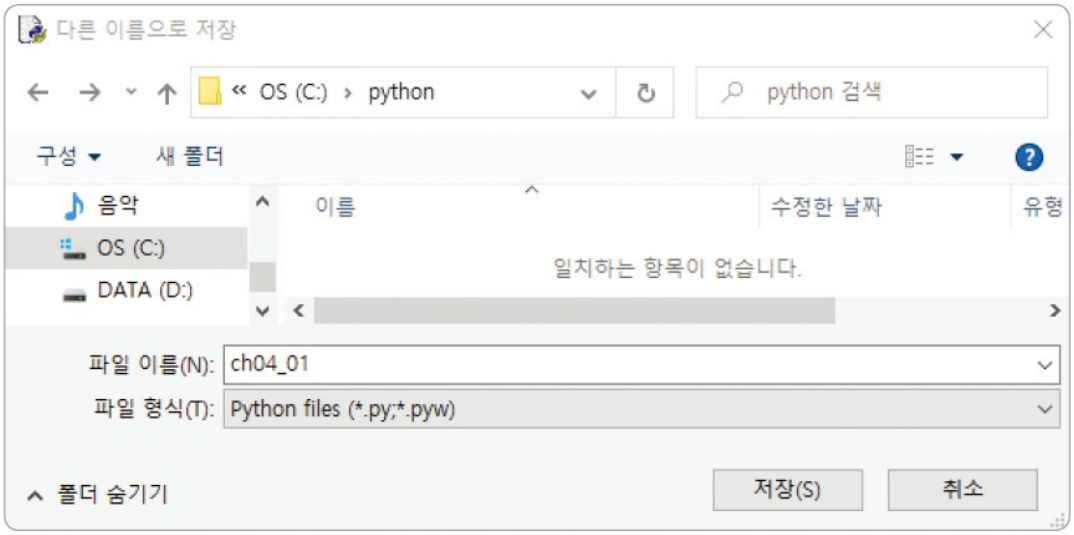


  - 2. 빈 화면이 뜨면 새로 만든 파일을 저장하기 위해서 [File]-[Save]를 클릭함



데이터 재활용

- ‘Hello Python!’을 5번 출력하는 프로그램 만들기
  - 3. 실습 파일을 저장할 폴더를 지정하고 파일 이름에 ‘ch04_01’을 입력한 후 [저장]을 클릭함



데이터 재활용

- ‘Hello Python!’을 5번 출력하는 프로그램 만들기
  - 4. 변수를 사용하지 않고 print() 함수를 5번 써서 코딩하기
    - [코드 4-1]은 실행하는 데 문제가 없음
    - 하지만 ‘Hello Python!’ 문자열을 ‘Hello C/C++!’로 수정해야 한다면 코드 전체를 수정해야 하는 문제점이 있음

코드 4-1 ch04_01.py

```
01   print('Hello Python!')
02   print('Hello Python!')
03   print('Hello Python!')
04   print('Hello Python!')
05   print('Hello Python!')
```

```
Hello Python!
Hello Python!
Hello Python!
Hello Python!
Hello Python!
```

데이터 재활용

- ‘Hello Python!’을 5번 출력하는 프로그램 만들기
  - 5. **1**~**3** 과정에서 파일 이름을 ‘ch04_02’로 설정하고, 다음과 같이 코딩하면 [코드 4-1]의 결과와 같은 것을 볼 수 있음

코드 4-2 ch04_02.py

```
01   introStr = 'Hello Python!'
02   print(introStr)
03   print(introStr)
04   print(introStr)
05   print(introStr)
06   print(introStr)
```

```
Hello Python!
Hello Python!
Hello Python!
Hello Python!
Hello Python!
```

데이터 재활용

- ‘Hello Python!’을 5번 출력하는 프로그램 만들기
  - 6. [코드 4-2]에서 1행을 ‘Hello C/C++!’으로 수정한 후 실행하기
    - 간단하고 빠르게 프로그램을 수정할 수 있음

코드 4-3 ch04_03.py

```
01  introStr = 'Hello C/C++!'
02  print(introStr)
03  print(introStr)
04  print(introStr)
05  print(introStr)
06  print(introStr)
```

```
Hello C/C++!
Hello C/C++!
Hello C/C++!
Hello C/C++!
Hello C/C++!
```

확인 문제

확인문제

다음은 온도 설정 프로그램이다. 빈칸을 채우시오.

```
01   currentTemp = 25    # 현재 온도
02   targetTemp = 30     # 설정 온도
03   print('현재 온도')
04   print(              )
05   print('설정 온도')
06   print(              )
07   print('설정 온도와 현재 온도의 차이')
08   print(targetTemp -  )
```

정답

currentTemp, targetTemp, currentTemp

데이터 복사 & 변경

데이터 복사

- 할당 연산자(=)를 이용하면 변수에 저장된 데이터를 다른 변수에 쉽게 복사할 수 있음

```
>>> var1 = 123
>>> var2 = var1
>>> print(var1)
123
>>> print(var2)
123
```

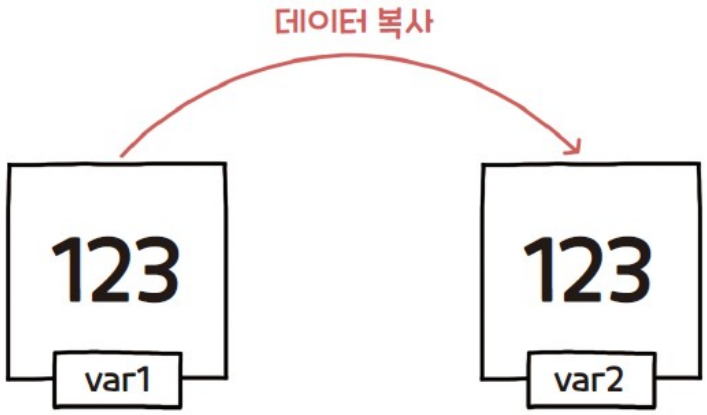


그림 4-14 var1 방에 있는 123을 var2 방으로 복사

데이터 복사 & 변경

- 데이터 변경
  - var1과 var2는 전혀 다른 방이며 서로 영향을 끼치지 않음
  - var2에 var1의 데이터를 복사한 후 var1의 데이터를 변경하면 var1은 변경되지만 var2는 변경되지 않는 것을 확인할 수 있음

```
>>> var1 = 321
>>> print(var1)
321
>>> print(var2)
123
```

데이터 복사 & 변경

- var1 변수에 문자, 문자열, 정수, 실수를 차례로 변경하고 출력하기

코드 4-4 ch04_04.py

```
01   var1 = 'H'                 # 문자
02   print(var1)
03   var1 = 'Hello Python!'     # 문자열
04   print(var1)
05   var1 = 10                  # 정수
06   print(var1)
07   var1 = 3.14                # 실수
08   print(var1)
09   var1 = True                # 논리형
10   print(var1)
```

```
H
Hello Python!
10
3.14
True
```

Section 09
입력 받은 데이터 저장하기

데이터를 입력 받아 저장하기

input( ) 함수

- 사용자에게 데이터를 입력 받을 때 사용하는 함수
- input() 함수로 사용자에게 입력 받은 데이터를 userData 변수에 저장하기

```
>>> userData = input()
Hello Python!          사용자 입력 부분
>>> print(userData)
Hello Python!
```

- print() 함수를 이용하여 데이터 입력을 유도하기

코드 4-5 ch04_05.py

```
01  print('데이터를 입력하세요.')
02  userData = input()
03  print(userData)
```

```
데이터를 입력하세요.
Hello Python!          사용자 입력 부분
Hello Python!
```

데이터를 입력 받아 저장하기

- input( ) 함수
  - print() 함수 대신 input() 함수의 매개변수(parameter)를 이용하기

코드 4-6 ch04_06.py

```
01   userData = input('데이터를 입력하세요.')
02   print(userData)
```

```
데이터를 입력하세요. Hello Python!
Hello Python!
```

데이터를 입력 받아 저장하기

- input( ) 함수의 주의 사항
  - input() 함수로 입력 받은 데이터는 무조건 문자열로 처리되며 사용자에게 정수나 실수로 유도해도 문자열로 처리됨

```
>>> userData = input('정수를 입력하세요.')
정수를 입력하세요. 10
>>> print(userData)
10
>>> type(userData)
<class 'str'>
```

문자(열)형

  - 사용자에게 입력 받은 데이터를 정수형, 실수형, 논리형으로 다루기 원한다면 자료형을 변환해야 함

```
>>> userData = input('정수를 입력하세요.')
정수를 입력하세요. 10
>>> userData = int(userData)
>>> type(userData)
<class 'int'>
```

자료형 변환을 통해서 정수형이 됐다.

확인 문제

확인문제

1. 다음은 사용자가 입력한 데이터를 화면에 출력하는 프로그램이다. 실행 결과를 보고 빈 칸을 채우시오.

```
01  userName = ''
02  userAge = ''
03  userName = input(                    )
04  print('사용자 이름')
05  print(              )
06  userAge = input(                    )
07  print('사용자 나이')
08  print(              )
```

```
이름을 입력하세요. 홍길동
사용자 이름
홍길동
나이를 입력하세요. 25
사용자 나이
25
```

2. 다음은 코드의 실행 결과 userData의 자료형으로 옳은 것은 무엇인가?

```
>>> userData = input('원하는 숫자를 입력하세요.')
원하는 숫자를 입력하세요. 20
```

① int　　② str
③ float　　④ bool

정답

1. '이름을 입력하세요.', userName, '나이를 입력하세요.', userAge　　2. ②

Section 10
터틀 프로그래밍

데이터를 입력 받아 저장하기

변수를 활용하여 정삼각형 그리기

- 정삼각형의 내각은 항상 120°로 일정함

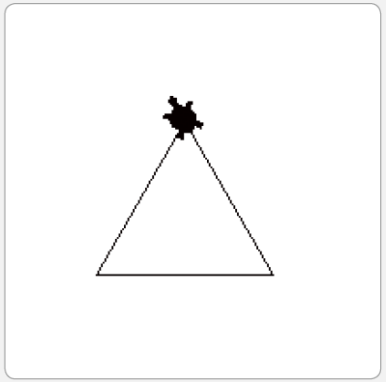
```
>>> import turtle
>>> t = turtle.Turtle()
>>> t.shape('turtle')
>>> angle = 120
>>> t.right(angle)
>>> t.forward(100)
>>> t.left(angle)
>>> t.forward(100)
>>> t.left(angle)
>>> t.forward(100)
```

120을 angle 변수에 저장한다.

오른쪽으로 angle만큼 회전한다.

왼쪽으로 angle만큼 회전한다.

왼쪽으로 angle만큼 회전한다.



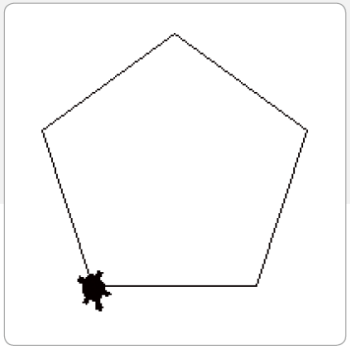
데이터를 입력 받아 저장하기

- 사용자에게 회전 각도와 전진 길이를 입력 받아 정오각형 그리기
  - 회전하는 각도의 변수 : angle / 전진하는 길이의 변수 : length
    - 정오각형의 내각은 72°로 일정함

```
>>> import turtle
>>> t = turtle.Turtle()
>>> t.shape('turtle')
>>> angle = int(input('회전 각도를 입력하세요. '))
회전 각도를 입력하세요. 72
>>> length = int(input('전진 길이를 입력하세요. '))
전진 길이를 입력하세요. 100
>>> t.forward(length)
>>> t.left(angle)
>>> t.forward(length)
>>> t.left(angle)
>>> t.forward(length)
>>> t.left(angle)
>>> t.forward(length)
>>> t.left(angle)
>>> t.forward(length)
```

length만큼 전진

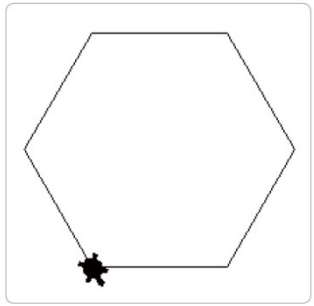
왼쪽으로 angle만큼 회전



확인 문제

확인문제

터틀을 이용해서 사용자가 회전 각도와 전진 길이를 입력하면 정육각형이 그려지는 프로그램을 만드시오.



정답

```
>>> import turtle
>>> t = turtle.Turtle()
>>> t.shape('turtle')
>>> angle = int(input('회전 각도를 입력하세요. '))
회전 각도를 입력하세요. 60
>>> length = int(input('전진 길이를 입력하세요. '))
전진 길이를 입력하세요. 100
>>> t.forward(length)
>>> t.left(angle)
>>> t.forward(length)
>>> t.left(angle)
>>> t.forward(length)
>>> t.left(angle)
>>> t.forward(length)
>>> t.left(angle)
>>> t.forward(length)
>>> t.left(angle)
>>> t.forward(length)
```

100px 전진

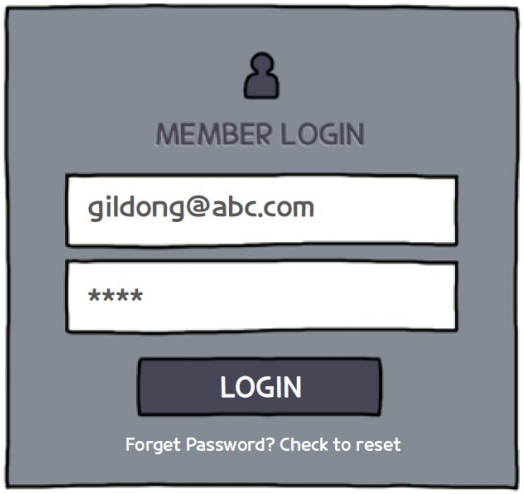
왼쪽으로 60도 회전

실전 예제

실전 예제1 – 비밀번호 메일 발송 프로그램

문제

다음은 비밀번호를 잊은 고객에게 비밀번호를 알려주는 메일 본문입니다. 아래 포맷에 따라 비밀번호를 비롯한 회원 개인정보를 발송하는 메일 프로그램을 만들어봅시다.




```
To. gildong@abc.com
▶ 아이디 및 비밀번호 확인
  홍길동 고객님 안녕하세요.
  홍길동 고객님의 아이디와 비밀번호는 아래와 같습니다.
  아이디 : gildong
  비밀번호 : 1234
```

실전 예제1 – 비밀번호 메일 발송 프로그램

해결

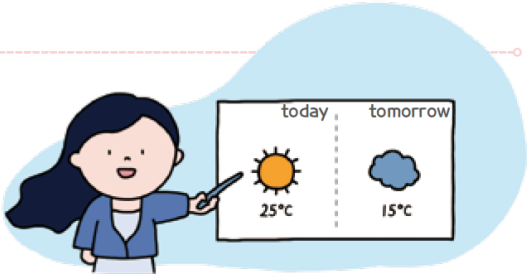
ch04_appEx_01.py

```
01  print('회원 정보를 입력하세요.')
02  userName = input('이름 :')
03  userMail = input('메일 :')
04  userId = input('아이디 :')
05  userPw = input('비밀번호 :')
06  print('-------------------------------------------------------')
07  print('To. ' + userMail)
08  print('▶ 아이디 및 비밀번호 확인')
09  print(userName + ' 고객님 안녕하세요.')
10  print(userName + ' 고객님의 아이디와 비밀번호는 아래와 같습니다.')
11  print('아이디 : ' + userId)
12  print('비밀번호 : ' + userPw)
13  print('-------------------------------------------------------')
```

실전 예제2 – 날씨 예보 프로그램

문제

날짜, 아침 기온, 낮 기온, 비올 확률, 미세먼지 수치, 일출 시간, 일몰 시간 등의 날씨 정보를 입력하면 실행 결과와 같이 내일 날씨 예보가 화면에 출력될 수 있도록 프로그램을 만들어봅시다.




```
내일 날씨 예보입니다.
월요일인 3월 30일의 아침 최저 기온은 -1도, 낮 최고 기온은 10도로 예보됐습니다.
비올 확률은 45%이고, 미세먼지는 좋음 수준일 것으로 예상됩니다.
일출 시간은 오전 6시 30분이고, 일몰 시간은 오후 7시 20분입니다.
바다의 물결은 남해 앞바다 0.5m, 동해 앞바다 1.5m, 서해 앞바다 0.5m 높이로 일겠습니다.
지금까지 3월 30일 월요일 날씨 예보였습니다.
```

실전 예제2 – 날씨 예보 프로그램

해결

ch04_appEx_02.py

```
01  print('내일 날씨 정보를 입력하세요.')
02  dayData = input('요일 : ')
03  dateData = input('날짜 : ')
04  lowestTemp = input('아침 최저기온 : ')
05  highestTemp = input('낮 최고기온 : ')
06  rainingPercent = input('비올 확률 : ')
07  fineDustStatus = input('미세먼지 : ')
08  sunriseTime = input('일출 시간 : ')
09  sunsetTime = input('일몰 시간 : ')
10  southOceanWave = input('남해 물결 높이 : ')
11  eastOceanWave = input('동해 물결 높이 : ')
12  westOceanWave = input('서해 물결 높이 : ')
13  print('내일 날씨 예보입니다.')
14  print(dayData + '요일인 ' + dateData + '의 ' + '아침 최저 기온은 ' + lowestTemp
      + '도, 낮 최고 기온은 ' + highestTemp + '도로 예보됐습니다.')
15  print('비올 확률은 ' +  rainingPercent + '%이고, 미세먼지는 ' + fineDustStatus
      + ' 수준일 것으로 예상됩니다.')
16  print('일출 시간은 ' + sunriseTime + '이고, 일몰 시간은 ' + sunsetTime + '입니다.')
17  print('바다의 물결은 남해 앞바다 ' + southOceanWave + 'm, 동해 앞바다 '
      + eastOceanWave + 'm, 서해 앞바다 ' + westOceanWave + 'm 높이로 일겠습니다.')
18  print('지금까지 ' + dateData + ' ' + dayData + '요일 날씨 예보였습니다.')
```

Thank you!

한빛아카데미 Hanbit Academy, Inc.