



CHAPTER 2. 변수와 단순한 데이터 타입



프로그램 작성 시 주의할 점

- 기호 사용
 - 대괄호([]), 중괄호({ }), 소괄호(())가 완전히 다르게 인식됨
 - 세미콜론(;)과 콜론(:), 점(.)의 개수를 잘못 쓰면 문제가 됨
- 들여쓰기로 블록 구조 표현함
 - 파이썬은 함수 정의, if 문 또는 반복문과 같은 구문의 범위를 들여쓰기로 표현 (다른 언어는 주로 { }로 표현)
 - 들여쓰기를 잘못 넣으면 에러가 발생하므로 어디까지가 하나의 블록인지 눈으로 확인하기 쉬운 것이 장점.

프로그래밍 사전 지식 (1/ 5)

함수, 메서드, 모듈

각각 프로그램 단위를 나타내는 말로 파이썬 표준 라이브러리는 함수, 메서드, 모듈의 형태로 제공됨

- 함수란?
 - 함수는 어떤 단위로 묶어서 나눌 수 있는 처리를 하나의 블록으로 만든 것
 - 함수에는 ‘인수’와 ‘반환 값’이 있는데 이는 수학의 함수 구조와 유사
 - 함수는 다른 프로그래밍 언어에서도 사용하는 프로그램의 기본 단위
 - 내장 함수란 파이썬에서 라이브러리 등을 따로 불러오지 않아도 언제나 함수 형태로 이용할 수 있는 것
 - 내장 함수는 함수에 따라서 인수와 반환 값이 있기도 하고 없기도 함.

프로그래밍 사전 지식 (2/5)

- 메서드란?
 - 메서드란 객체 지향 프로그래밍에서 객체를 제어 또는 동작시키는 것을 의미
 - 객체 지향 프로그래밍은 프로그램이나 데이터를 단계(절차)별로 관리하는 것이 아니라 프로그램에서 다루는 대상 별로 관리하는 방법
 - 예를 들어 게임에서는 플레이어나 규칙, 급여 관리 시스템에서는 종업원이나 급여, 라즈베리 파이에서는 컴퓨터에 연결한 스위치나 LED 등이 객체가 됨
 - 보통 객체는 객체에 관련된 정보(데이터 또는 프로퍼티라고도 함)와 객체를 조작하는 메서드로 구성됨
-

프로그래밍 사전 지식 (3/5)

- 파이썬에서는 프로그램에서 다루는 모든 것을 ‘객체’로 보며, 자주 사용하는 객체와 메서드는 각각 내장 객체와 내장 메서드로 구현되어 있음
- 예를 들어 문자열 객체에 어떤 조작을 하려면 미리 준비된 문자열 객체의 내장 메서드를 이용
- 프로그램에서 내장 메서드를 사용하려면 메서드를 사용할 객체와 메서드명 사이에 점(.)을 이용하여 표현
- 문자열에서 지정한 문자의 위치를 찾는 index() 메서드 사용 예

```
>>> letters = 'Python'
>>> letters.index('y')
1
```

프로그래밍 사전 지식 (4/5)

- 모듈이란?
- 모듈은 프로그래밍이나 공학에서 널리 사용하는 단어로 시스템을 구성하는 하나의 독립적인 요소로, 프로그래밍 언어에서 모듈이란 함수나 메서드보다도 더욱 큰 단위를 의미
- 파이썬 표준 라이브러리는 내장 함수 같은 미리 준비된 기능 외에도 더욱 복잡한 처리나 전문적인 내용에 특화된 다양한 객체나 함수, 상수(미리 정해진 숫자)를 정의해 둔 파일이 있는데 이런 파일을 ‘모듈’이라 함
- 예를 들어 math 모듈에는 수학에서 사용하는 함수나 상수가 포함되어 있음
- 파이썬에서 모듈 기능을 사용하려면 프로그램 시작 부분에서 사용하려는 모듈을 임포트(import)해야 함
- 자신이 만든 프로그램을 모듈로 만들어서 다른 프로그램에서 사용할 수도 있는데 이때도 프로그램 시작 부분에서 자신이 만든 모듈을 임포트해야 함

프로그래밍 사전 지식 (5/5)

〈대표적인 모듈과 관련 기능〉

모듈명	기능	소개하는 자료형, 메서드, 함수와 작성법
math	수학 관련 모듈 〈import 문〉 import math	math.sqrt(a): a의 제곱근을 돌려줌 math.sin(a): a의 사인 값을 돌려줌 math.radians(a): a의 라디안을 돌려줌 math.pi: 원주율 3.141592...(상수)
sys	운영체제, 시스템 관련 모듈 〈import 문〉 import sys	sys.maxint(): 파이썬의 정수형이 지원하는 정수 최댓값(상수)
random	난수 관련 모듈 난수란 예측할 수 없는 임의의 값으로, 게임이나 뽑기, 암호 생성 등에 사용함 〈import 문〉 import random	random.randint(a, b): a와 b 사이에 있는 임의의 값을 난수를 돌려줌
time	시각 관련 모듈 〈import 문〉 import time	time.asctime(): 인수를 지정하지 않으면 현재 시각을 돌려줌 time.sleep(): 인수로 지정한 초만큼 처리를 정지함 time.time(): 에포크(epoch, 신기원) 시간에서 경과한 시간을 초로 돌려줌. 인수는 필요 없지만 괄호(())는 생략할 수 없음. 에포크 시간은 time.gmtime(0)으로 얻을 수 있음

변수를 사용한 프로그램 (1/2)

- 변수란
 - 숫자나 문자열 등의 데이터에 라벨(이름)을 붙여 컴퓨터가 기억하도록 하는 기능
 - 기억하기 어려운 데이터를 기억하기 쉬운 이름으로 저장하여 원할 때 사용할 수 있음
 - 데이터에 이름을 붙임으로써 의미 부여 가능
 - 변수의 이름은 변수가 갖는 의미를 나타내는 영어 단어를 사용하는 것이 유용함

변수를 사용한 프로그램 (2/2)

- 변수에 값을 대입하기
 - 변수와 값 사이에 '=' 를 이용
 - 확인하기: 변수명을 입력한 후 Enter 또는 `print(변수명)`

변수에 사용할 수 있는 문자

- 변수 이름으로는 문자와 숫자 및 밑줄(`_`) 사용 가능 단, 첫 번째 문자에 숫자는 불가능
- 변수 이름에는 공백을 사용할 수 없음
- 예약어(파이썬에 이미 정의된 문자열) 또는 함수명은 사용할 수 없음.

```
>>> import keyword
>>> keyword.kwlist
['False', 'None', 'True', 'and', 'as', 'assert', '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']
>>>
```

파이썬의 내장 자료형

- 파이썬에 미리 준비된 자료형
- 파이썬은 내장 자료형 데이터도 객체 지향에서 객체로 다룸
- 내장 자료형 데이터는 자료형마다 준비된 메서드나 내장 함수 등을 써서 다양하게 조작할 수 있음

내장 자료형	설명	작성법	불변/가변
정수	소수가 없는 숫자	<code>a = 10</code> , <code>a = -10</code>	불변
부동소수	소수가 있는 숫자	<code>a = 10.5</code>	불변
불	True 또는 False 값을 가진 자료형	<code>a = True</code>	불변
문자열	문자 데이터를 순서대로 나열한 데이터. 문자가 하나뿐이더라도 문자열이라고 부름	<code>a = 'Python'</code> 문자열의 0번째(첫 글자) 문자는 <code>a[0]</code> 으로 지정	불변
리스트	숫자나 문자열 등의 요소를 순서대로 나열한 구조체	<code>a = ['p','y','t','h','o','n']</code> 리스트 <code>a</code> 의 0번째(첫 글자) 요소는 <code>a[0]</code> 으로 지정	가변
튜플	리스트와 비슷하지만 불변성 때문에 바꿔 쓸 수 없음	<code>a = ('p','y','t','h','o','n')</code> 리스트 <code>a</code> 의 0번째(첫 글자) 요소는 <code>a[0]</code> 으로 지정	불변
딕셔너리	각 데이터에 이름(키)이 있는 데이터 구조	<code>a = {'happy':('^~^'), 'sad':('ToT')}</code> 딕셔너리 <code>a</code> 의 키 <code>happy</code> 에 대응하는 값은 <code>a['happy']</code> 로 지정	가변

문자열 (1/4)

- 문자들을 다루는 데이터형
 - 문자들을 작은따옴표(') 혹은 큰따옴표(")로 둘러싸서 문자열형을 지정함
 - 작은따옴표를 세 개 혹은 큰따옴표를 세 개 연속하여 붙여 복수행의 문자열을 만들 수 있음

```
>>> ''' hello  
bye  
'''  
' hello\nbye\n'  
>>>
```

내용 중 \n은 개행을 뜻하는 기호(개행 코드)

문자열 (2/4)

- 문자열형의 편리한 기능 – 메소드 이용

- `upper()`: 문자열형 데이터의 문자들을 전부 대문자로 바꿔주는 기능
- `lower()`
- `title()`
- `count()`:
 - 지정한 문자가 몇 개 있는지 세는 기능
 - () 안에 세고 싶은 문자를 넣음

※ 참고) 각 데이터형 별로 메소드가 존재함. 메소드를 사용하기 위해서는 변수와 데이터형의 메소드를 점(.)으로 연결

문자열 (3/4)

- 문자열 조작

- +를 사용한 문자열 조작

문자열형 간에만 연결 가능 : 문자열형과 수치형을 +로 연결하면 타입 오류 (TypeError)가 발생됨

- *을 사용한 문자열 조작

- 문자열형 데이터에 숫자를 곱하면 문자열이 반복됨

- 문자열형과 수치형 간에만 사용 가능: 문자열형 간에 곱셈(*)하면 타입 오류 (TypeError)가 발생됨

- '\t' (탭), '\n' (개행) 문자

문자열 (4/4)

- 공백 잘라내기 메서드

`lstrip()`

`rstrip()`

`strip()`

숫자

- 숫자를 다루는 데이터형
 - 정수(integer, int)
소수점을 갖지 않는 숫자
정수를 대입한 변수도 정수 타입의 수치형으로 다뤄짐
 - 부동소수점(소수)(float)
소수점이 들어간 숫자
 - 복소수(complex)
허수부를 j 혹은 J로 표현함
허수부의 계수가 1일 때도 1이라는 숫자를 생략하지 않음
같은 수치형인 정수나 소수와 계산 가능
 - 숫자를 문자열로 다루는 경우도 있으며, 이때는 산술적 의미에서 연산할 수 없음

주석

■ 주석

- 프로그램을 작성하다 숫자의 의미나 처리 내용 등을 설명하고 싶을 때 작성
- # 부터 해당 줄 끝까지를 주석으로 처리
- 주석을 한글로 작성하려면 문자 코드를 지정해야 함 파이썬 2.x 버전은 표준 문자 코드가 아스키(ASCII)이므로 그대로는 사용할 수 없음
- 라즈비안 표준 문자 코드를 Unicode로, 인코딩 방식은 UTF-8로 설정했으니 그에 맞게 파이썬 프로그램도 UTF-8로 지정(`# coding: utf-8`)



연산자

산술연산자

산술연산자	사용법	뜻
+	1 + 1	덧셈
-	2 - 2	뺄셈
*	3 * 3	곱셈
/	4 / 4	나눗셈(몫)
%	5 % 3	나눗셈(나머지)
**	6 ** 2	거듭제곱

비교연산자

비교연산자	예	뜻
>	$x > y$	x는 y보다 크다
>=	$x \geq y$	x는 y와 같거나 크다
<	$x < y$	x는 y보다 작다
<=	$x \leq y$	x는 y와 같거나 작다
==	$x == y$	x와 y는 같다
!=	$x != y$	x와 y는 같지 않다

논리연산

- ‘참’ 과 ‘거짓’ 조건을 논리곱(AND), 논리 합(OR), 부정(NOT)으로 연산하는 것
- 파이썬에서 참은 True로, 거짓은 False로 표현
- True 또는 False 값을 갖는 자료형을 불(Boolean)이라고 함
- 파이썬에서 논리곱은 and, 논리합은 or, 부정은 not
- 논리 연산의 동작 예

```
>>> a = True  
>>> b = False  
>>> a and b
```

```
>>> a = True  
>>> b = False  
>>> a and not b
```

```
>>> a = True  
>>> b = False  
>>> a or b
```

논리연산

- 프로그램에서 다루는 논리 연산 - 조건에 따라 처리 변경
 - 프로그램에서 if 문 등과 조합해서 주어진 조건에 따라 처리할지 판정할 때 논리 연산을 함
 - 두 조건이 있을 때, 논리곱(AND)은 두 조건을 모두 만족해야(두 조건이 모두 True 일 때) 처리
 - 논리합(OR)은 조건을 하나라도 만족하면 처리
 - 조건 결과를 반전시키고 싶으면 조건식 앞에 부정(NOT)을 붙임