

1장

왜 프로그래밍을 꼭 배워야 할까?

1장 왜 프로그래밍을 꼭 배워야 할까?

1.1 프로그래밍이 중요한 이유는?

1.2 알고 있어야 하는 내용과 이 책에서 배울 내용

1.3 프로그래밍 학습 계획

1.4 요약

1.1 프로그래밍이 중요한 이유는?



1.1.1. 프로그래밍은 전문가용이 아니다

- » 프로그래밍은 읽기, 쓰기, 수학, 요리와 다를 바 없는 기술이다
- » 프로그래밍을 조금만 알아두면 더 효율적으로 처리하고 싶은 일이 있을 때 남의 손을 빌리지 않고 원하는 일을 해낼 수 있다



1.1.2. 삶을 개선하라

- » 프로그램을 만들면 데이터를 읽어서 합계를 계산하거나 정해진 날짜 범위에 따라 체크를 분류하는 등 여러 가지 일을 할 수 있다
- » 짧은 프로그램을 작성해 모든 파일 이름을 자동으로 바꿀 수 있다
- » 방정식에서 빠진 매개변수를 찾아 계산해주는 프로그램을 작성하면 손으로 방정식을 푼 다음, 결과가 맞는지 프로그램을 실행해 검증할 수 있다
- » 학생 이름, 이메일 주소, 성적을 파일에서 읽어 미리 정해진 메시지 틀에 채워 넣은 뒤 자동으로 이메일을 전송하는 프로그램을 짤 수 있다



1.1.3. 직접 도전하라

- » 언뜻 보기에 프로그래밍은 기술적인 것처럼 느껴진다. 실제로 처음은 그렇다
- » 프로그래밍을 통해 한 가지 일을 처리하는 여러 가지 방법을 배우고 나면 어떤 방법을 사용하는 게 가장 좋은지 결정해야 한다
- » 배운 지식을 바탕으로 이런 결정을 내림으로써 목표를 설정하고 그 목표를 좀 더 효율적으로 달성하는 방법을 결정하는 데 비판적으로 사고하는 방법을 배울 수 있다



1.2.3. 직접 도전하라

- » 프로그래밍을 통해 한 가지 일을 처리하는 여러 가지 방법을 배우고 나면 어떤 방법을 사용하는 게 가장 좋은지 결정해야 한다
- » 배운 지식을 바탕으로 이런 결정을 내림으로써 목표를 설정하고 그 목표를 좀 더 효율적으로 달성하는 방법을 결정하는 데 비판적으로 사고하는 방법을 배울 수 있다

1.2 알고 있어야 하는 내용과 이 책에서 배울 내용



알고 있어야 하는 내용

» 변수

- 초등학교나 중학교에서 방정식을 배웠다면 변수(variable)도 알 것이다.

» 명제의 참/거짓

- 명제(statement)란 참/거짓을 정할 수 있는 문장이다.
- 부정어를 사용해서 문장의 참/거짓을 뒤집을 수도 있다.

» 명제 연결하기

- 둘 이상의 명제를 연결할 때는 그리고(and)나 또는(or)을 사용한다.
- '그리고/~고'는 두 명제가 모두 참일 때만 참이고, '또는/~거나'는 두 명제가 모두 거짓일 때만거짓이다.

» 결정하기

- 여러 명제가 있을 때 '만약 ~라면'을 사용해 어떤 명제의 참/거짓에 따라 결정을 내릴 수 있다.

» 흐름도의 흐름 따라가기

- 흐름도(flowchart)를 몰라도 이 책을 이해할 수 있지만, 흐름도를 이해하는 데
- 필요한 기술은 기본적인 프로그래밍을 이해하는 데 필요한 기술과 같다.

알고 있어야 하는 내용

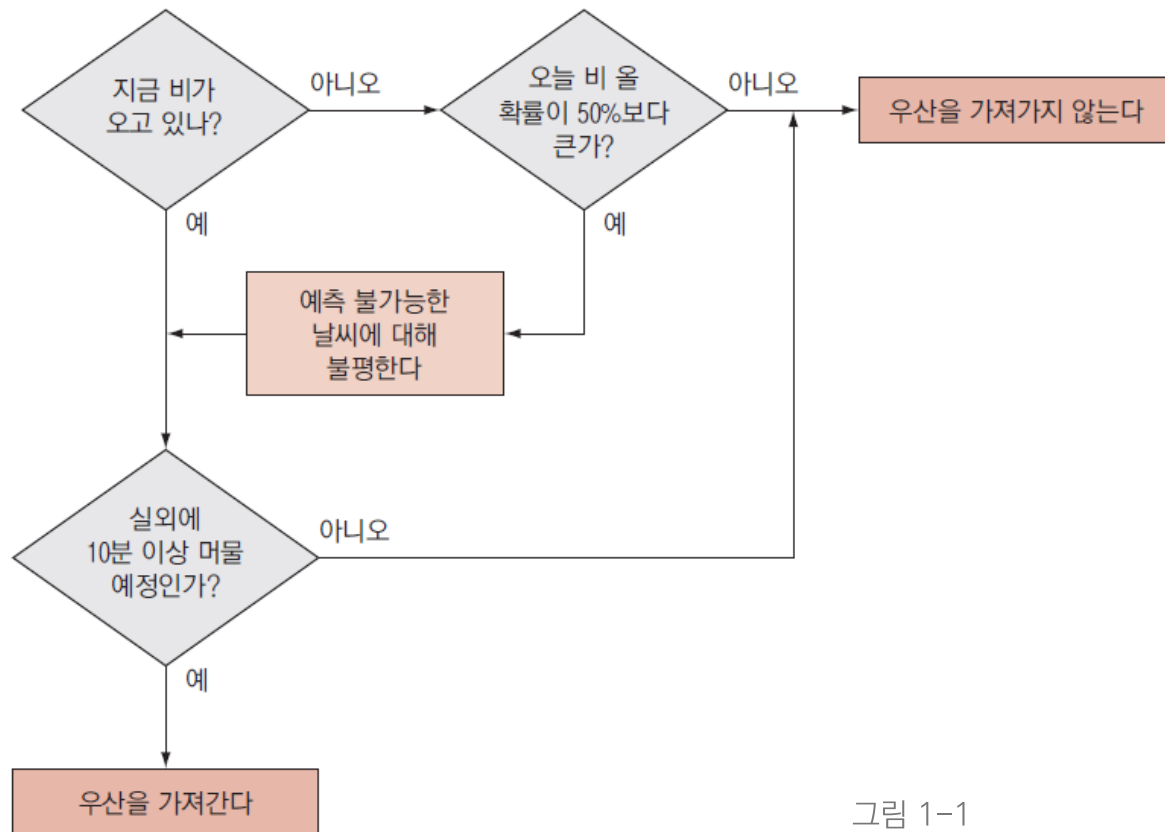


그림 1-1
우산을 가져갈지 여부를 결정하는 흐름도



프로그래밍 언어와 관계 없이 적용할 수 있는 기본 개념

- » 변수, 식, 문장 사용하기
- » 조건에 따라 결정하는 프로그램 만들기
- » 특정 조건이 성립하는 동안 작업을 반복 수행하는 프로그램 만들기
- » 언어가 제공하는 연산을 활용해 더 효율적인 프로그램 짜기
- » 코드를 더 읽기 좋게 만드는 방법과 큰 작업을 작은 덩어리로 나눠 프로그램을 유지보수(maintenance) 하기 쉽게 만들기



파이썬 프로그래밍 언어의 세부 사항

- » 언어 구문을 사용하는 방법
- » 서로 잘 맞물려 작동하는 코드 덩어리를 모아 더 복잡한 프로그램을 작성하는 방법
- » 다른 사람이 작성한 코드를 여러분이 작성한 프로그램에 사용하는 방법
- » 작성한 프로그램이 제대로 작동하는지 효과적으로 검사하기 위한 디버깅(debugging)이나 테스트(testing) 방법
- » 키보드나 마우스 같은 장치와 상호작용하는 프로그램을 작성하는 방법
- » 데이터 위주의 프로그램이나 수학적 프로그램을 작성하는 방법

1.3 프로그래밍 학습 계획



1.3.1. 첫걸음을 떼자

- » 이 책은 한 번도 프로그램을 짜본 적이 없는 사람을 위한 책이다
- » 앞부분의 몇 장은 연습 문제를 직접 풀어 보는 게 중요하다
- » 파이썬을 사용하여 직접 문제를 풀면서 프로그래밍 기본 개념을 배울 수 있다
- » 다른 프로그래머가 작성한 패키지(package)를 살펴보고 패키지를 통해 더 복잡한 프로그램을 만드는 방법을 배운다
- » 일정한 구조를 지닌 파일을 읽는 방법, 파일을 읽어서 수집한 데이터를 분석하는 방법, 분석한 결과 데이터를 다른 파일에 기록하는 방법을 살펴본다



1.3.2. 연습하고, 연습하고, 연습하고, 연습하라

- » 프로그램을 작성했을 때 오류가 발생해도 당황하지 말자. 예기치 못한 실수를 수정하는 과정에서 프로그래밍과 파이썬 언어를 더 잘 이해할 수 있다.
- » 연습 문제는 각 장에서 제시한 중요한 개념을 다루므로 이 책이 다루는 내용을 이해하려면 모든 연습 문제를 시도해 봐야 한다.
- » 좀 더 도전하고 싶다면 모든 연습 문제를 다른 방식으로 풀어 보고, 일상적으로 부딪치는 문제를 해결하는 새로운 프로그램을 작성해 보자!



1.3.3. 프로그래머처럼 생각하라

- » 프로그래머는 이미 존재하는 알고리즘이나 꼭 만족해야 하는 요구 조건을 바탕으로 프로그램을 구현한다.
- » 프로그래머는 프로그래밍 언어의 세부 사항을 잘 알고 있고, 효율적이면서 올바른 코드를 빨리 구현해야 한다.
- » 실제로는 프로그래머와 컴퓨터 과학자의 역할이 겹치는 경우가 많기 때문에 경계가 항상 명확하지는 않다.
- » 이 책은 컴퓨터에게 자세한 지침을 제시하여 컴퓨터가 여러분이 원하는 목표를 달성하게 만드는 방법을 보여 주고, 여러분이 그런 방법에 익숙해지게 도울 것이다.



1.3.4. 요약

- » 기본적인 프로그래밍 개념만 사용해도 좀 더 편하게 일상생활을 할 수 있다.
- » 때로 아주 간단한 프로그램을 작성하는 것만으로도 삶이 더 편해진다.
- » 프로그래밍은 기술이기 때문에 많이 연습할수록 더 잘 할 수 있다.
- » 이 책을 읽으면서 여러분이 직접 하고 있는 지루한 작업을 찾아낸 뒤, 프로그래밍을 통해 좀 더 효율적으로 해결할 수 있는지 생각해 보고, 실제로 개선하려고 시도해 보자.