

# Learn to Program

Young Hand

2017년 8월 7일

## 차 례

<b>1</b>	<b>프로그래밍</b>	<b>1</b>
1.1	데이터와 변환 . . . . .	1
1.2	변수 . . . . .	1
1.3	제어문 . . . . .	1
1.4	함수 . . . . .	1
1.5	구조화 . . . . .	2
1.6	연습문제 . . . . .	2

# 1 프로그래밍

## 1.1 데이터와 변환

데이터와 이를 변환하는 논리가 프로그래밍이다. 데이터의 종류를 타입이라고 한다. 데이터 타입이 정해지면 이를 변환하는 논리도 정해진다.

예를 들어, 정수가 있다면 사칙연산이 가능하다. 예를 들어, 문자열이 있다면 문자열의 길이를 찾거나 합치거나 역으로 바꾸거나 부분 문자열을 찾거나 할 수 있다. 예를 들어, 실수가 있다면 미적분까지 가능하다.

종류(타입)이 있는 데이터에 대해 변환(함수, 연산)을 논리(if)에 따라 적용하는 것이 프로그램이다. 또는 데이터 종류(타입)에 대해 변환을 만드는 것이 프로그램이다.

## 1.2 변수

프로그래밍이 복잡해지는 이유 중 하나는 이 데이터를 어딘가에 보관해야 하기 때문이다. 당연히 물리적인 메모리인 메인 메모리나 디스크에 저장되는데 프로그램 코드 안에도 메모리를 신경써야 하기 때문에 복잡해진다.

프로그램 안에서 데이터를 보관하는 장소를 변수라고 한다. 변수는 임시 메모리에 저장될 수도 있고 메인 메모리에 저장될 수도 있다. 어디에 저장될 지를 결정하는 몇 가지 규칙이 있고 이를 익히면 된다.

## 1.3 제어문

데이터에 대한 변환 기능을 수행할 때 연산자를 사용하게 되지만 논리에 따라 흐름을 구성하기 위해서는 조건에 따라 변경을 선택해야 한다. 이를 위해 필요한 구문들을 제어문이라고 한다.

## 1.4 함수

데이터(들)를 입력으로 받아서 변환된 데이터(들)를 출력으로 제공하는 어떤 것이 함수이다. 예전에 수학 시간에 배운 함수와 비슷하지만 보다 일반적이며 훨씬

복잡한 내용을 포함할 수 있다. (그래서 버그도 많이 생긴다)

## 1.5 구조화

데이터, 변수, 제어문, 함수로만 프로그램을 만들면 엄청나게 많은 함수를 갖게 된다. 그러면 뭐가 뭔지 모르는 상태가 되기 때문에 다양한 정리 방법이 고안 되었다. 그 중에 많이 쓰이는 게 객체지향프로그래밍 (Object Oriented Programming, 줄여서 OOP)이 있는데 클래스 단위로 함수와 변수들을 모아서 관리한다.

일단 그렇게 모으는 도구로 시작했는데 쓰다 보니 이것 저것 유용할 것 같은 기능들이 생각나서 추가하게 되었고 언어는 복잡해졌는데 원래 기능인 모으는 기능보다 나은 것은 아직까지 잘 안 보인다.

## 1.6 연습문제

**Ex 1.1** (계산기를 언어로 만들기). *calc*로 실행하는 윈도우 계산기에는 값을 지정하는 기능이 없다. 값 지정 기능이 있는 계산기 용 언어를 같이 만들어 보자.