



고등학교 소프트웨어 교육

# ☆ 소프트웨어와 친해지기



이영준 · 최정원 · 안상진 · 전준호 지음



미래창조과학부



한국과학창의재단



## 머리말

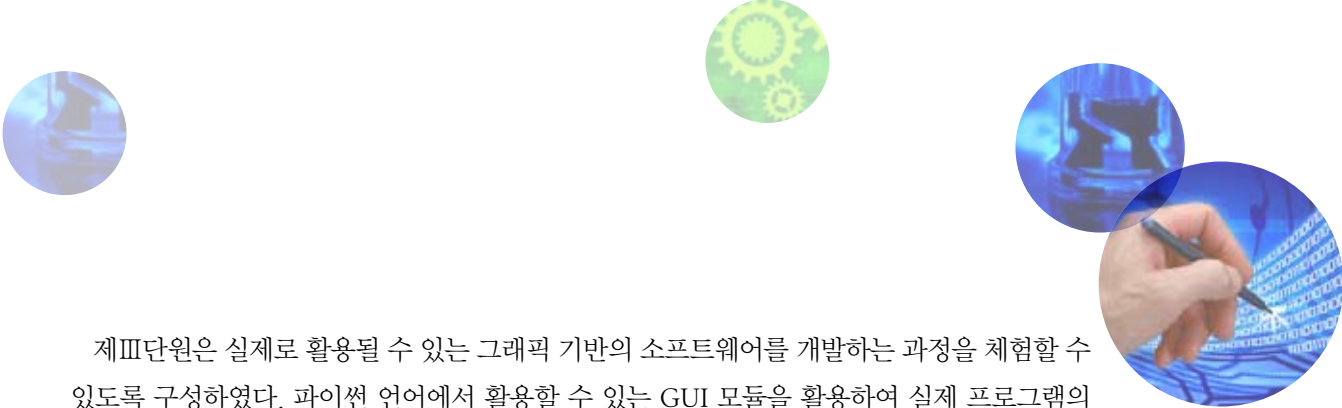
오늘날 컴퓨팅 기술의 발달은 사회를 급속도로 성장·발전시키는 원동력이 되었고, 인간의 사고만으로 해결하기 어려웠던 복잡한 문제들을 쉽게 해결할 수 있도록 도와주었을 뿐 아니라, 방대한 양의 자료를 수집, 가공하여 보다 유의미한 정보를 산출할 수 있도록 하고 있다. 따라서 미래의 지식 정보 사회를 발전시키기 위한 인재는 과학, 인문, 예술 등 사회 모든 분야의 발전을 위해 컴퓨팅 사고력을 반드시 갖추어야 한다. 학생들은 컴퓨팅의 능력을 깨닫고 그 영향력을 이해하기 위해서 교육과정에서 다양한 컴퓨팅 기기를 접하고 그 활용 방법을 이해하고 문제 해결 과정에 적용해 보는 경험이 필요하다.

본 교재는 고등학교 학생들이 컴퓨팅 기기와 소통할 수 있는 방법인 프로그래밍 언어를 활용하여 효과적이고 효율적인 문제 해결을 할 수 있는 경험을 제공하도록 구성하였다. 또한 실제 사용 가능한 소프트웨어를 설계하고 제작하는 경험을 통해 학생들의 창의적인 문제 해결력을 기를 수 있도록 하였다.

제Ⅰ단원은 고급 언어 중 하나인 파이썬 프로그래밍 언어를 알아보고, 터틀 모듈을 활용하여 파이썬 프로그래밍 언어를 익힐 수 있도록 구성하였다. 터틀 모듈로 그림을 그리는 과정을 학습하며 프로그래밍의 기본적인 개념인 순차, 반복, 조건, 변수, 함수 등의 개념을 학습하도록 하였다.

제Ⅱ단원은 파이썬 언어를 이용한 다양한 수준의 문제 해결을 경험하도록 하여 학생들의 컴퓨팅 사고력을 증진시킬 수 있도록 하였다. 앞에서 학습한 프로그래밍 개념을 활용하여 문제를 해결하기 위한 알고리즘을 설계하고, 실제 프로그래밍 언어로 구현해 보면서 학생들이 컴퓨팅 사고력을 자연스럽게 함양할 수 있도록 구성하였다.





제Ⅲ단원은 실제로 활용될 수 있는 그래픽 기반의 소프트웨어를 개발하는 과정을 체험할 수 있도록 구성하였다. 파이썬 언어에서 활용할 수 있는 GUI 모듈을 활용하여 실제 프로그램의 인터페이스를 구성하고 내부적인 기능을 제작하는 과정을 통해 우리가 사용하는 소프트웨어가 어떠한 방식으로 제작되는지 학생들이 경험하고 이해할 수 있도록 하였다.

제Ⅳ단원은 앱 인벤터를 활용하여 실제 모바일 기기에서 동작하는 어플리케이션을 제작하는 경험을 가질 수 있도록 구성하였다. 모바일 기기에 포함된 다양한 입·출력 장치와 센서를 활용하는 소프트웨어를 제작하는 경험을 통해 학생들이 컴퓨터뿐 아니라 다양한 컴퓨팅 기기에 소프트웨어가 제작되고 사용된다는 것을 이해할 수 있도록 하였다.

본 교재를 통해 학생들이 컴퓨팅 기기가 제공하는 무한한 가능성에 대한 인식을 가지고 미래 사회의 문제 해결을 위해 컴퓨팅 시스템의 능력을 창의적으로 활용할 수 있는 역량을 갖추길 바란다.







## 소프트웨어와 프로그래밍

- |                  |    |
|------------------|----|
| 01. 문제 해결과 컴퓨팅   | 8  |
| 02. 터틀과 함께하는 파이썬 | 19 |



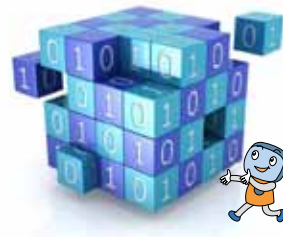
## 파이썬과 문제 해결

- |               |    |
|---------------|----|
| 01. 알고리즘      | 38 |
| 02. 간단한 문제 해결 | 42 |
| 03. 알고리즘 최적화  | 60 |





### GUI 프로그래밍



- |              |    |
|--------------|----|
| 01. 계산기 프로젝트 | 76 |
| 02. 자판기 프로젝트 | 86 |



### 앱 프로그래밍



- |              |     |
|--------------|-----|
| 01. 주소록 앱    | 100 |
| 02. 안전 알리미 앱 | 116 |
| 03. 게임 앱     | 127 |



# CHAPTER





# 소프트웨어와 프로그래밍

## 01 문제 해결과 컴퓨팅

## 02 터틀과 함께하는 파이썬

컴퓨팅 시스템을 사용하다 보면 내가 원하는 기능을 모두 가지고 있는 소프트웨어를 찾기 어려울 때가 있다. 내가 원하는 방식으로 문제를 해결하는 기능을 수행하는 소프트웨어를 제작하기 위해서는 컴퓨팅 시스템과 대화할 수 있는 프로그래밍 언어를 익히고, 문제를 해결하는 체계적인 사고 과정을 거쳐야 한다. 이 단원에서는 소프트웨어 제작과 문제 해결 과정에 대해 알아보고, 텍스트 기반 프로그래밍 언어인 파이썬을 사용하는 방법을 익혀보도록 한다.

# 01



## 문제 해결과 컴퓨팅

### ! 학습 목표 !

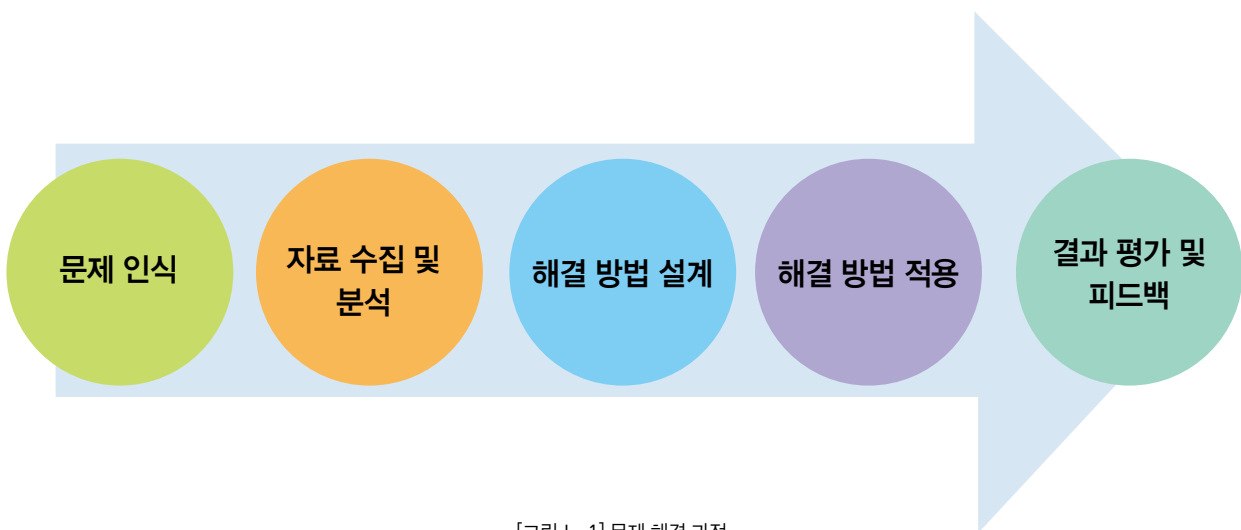
1. 문제 해결을 위해 컴퓨팅 시스템을 활용하는 이유를 설명할 수 있다.
2. 프로그래밍과 프로그래밍 언어가 무엇인지 설명할 수 있다.
3. 소프트웨어 제작 과정을 설명할 수 있다.
4. 파이썬 개발 환경을 설정할 수 있다.

### 1. 문제 해결과 컴퓨팅

#### (1) 문제 해결

현대 사회에는 해결해야 하는 많은 문제들이 있다. 간단한 문제는 다른 도움 없이도 해결할 수 있지만, 문제가 복잡하거나 고려해야 하는 요소가 많다면 혼자 힘으로 해결하기 어려울 때가 있다. 사람들은 이런 어려움을 해결하기 위해 다양한 도움을 필요로 하고 있고, 현대 사회에서는 컴퓨팅 시스템이 이를 위하여 큰 역할을 담당하고 있다.

복잡한 문제를 해결하기 위해서는 일반적으로 다음과 같은 단계를 거친다.



[그림 1-1] 문제 해결 과정

## 문제 해결 과정

### 문제 인식

문제를 해결하기 위해서는 가장 먼저 주어진 상황에서 무엇이 문제인지를 정확하게 알 수 있어야 한다. 문제의 현재 상태를 파악하고 도달해야 할 목표 상태를 파악한다.



### 자료 수집 및 분석

문제 해결에 어떤 자료가 필요한지 생각해 보고 관련된 다양한 자료를 수집한다. 수집한 자료를 바탕으로 문제 해결을 위한 추가 정보들을 도출하고 문제 해결의 단서가 되는 정보를 분류한다.



### 해결 방법 설계

수집하고 분석한 자료와 정보에 근거하여 내가 원하는 목표 상태를 이루기 위한 해결 방법을 고안한다. 해결 방법은 여러 가지일 수 있으며, 다양한 해결 방법 중 가장 효과적이고 효율적인 것을 선택한다.



### 해결 방법 적용

해결 방법을 적용하여 원하는 결과를 도출한다. 해결 방법을 실제로 적용하기 어려운 경우에는 모의 상황이나 컴퓨터 시뮬레이션을 활용하여 가상의 결과를 도출할 수도 있다.



### 결과 평가 및 피드백

해결 방법을 적용하여 원하는 결과가 나타났는지를 확인한다. 해결 방법의 문제나 자료의 오류 때문에 결과가 만족스럽지 않은 경우에는 자료를 추가적으로 수집하거나 해결 방법을 수정하여 원하는 결과가 나올 수 있도록 한다.

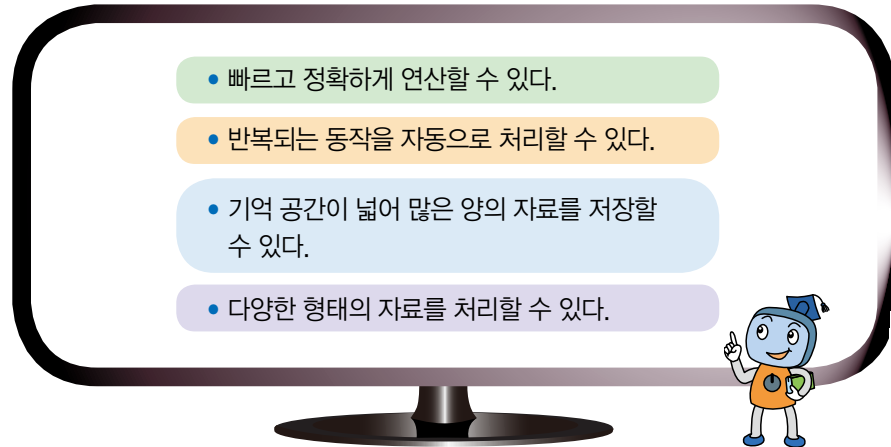




## (2) 컴퓨팅 시스템의 활용

우리가 문제를 해결하기 위해 사용할 수 있는 자원들 중 컴퓨팅 시스템을 사용하는 목적은 무엇일까? 컴퓨팅 시스템은 다음과 같은 특징이 있기 때문에 문제 해결 과정에서 다양하게 활용된다.

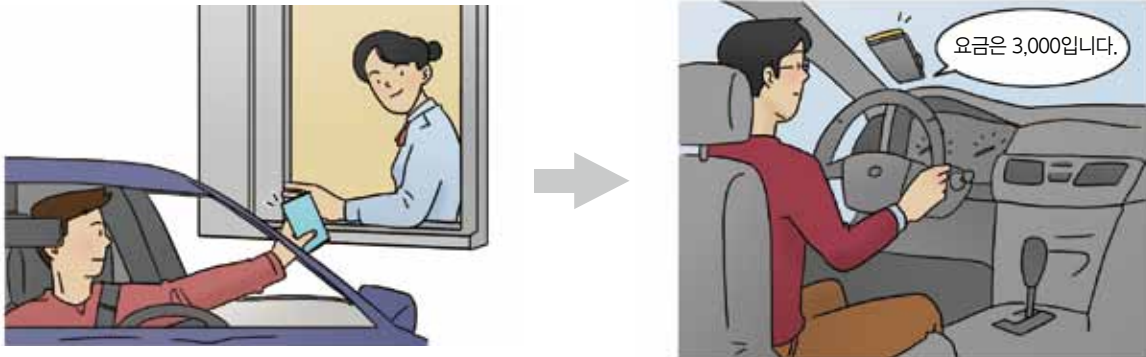
다양한 정보가 디지털화되어 처리되는 현대 사회에서 디지털 정보를 빠르고 정확하게 처리할 수 있는 컴퓨팅 시스템은 문제 해결의 가장 좋은 도구라고 할 수 있다.



[그림 1-2] 컴퓨팅 시스템의 특징

### 1) 소프트웨어 제작의 의미

소프트웨어는 여러 사람이 오랜 시간 반복적으로 수행해야 하는 작업, 사람의 인지 능력으로는 수행하기 어려운 일 등을 컴퓨팅 시스템이 자동으로 수행하도록 함으로써 문제 해결의 속도를 높인다. 아래 그림과 같이 고속도로 요금소에 설치된 통행료 자동 정산 시스템은 자동으로 통행료를 정산해 주는 소프트웨어를 단말기에 설치함으로써 편리함을 추구한 문제 해결의 좋은 예이다.

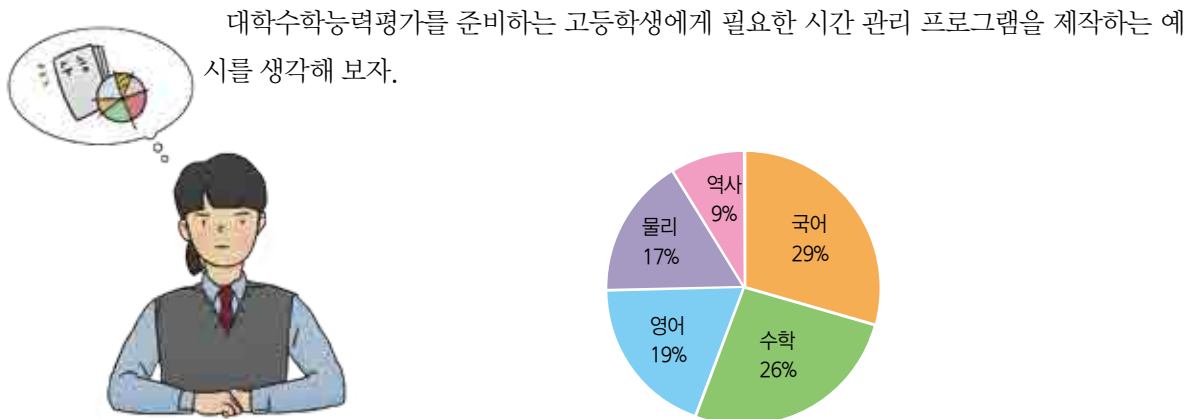


[그림 1-3] 소프트웨어를 활용한 문제 해결의 예

## 2) 소프트웨어 제작 과정

소프트웨어 제작을 결정했다고 해서 바로 프로그래밍을 시작한다면 설계 도면 없이 집을 짓는 것과 같다. 소프트웨어 제작은 프로그래밍만을 의미하지 않는 것과 같은 원리이다. 소프트웨어가 갖추어야 할 기능들은 무엇이고, 이 기능들을 어떻게 구현할 것인가에 대해 생각해볼 수 있어야 한다. 또한 주어진 시간 내에 완성할 수 있도록 하려면 시행착오를 최소화할 수 있어야 한다. 이 모든 것들은 소프트웨어를 어떤 과정을 거쳐 제작할지에 포함되어야 하는 사항들이며, 이 과정이 명확할 때 효과적이고 효율적인 제작이 가능해진다. 따라서 소프트웨어 제작 과정을 설계하는 것은 간과해서는 안 될 중요한 사항이다.





[그림 1-5] 시간 관리 프로그램 제작을 위한 과목별 시간 비중 예시

### ① 문제 인식

소프트웨어를 제작하기 위해서는 제작하고자 하는 목적이 무엇인지를 명확하게 파악하고 있어야 한다. 제작 목적은 소프트웨어가 어떤 기능들을 갖도록 할 것인지, 소프트웨어를 완성하기 위해서는 얼마나 많은 시간과 비용이 투입되어야 구현할 수 있는지, 소프트웨어를 사용하는 대상은 누구인지 등 소프트웨어 제작에 앞서 고려해야 할 사항들을 포함하고 있으며 개발 방향을 제시해 주기 때문이다.

고등학생이 사용할 시간 관리 프로그램이 갖추어야 할 기능들은 무엇이 있을까? 다음과 같이 정리해 볼 수 있다.

제작하려는 소프트웨어를 사용하는 대상이 누구인지에 대하여 생각해 보는 것은 중요한 사항이다. 소프트웨어를 사용하는 사람이 한국 사람인데 아랍어로만 사용할 수 있다면 잘못 제작된 것이다. 또한 7세 어린이가 사용할 소프트웨어인데 전문 용어로 제작되어 있거나 사용법이 복잡하다면 어린이들은 이 소프트웨어를 사용할 수 없게 된다.



- 활동 카테고리 추가 및 삭제
- 하루 시간 사용 패턴 분석
- 일주일 동안의 시간 사용 패턴 분석
- 한 달 동안의 시간 사용 패턴 분석
- 특정 기간 동안의 시간 사용 패턴 분석
- 수행할 활동의 목표 사용 시간 설정 및 도달 여부 분석

### ② 소프트웨어 설계

문제 인식 단계에서 제작 목적을 바탕으로 어떤 소프트웨어를 제작할 것인지 그 기능들을 명세화 하였다면, 소프트웨어 설계 단계에서는 그 기능들을 실제로 어떻게 구현할 것인가에 대하여 설계하도록 한다. 이 단계에서 인터페이스를 디자인하고 소프트웨어의 각 기능이 동작하게 되는 알고리즘을 작성한다.



시간 관리 프로그램 제작으로 다시 돌아가 생각해보자. 먼저 시간 관리 프로그램에서 보여지는 화면 인터페이스를 설계한다. 인터페이스 설계 시에는 내가 활동하는 내역을 등록하는 버튼, 필요없는 활동 내역을 삭제하는 버튼, 날짜별, 주간, 월간, 설정 기간 동안의 활동 시간과 패턴을 분석하는 버튼, 내가 도달해야 할 활동 목표 시간 설정 버튼, 시간 사용과 패턴 분석 결과를 보여주는 창을 화면의 어떤 위치에 보여줄지를 포함한다. 그리고 각 버튼들이 클릭될 때 어떤 동작이 어떻게 수행될지에 대한 처리 알고리즘을 설계한다.

인터페이스

<활동 내역 관리창>

활동 내역 추가 버튼	활동 내역
활동 내역 삭제 버튼	-학교 일정 · 학교 수업
	-학원 · 영어 학원
	· 수학학원
	-식사 · 아침
	· 점심
	· 저녁
	· :

<시간 사용 및 패턴 분석창>

오늘의 시간 사용 내역 버튼	11월 24일
주간 시간 사용 내역 버튼	06:00 아침 운동
월간 시간 사용 내역 버튼	07:00 아침 식사
기간 설정 버튼	08:00 등교 준비
	09:00 :
	:
	:

### 활동 시간 누적 알고리즘

만약 활동 내역이 개인 공부 카테고리이면,  
개인 공부 시간에 활동 내역 시간을 포함.

그렇지 않은 경우, 활동 내역이 학원 카테고리이면,  
학원 시간에 활동 내역 시간을 포함.

그렇지 않은 경우,  
기타 시간에 활동 내역 시간을 포함.

복잡한 과제를 수행할 때에는 소프트웨어 설계 단계가 보다 구체적이고 면밀해야 시행착오를 줄일 수 있다. 설계는 어떤 과제를 어떤 시점에 수행해야 할지를 안내하고 소프트웨어 제작 과정을 한 눈에 파악할 수 있도록 하기 때문이다.

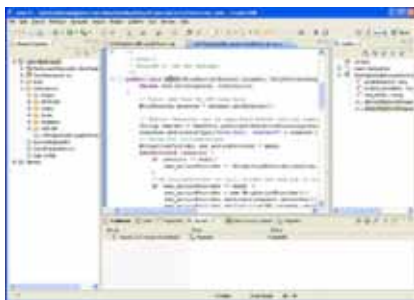


[그림 1-6] 인터페이스 및 알고리즘 설계 예시

### ③ 소프트웨어 구현

소프트웨어를 어떻게 제작할 것인가에 대한 설계를 마치고 나면 이 설계를 바탕으로 소프트웨어를 제작하게 된다. 소프트웨어 구현 단계에서는 소프트웨어 제작에 적절한 프로그래밍 언어를 선택하고 프로그래밍을 통해 각 기능을 구현한다.

시간 관리 프로그램을 구현하는 단계를 생각해 보자. 앞에서 설계한 사용자 화면과 각 기능들의 알고리즘을 토대로 선택한 프로그래밍 언어로 소프트웨어를 구현한다.



[그림 1-7] 프로그래밍 과정



[그림 1-8] 시간 관리 프로그램

#### ④ 테스트와 디버깅

소프트웨어 구현이 완료되면 제작한 소프트웨어가 잘 동작하는지를 시험 운영해 보고, 오류가 발생하면 원인을 찾아 이를 수정해야 한다. 이 단계는 제작한 소프트웨어가 포함해야 할 기능들이 정확히 실행되는지를 확인하는 것뿐 아니라 추가적으로 필요한 기능은 없는지 등을 탐색하고 보다 정교한 소프트웨어를 만들어낼 수 있도록 한다.

시간 관리 프로그램을 구현하고 나면 제대로 동작하는지 테스트를 한다. 시뮬레이션을 통해 시스템의 동작을 확인하거나 컴퓨터나 스마트 기기에 프로그램을 설치하여 테스트를

하는 과정을 거친다. 시간 관리 프로그램에서 자투리 시간을 분석해 주는 기능이 추가되면 좋겠다고 판단되면 다시 설계 단계로 돌아가 자투리 시간 분석 버튼을 사용자 화면의 어느 위치에 둘 것인지 설계하고, 해당 기능의 알고리즘을 작성한 후 소프트웨어를 구현하고 또 다시 테스트와 디버깅 과정을 거치게 된다. 이 과정에서 시간 관리 프로그램은 보다 정교하고 우수한 소프트웨어로 거듭나게 된다.



#### ⑤ 소프트웨어 유지 보수

소프트웨어가 완성되었다고 해서 모든 제작 과정이 종료된 것은 아니다. 시간이 지날수록 새로운 기능들을 추가하게 되기도 하고 불필요하게 된 기능은 제거하기도 한다. 또한 운영체제가 업그레이드되면서 새로운 운영체제에서 동작하도록 하려면 소프트웨어를 변경해야 하는 경우도 발생할 수 있다. 따라서 소프트웨어의 유지 보수는 제작한 소프트웨어를 얼마나 오래 쓸 수 있도록 하는가에 영향을 미치는 중요한 단계이다.

시간 관리 프로그램이 대중화되면 사용자들이 증가하므로 개선 의견들이 제시될 수 있다. 그중 필요하다고 판단되는 의견들을 반영하여 소프트웨어를 보다 우수한 기능을 가진 것으로 개선할 수 있다.

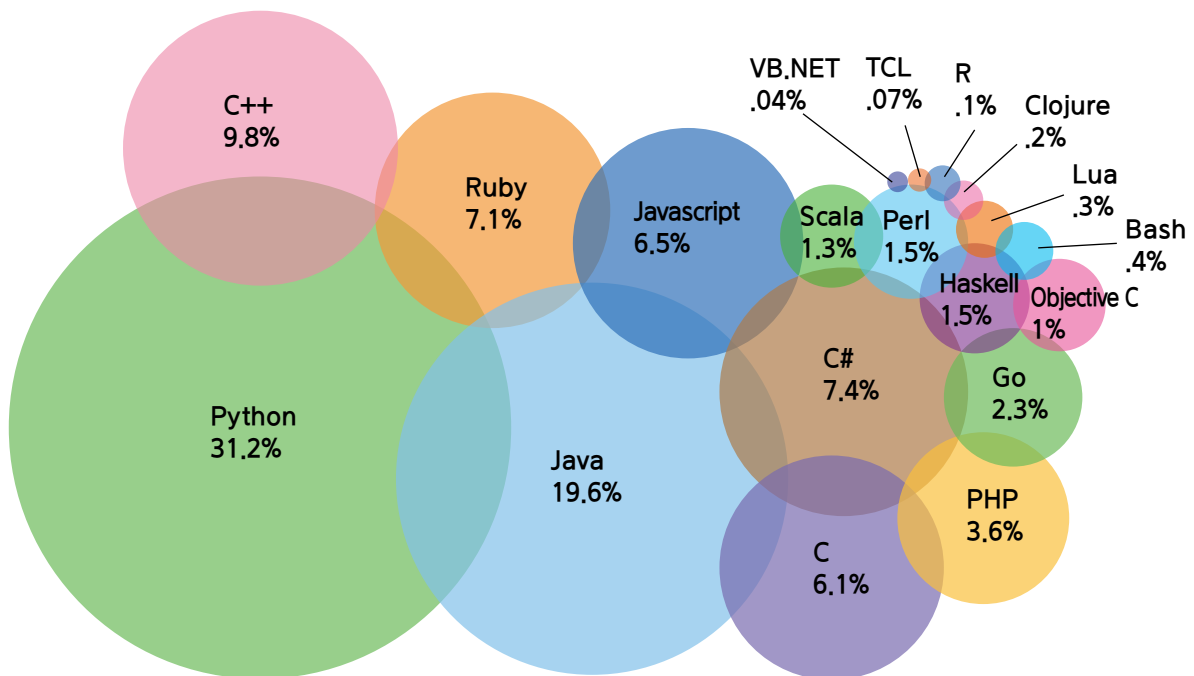
## 2. 프로그래밍

### (1) 프로그래밍 언어

소프트웨어를 제작하기 위해서는 다양한 컴퓨팅 지식이 필요하지만, 그 중에서도 소프트웨어를 제작하는 도구인 프로그래밍 언어를 사용할 수 있어야 한다. 프로그래밍 언어는 컴퓨팅 시스템이 작업을 수행할 수 있도록 명령을 내리는 도구이다. 우리가 다른 사람과 소통하기 위하여 그 사람이 이해할 수 있는 언어를 사용하듯 컴퓨팅 시스템과 소통하기 위해서는 컴퓨팅 시스템이 이해할 수 있는 프로그래밍 언어를 사용해야 한다.

### (2) 프로그래밍 언어의 종류

프로그래머들은 다양한 목적에 따라 개발된 많은 종류의 프로그래밍 언어를 사용해 왔다. 최근에는 블록 기반의 프로그래밍 언어들이 개발되고 있지만, 실제적인 소프트웨어를 제작하기 위해서는 명령어를 직접 입력하는 텍스트 기반의 프로그래밍 언어를 사용할 수 있어야 한다. 요즘 우리가 사용할 수 있는 프로그래밍 언어는 사람이 이해하기 쉬운 형태의 고급 언어이고, 대표적인 고급 언어로는 C, C++, Java, Python 등이 있다.



출처: <http://blog.codeeul.com/codeevalblog/2015>

[그림 1 -9] 2015년 기준 가장 인기있는 프로그래밍 언어



### 3. 파이썬

이 책에서는 소프트웨어 제작 과정을 체험하기 위해 파이썬(Python)이라는 프로그래밍 언어를 사용할 것이다. 파이썬의 특징을 살펴보고, 파이썬 언어를 사용하기 위한 프로그래밍 개발 환경을 갖추는 방법을 알아보자.

#### (1) 파이썬이란

파이썬은 1991년 프로그래머인 귀도 반 로섬이 발표한 고급 프로그래밍 언어이다. 다른 프로그래밍 언어보다 문법이 상대적으로 간단하고 라이브러리가 풍부하여 사용하기 편리하다.

##### 파이썬의 특징

- 다른 언어에 비해 간결하고, 코드의 가독성이 높다.
- 다른 언어와 연결하기 쉽다.
- 인터프리터 언어로 한 번에 한 줄의 코드가 실행되어 실행 상태를 확인하기 쉽다.
- 객체 지향(object-oriented) 언어이다.
- 다양한 라이브러리가 있어 필요한 기능을 찾아 쓰기 쉽다.



#### 파이썬으로 개발한 소프트웨어

비트토렌트	드롭박스	이브 온라인
		
파일 공유 프로그램인 비트토렌트는 개발부터 5.3버전까지 파이썬으로 개발되었다.	클라우드 컴퓨팅을 통한 웹 기반의 파일 공유 서비스인 드롭박스의 운영 프로그램은 파이썬으로 개발되었다.	우주를 배경으로 하는 온라인 게임인 이브 온라인은 파이썬의 한 종류인 스택리스 파이썬으로 개발되었다.

이 외에도 여러 소프트웨어에서 간단한 명령을 처리하기 위한 스크립트 언어로 파이썬이 사용되고 있다. [https://en.wikipedia.org/wiki/List\\_of\\_Python\\_software](https://en.wikipedia.org/wiki/List_of_Python_software) 에서 파이썬이 사용된 여러 소프트웨어의 목록을 볼 수 있다.

## 파이썬 설치와 실행

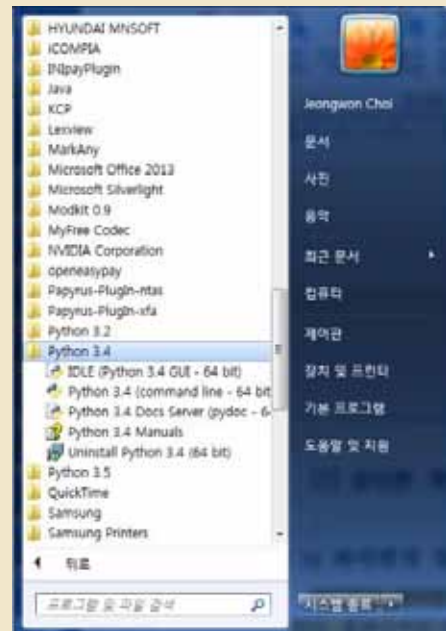
- 파이썬은 공식 홈페이지(<http://www.python.org>)에서 무료로 다운로드할 수 있다. 파이썬 3.4버전을 설치해 보자.



파이썬을 설치할 때는 운영체제와 파이썬 버전을 반드시 확인하도록 한다. 또한 윈도우용 파이썬은 운영체제에 따라 32bit와 64bit 설치 파일이 별도로 있으니 운영체제에 맞게 설치하도록 한다.



- 파이썬의 실행 방식은 여러 가지가 있지만 일반적인 그래픽 환경에서 프로그래밍하기 편리한 IDLE을 사용해 보도록 한다. 파이썬 설치 폴더에서 IDLE을 찾아 실행시켜 보자.



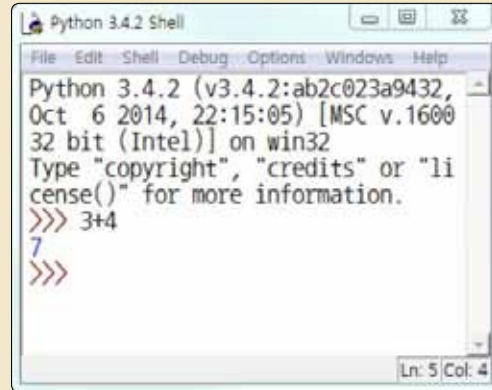
## (2) 파이썬 개발 환경

파이썬은 인터프리터 방식의 프로그래밍 언어이기 때문에 한 줄 한 줄 코드를 입력할 때마다 명령이 바로 실행된다. IDLE에서 명령이 실행되는 것을 확인해 보자.

해 보기

- 오른쪽 그림처럼 “3+4”를 입력하여 결과를 확인해 보자.  
그리고 아래 명령어도 입력해 보자.

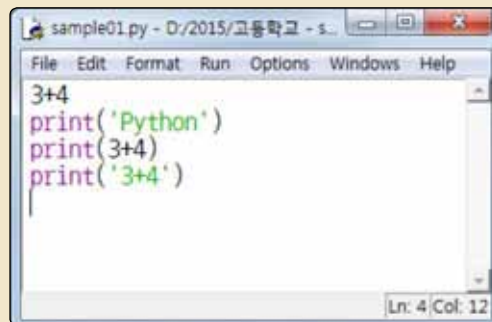
```
>>>print('Python')
```



위의 그림처럼 한 줄씩 코딩하는 창을 셸 창이라고 한다. 셸 창은 프로그램의 진행 결과를 바로바로 확인할 수 있지만, 프로그램 코드를 한 눈에 보기 어렵고 코드의 관리가 편리하지 않아 복잡한 프로그래밍에는 효율적이지 않다. 따라서 일반적으로는 코드를 모두 입력하고 실행시키는 코드 창을 사용한다.

해 보기

- 셸 창의 메뉴에서 [File] - [New File]을 클릭하여 코드 창을 열고, 새 창이 나타나면 오른쪽 그림과 같은 코드를 입력하여 프로그램을 실행시켜 보자.



코드 창에서 작성한 프로그램을 실행하기 위해서는 코딩이 끝난 후 반드시 코드를 저장해야 한다. 작성한 파이썬 프로그램을 실행시키기 위해서는 Run메뉴의 “Run Module”을 클릭한다. 번거로움을 줄이기 위해서는 단축키(F5)를 누르면 편리하다.





# 터틀과 함께 하는 파이썬

## | 학습 목표 |

- 1. 파이썬 프로그램에 모듈을 가져올 수 있다.
- 2. 터틀 모듈을 활용하여 파이썬으로 도형을 그릴 수 있다.
- 3. 순차, 반복, 조건에 관련된 파이썬 명령어를 사용할 수 있다.
- 4. 함수를 만들어 프로그램에 활용할 수 있다.

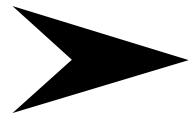
## 1. 파이썬 배우기

파이썬은 다른 프로그래밍 언어보다 문법이 쉽다는 장점이 있지만 기본적으로는 명령어를 일일이 입력하는 방식이기 때문에 명령어에 익숙해질 필요가 있다. 파이썬에 포함되어 있는 터틀(turtle)과 함께 파이썬 명령어를 사용하는 방법을 익혀 보도록 하자.

### (1) 터틀

파이썬에는 그림을 그릴 때 사용하는 터틀이 있다. 오른쪽 그림과 같이 화살표 머리 부분처럼 생긴 터틀이 돌아다니면서 자신의 흔적을 남겨 그림을 그린다.

파이썬에서 터틀을 이용하여 그림을 그리기 위해서는 파이썬 프로그램의 가장 앞부분에 터틀 모듈을 불러와야 한다.



[그림 1-10] 터틀

```
>>>from turtle import *
```

위의 명령을 수행하면 이 코드에서 터틀 모듈을 사용할 수 있다.

### 보충 학습

#### 모듈(module)

모듈은 프로그래밍을 편리하게 할 수 있도록 미리 만들어진 다양한 함수의 집합을 말한다. 파이썬 모듈은 독립적인 파이썬 파일로 만들어져 있고, import 구문을 활용하여 불러올 수 있다. 모듈을 가져올 때는 아래의 구문 방식을 활용한다.

```
>>>from (모듈 이름) import (함수 이름)
```

함수 이름에 와일드카드(\*)를 사용하면 모듈에 포함된 모든 함수를 가져오게 된다.

## (2) 사각형 그리기

컴퓨팅 시스템은 주어진 명령을 정확하고 빠르게 수행할 수 있지만, 명령이 구체적이지 않으면 우리가 원하는 결과를 보여주지 못한다. 우리가 원하는 결과를 얻기 위해 명령을 정확하게 내리는 방법을 익혀 보자.

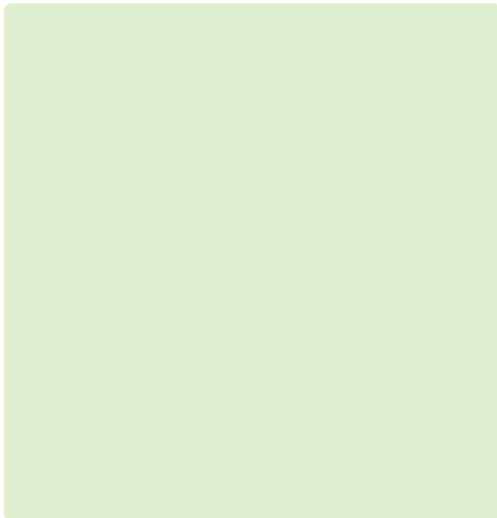


### 1) 단순히 그려 보기

파이썬에서 사각형을 그리기 위해서는 인간이 이해할 수 있는 ‘사각형을 그려라’는 명령 대신 사각형을 그리기 위한 구체적인 동작 하나하나를 명령해야 한다.



아래에다 사각형을 그려 보자.



사각형을 그리는 동작을 설명해 보자.



## 2) 파이썬으로 그려 보기

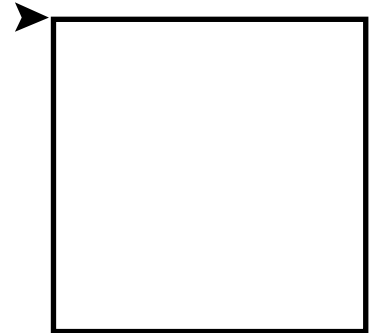
위에서 사각형을 그리는 동작을 구체적으로 구분하여 설명하였다. 이 설명을 파이썬 코드로 옮겨 보자.

임시 코드	파이썬 코드
터틀 모듈을 사용할 것을 선언	<code>from turtle import *</code>
첫 번째 변 그리기(100)	<code>forward(100)</code>
오른쪽으로 90° 회전하기	<code>right(90)</code>
두 번째 변 그리기(100)	<code>(</code>
오른쪽으로 90° 회전하기	<code>(</code>
세 번째 변 그리기(100)	<code>(</code>
오른쪽으로 90° 회전하기	<code>(</code>
네 번째 변 그리기(100)	<code>(</code>
오른쪽으로 90° 회전하기	<code>(</code>

앞으로 이동하기:  
`forward`(픽셀 단위의 이동거리)  
 회전하기: `right`(각도), `left`(각도)



코드를 저장하고 실행시킨 후 오른쪽 그림과 같은 결과가 출력되는지 확인해 보자.

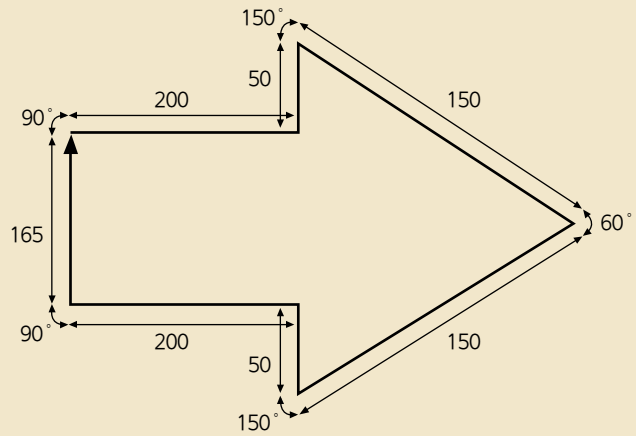


### 픽셀(pixel)

픽셀은 화면을 구성하는 하나의 사각형 모양의 점을 말한다. 흔히 말하는 해상도 수치의 단위가 바로 픽셀이다. 만약 1,024x768의 해상도를 가지는 화면이라면 가로로는 1,024개의 픽셀이, 세로는 768개의 픽셀이 있다는 것을 의미한다. 오른쪽 화살표는 작은 화살표 그림을 확대한 것으로 작은 픽셀이 모여 화살표 테두리를 이룬 것을 확인할 수 있다.



- 오른쪽 그림과 같은 그림을 그리려면 어떻게 해야 할지 프로그램을 작성해 보자.



화살표를 그리는 동작

파이썬 코드

```
from turtle import *
```

### 3) 효율적으로 그리기

앞에서 작성했던 사각형 그리기 코드를 다시 살펴보자.

```
from turtle import *  
forward(100) ..... ①  
right(90) .....  
forward(100) ..... ②  
right(90) .....  
forward(100) ..... ③  
right(90) .....  
forward(100) ..... ④  
right(90) .....
```

위의 코드에서는 터틀이 앞으로 100픽셀 간 후 오른쪽으로 90도 회전하는 명령이 총 4회 등장한다. 이런 코드는 효율적이지 않은 프로그램을 작성한 것이라 볼 수 있다. 이럴 때는 특정 명령이 자동으로 반복되도록 함으로써 효율성을 높일 수 있다.

해 보기

▶ 사각형을 그리는 코드를 아래와 같이 바꾸어보고 실행 결과를 살펴보자.

```
from turtle import *  
for i in range(4):  
    forward(100)  
    right(90)
```

#### 들여쓰기

파이썬에서 들여쓰기는 포함 관계를 의미한다. 들여쓰기 된 명령은 한 단계 위의 명령에 포함되어 있는 명령이라는 의미이다. 위의 코드에서는 반복하고자 하는 forward와 right 명령을 for 명령어보다 한 단계 들여쓰기하여 파이썬에 for 명령에 따라 반복되어야 하는 내용임을 알려준다.

위의 코드에서 두 번째 줄 ‘for i in range(4):’는 i가 0부터 3까지의 숫자로 4번 변하는 동안 for문 안의 명령을 반복하라’는 의미를 갖는 명령이다. 위의 코드가 실행되는 과정을 나타내면 아래 표와 같다. i의 값이 0부터 시작하여 하나씩 증가되어 가는 과정에서 for에 내포된 forward(100)과 right(90) 명령이 반복적으로 수행하도록 한다.

i=0	i=1	i=2	i=3
forward(100) right(90)	forward(100) right(90)	forward(100) right(90)	forward(100) right(90)

이렇게 같은 명령을 연속적으로 실행시키는 명령 구조를 반복 구조라고 한다.

### for

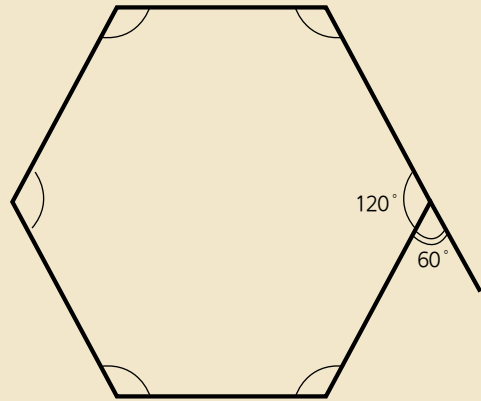
for 문은 특정한 횟수만큼 안쪽에 포함된 명령을 수행하게 하는 명령이다. 앞에서처럼 range( ) 함수와 함께 사용하여 지정된 횟수만큼 반복하게 할 수도 있고, 특정한 자료 집합과 함께 사용하여 해당 자료 집합만큼 반복하게 할 수도 있다.

### range()

range( )는 괄호 안에 표시된 숫자만큼의 범위를 표현하는 명령어이다. 단, 숫자 범위의 시작은 0부터 시작된다. 따라서 range(4)는 4개의 숫자 범위를 표현하지만 0부터 시작하여 1, 2, 3이 될 때까지의 범위를 의미한다. 즉, 4는 포함되지 않으며 4가 되기 전까지 정수의 범위를 나타낸다.



- 다음과 같이 한 변이 100픽셀인 정육각형이 있다. 이를 그리려면 어떻게 해야 할지 프로그램을 작성해 보자.



## 정육각형을 그리는 동작

## 파이썬 코드

```
from turtle import *
```

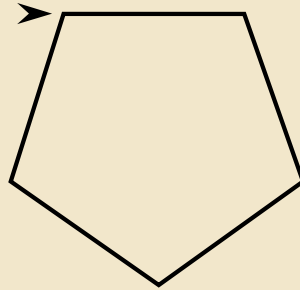


## 2. 정다각형 그리기

앞에서 사각형을 그리기 위해서 사각형 각 변의 길이와 회전하는 각도를 설정하고 변의 개수만큼 명령을 반복하였다. 그렇다면, 이 코드를 활용해서 다른 다각형을 그려 보자.

해 보기

- 3이라고 입력하면 삼각형을, 4라고 입력하면 사각형을, 5라고 입력하면 오각형을 그리는 프로그램을 작성해 보자.  
(단, 한변의 길이는 100픽셀이다.)



```
>>> ----- RESTART -----  
>>> 숫자를 입력해주세요 (삼각형: 3, 사각형: 4, 오각형: 5): 5  
>>> |
```

- 위의 프로그램 실행 화면을 보고 이 프로그램의 동작을 단계별로 간략히 설명해 보자.

Blank area for writing the explanation of the program's operation.

## 1) 알고리즘 설계

입력받은 숫자에 따라 다각형을 그리는 프로그램을 제작하기 위해서는 어떤 방법과 절차로 만들 것인지에 대한 구체적인 알고리즘 설계가 먼저 이루어져야 한다. 알고리즘을 미리 설계해 보면 프로그래밍의 방향을 안내하고 어떤 기능들을 구현해야 할지를 눈으로 확인할 수 있기 때문에 코딩을 진행하는 데 도움이 된다.

해 보기

❶ 위의 프로그램 동작 단계를 참고하여 아래 프로그램 구현을 위한 알고리즘을 완성해 보자.

### 알고리즘

1. 터틀을 불러온다.
2. 숫자 입력(3, 4, 5)을 물어본다.
3. 만약 숫자가 3이라면
4. ( )을 그리는 코드를 실행시킨다.
5. ( )
6. 사각형을 그리는 코드를 실행시킨다.
7. ( )
8. ( )



## (2) 프로그램 구현

앞에서 제시된 다각형 그리기 알고리즘을 바탕으로 각 단계별로 코딩을 진행해 보자.

해 보기

● 새 코드 창을 띄우고 각 단계별로 코딩한다.

① 터틀 모듈을 불러온다.

```
>>>from turtle import *
```

② 숫자 입력을 물어본다.

이 부분은 3~5까지의 정수 중 하나를 입력받는 부분이다. 따라서 할 일은 다음과 같다.

- \* 입력 안내하기
- \* 키보드로부터 숫자 입력 받기
- \* 입력 받은 숫자 저장하기

변수의 이름을 정할 때에 그 변수가 어떤 용도로 사용되는지 쉽게 알 수 있도록 이름을 정하면 코딩시 직관적으로 활용할 수 있다.



위의 기능을 모두 구현하기 위한 코드는 다음과 같다.

```
num = input('숫자를 입력해주세요(삼각형: 3, 사각형: 4, 오각형: 5): ')
```

위의 코드에서는 임의로 num이라는 변수를 사용하였다. 프로그래밍에서 변수는 값을 저장할 수 있는 공간이라고 말할 수 있다. 수학에서의 변수 개념과 달리 프로그래밍에서 사용하는 변수는 자료의 종류를 구별하고(정수, 실수, 문자형 등), 그 값이 언제든지 바뀔 수 있는 하나의 주소와 같은 역할을 담당한다.

위의 코드를 저장하고 실행시키면 다음과 같은 결과를 확인할 수 있다.

```
>>>  
숫자를 입력해주세요 (삼각형: 3, 사각형: 4, 오각형: 5):
```

보충 학습

input( )

input( )은 키보드의 입력을 받아 변수에 저장하는 역할을 하는 함수이다. 괄호 안에 사용자에게 전달하는 메시지를 입력할 수 있으며, 입력한 내용은 문자 형태의 값으로 저장된다.



❶ 위의 코드를 완성했다면 다음 단계의 명령도 파이썬 코드로 만들 수 있을 것이다.

- ⑤ 만약 숫자가 4라면
- ⑥ 사각형을 그리는 코드를 실행시킨다.
- ⑦ 만약 숫자가 5라면
- ⑧ 오각형을 그리는 코드를 실행시킨다.

❶ 위의 코드를 활용하여 아래에 비어 있는 코드를 완성해 보자. 코드를 완성한 후 프로그램을 실행시켜 보고 정상적으로 동작하는지 확인해 보자.

```
elif num == '4':  
    (  
        (  
            (  
  
elif (  
    (  
        (  
            (  

```

## 보충학습

elif, else

elif는 if와 함께 사용하는 명령어로 if에서의 조건이 거짓인 경우 다시 또 다른 조건의 참과 거짓을 판단할 때 사용한다. elif는 “그렇지 않은 경우 만약 ~라면”이라는 의미를 갖고 있으며 영어에서 else if를 줄여 표현한 것이다.

else는 if와 elif의 조건이 거짓인 경우 수행하게 될 명령을 작성할 때 사용한다. 따라서 “그 밖의 경우”라는 의미를 갖고 있다. 따로 수행할 명령이 없다면 생략할 수 있다.

주의할 점은 if, elif, else 명령어를 사용한 조건문을 만들 때 if, elif, else가 사용된 줄의 마지막에 반드시 콜론(:)을 써 주어야 파이썬이 조건문을 정확하게 파악할 수 있다.



### (3) 효율적인 코딩

앞에서 작성한 코드를 살펴보자. 더욱 효율적으로 코딩할 수 있는 부분은 없는지 생각해 보자.

해 보기

- 위에서 작성한 삼각형, 사각형, 오각형을 그리는 코드를 살펴보고, 각 코드 간의 공통점에 대해 설명해 보자. 반복적으로 활용할 수 있는 부분이 있는가?

공통적으로 사용되는 부분

- 위의 공통점을 활용하여 n각형을 그리는 방법을 설명해 보자.

n각형 그리기

- 앞에서 작성한 프로그램의 알고리즘을 다음과 같이 변경한다.

프로그램 알고리즘

터틀을 불러온다.  
숫자 입력을 물어본다 - 3, 4, 5  
만약 숫자가 3이라면  
    삼각형을 그리는 코드를 실행시킨다.  
만약 숫자가 4라면  
    사각형을 그리는 코드를 실행시킨다.  
만약 숫자가 5라면  
    오각형을 그리는 코드를 실행시킨다.

터틀을 불러온다.  
n각형을 그리는 함수를 정의한다.  
만약 숫자가 3이라면  
    위의 함수를 불러와 삼각형을 그린다.  
만약 숫자가 4라면  
    위의 함수를 불러와 사각형을 그린다.  
만약 숫자가 5라면  
    위의 함수를 불러와 오각형을 그린다.

① n각형을 그리는 함수를 만든다.

함수는 전달받은 값을 활용하여 특정한 명령을 수행한 다음 결과를 돌려주는 프로그램 안의 작은 프로그램이라고 할 수 있다. 함수는 프로그램에서 반복적으로 사용할 수 있고, 프로그램의 구조를 간단하게 만들어 준다.

위에서 작성했던 n각형을 그리는 단계를 활용하여 다음 함수를 작성한다.

```
from turtle import *

def makepolygon(num):
    for i in range(
    )
        forward(100)
        right(
    )
```

보충  
학습

def

def 명령어는 내가 활용하고자 하는 함수를 만들어 주는 명령어이다. def 명령은 아래와 같이 사용할 수 있다.

>>> def 함수이름(전달받는 값):

괄호 안에는 전달받은 값이 들어가 def에 내포된 부분에서 활용할 수도 있다. def 아래쪽의 명령은 들여쓰기를 해야 하며, 프로그램의 다른 부분에서 함수를 사용하기 위해서는 함수 이름과 입력값을 전달해 주면 된다.

② 위의 함수를 불러와 삼각형을 그린다.

위에서 함수를 작성하였으면 이제 활용할 차례이다. 앞에서 작성했던 삼각형 코드를 다음과 같이 변경해 보자.

알고리즘

```
if num == '3':
    for i in range(3):
        forward(100)
        right(120)
```

```
if num == '3':
    makepolygon(int(num))
```

`int()`

`int()` 명령은 괄호 안의 실수나 문자열 값을 정수로 바꾸어 주는 명령이다. 여기에서 `num`을 바로 대입하지 않은 이유는 `input`을 통해 받은 값은 항상 문자열이기 때문이다.

③ 사각형과 오각형을 그리는 명령도 위와 같이 변경해 보자.

## 사각형 그리기

```
elif num == '4':
    for i in range(4):
        forward(100)
        right(90)
```

```
elif num == '4':
    (                )
```

## 오각형 그리기

```
elif num == '5':
    for i in range(5):
        forward(100)
        right(72)
```

```
(                )
(                )
```



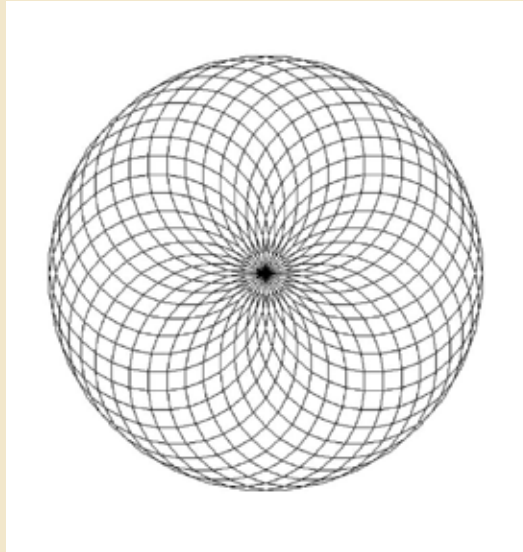
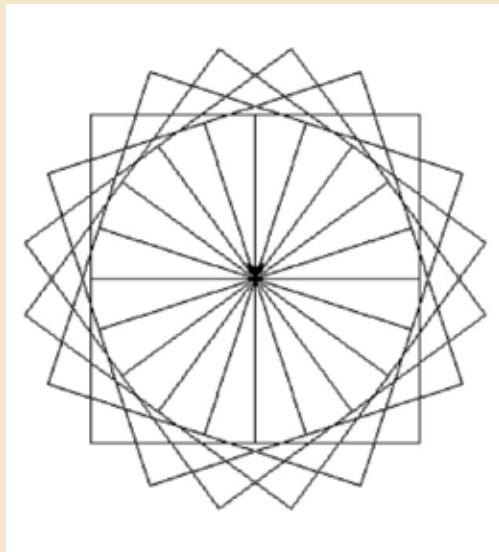
- 앞에서 만든 다각형 그리기 프로그램은 '3', '4', '5' 외에 다른 문자가 입력되면 아무런 수행을 하지 않는다. '3', '4', '5' 외에 다른 문자가 입력될 경우 "잘못 선택하셨습니다."라는 안내 메시지를 출력해 준다면 보다 사용하기 편리한 프로그램이 될 수 있다. 프로그램을 수정해 보자.



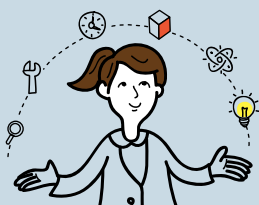
- 앞에서 만든 다각형 그리기 프로그램은 한 번만 입력을 받고, 입력받은 내용에 따라 다각형을 그리는 것으로 종료되는 프로그램이었다. 만약, 다각형을 연속해서 여러 개 그리고 싶다면 어떻게 해야 할까? 그리고 다각형 그리기를 멈추고 싶을 때 "quit"를 입력하여 종료하도록 해보자.



- 터틀을 이용해서 다음 그림을 그려 보자.



- ① 현대 사회의 복잡한 문제를 해결할 때는 문제 인식, 자료 수집 및 분석, 해결 방법 설계, 해결 방법 적용, 결과 평가 및 피드백의 순서를 거친다.
- ② 컴퓨팅 시스템은 디지털화된 자료를 빠르고 정확하게 처리할 수 있기 때문에 문제 해결에 활용된다.
- ③ 소프트웨어를 제작하기 위해서는 다양한 컴퓨팅 지식과 함께 프로그래밍 언어를 알아야 한다. 프로그래밍 언어는 사람이 컴퓨팅 시스템에게 명령을 내리기 위해 필요한 언어로 현재는 사람이 이해하기 쉬운 형태의 고급 언어가 사용된다.
- ④ 파이썬은 프로그래밍하기 쉬우면서 빅데이터 분석, 웹 앱 제작, 게임 제작 등 많은 소프트웨어들을 구현할 수 있는 텍스트 기반 프로그래밍 언어이다. 파이썬에서 작성된 프로그램은 왼쪽에서 오른쪽으로, 위에서 아래로 실행된다.
- ⑤ 파이썬에서 터틀로 그림을 그리기 위해서는 터틀의 동작을 제어하는 명령들을 포함하는 터틀 모듈을 불러와야 한다. 터틀 모듈을 불러오면 파이썬은 사용자가 터틀을 사용할 것이니 준비를 하라는 요청으로 인식하고, 작성되는 명령들을 터틀 동작과 관련하여 해석한다.
- ⑥ 터틀이 움직이도록 하기 위해서는 이동 방향을 제시하는 동작 유형과 이동 거리나 회전 각도를 제시하는 수치값이 필요하다.
- ⑦ 같은 명령을 일정 횟수만큼 반복하기 위한 반복문을 만들기 위해서는 for 명령어를 사용한다. 그리고 반복 횟수는 range() 함수를 사용하여 구현한다.
- ⑧ 변수는 변하는 내용을 저장해 놓을 때 사용하는 공간으로 숫자, 문자, 문자열 등 다양한 내용을 저장할 수 있다. 변수를 사용하기 위해서는 변수 이름을 정하고 변수에 저장될 값을 삽입하면 된다.
- ⑨ 특정 내용을 입력받고자 할 때에는 input(), 특정 내용을 출력하고자 할 때에는 print() 함수를 사용한다. 그리고 보여주는 내용이 문자열일 때에는 함수의 괄호 안에 작은따옴표(' ')로 묶어 작성하며, 변수일 경우에는 변수명을 작성한다.
- ⑩ 파이썬에서 등호(=)는 수학에서처럼 같다는 의미를 갖는 것이 아니라 등호 오른쪽의 내용을 왼쪽으로 저장한다는 의미를 갖고 있다. 그리고 수학에서의 같다를 표현하기 위해서는 2개의 등호(==)를 쓴다.
- ⑪ 특정 조건의 참 거짓에 따라 서로 다른 동작이 수행되어야 할 경우 if, elif, else의 명령을 사용하여 조건문을 만든다. 단, 조건문을 만들 때 문장의 마지막에 반드시 콜론(:)을 사용해야 한다.
- ⑫ 비연속적으로 같은 명령을 여러 번 활용해야 할 경우 미리 특정한 명령의 집합을 만들어 놓고 필요한 시점에 호출하여 사용하는 것을 함수라 한다. 함수는 프로그래밍의 효율성을 높일 수 있으며, 함수를 정의할 때에는 “def 함수명(전달받은 값):”로 시작하며 함수가 호출되기 전에 정의되어 있어야 한다.





## A central computer mouse is the focal point, with a thick grey cord extending upwards. Radiating from the mouse are several line-art icons: a hand clicking a button, a hand placing a cube on top of another, three interlocking gears, a magnifying glass over a bar chart, a network diagram with three nodes, a lightbulb, a computer monitor displaying a bar chart, a megaphone, a paper airplane, and a small cloud. The background is a solid light green.

# 파이썬과 문제 해결

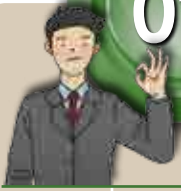
01 알고리즘

02 간단한 문제 해결

03 알고리즘 최적화

최종적으로 소프트웨어를 제작하기 위해서는 프로그래밍 언어로 표현하는 과정이 필요하지만, 문제 해결 과정 자체는 논리적인 사고 과정이다. 특정한 문제에 대한 해결 방법을 논리적인 알고리즘으로 표현한 다음 이를 프로그래밍 언어로 옮기게 되는 것이다. 이 단원에서는 다양한 논리적 문제를 프로그래밍을 통해 해결하는 과정을 경험하면서 컴퓨팅 시스템으로 문제를 해결하는 과정에 어떠한 특징이 있는지 알아보도록 한다. 또한 보다 효율적인 알고리즘을 제작하기 위한 사고 과정을 익히도록 한다.

# 01



## 알고리즘

### Ⅰ 학습 목표 Ⅰ

1. 알고리즘의 개념을 설명할 수 있다.
2. 알고리즘의 조건을 설명할 수 있다.
3. 여러 가지 알고리즘의 표현 방법에 대해 설명할 수 있다.
4. 알고리즘의 효율성을 평가할 수 있다.



### 1. 알고리즘 개요

#### (1) 알고리즘의 의미



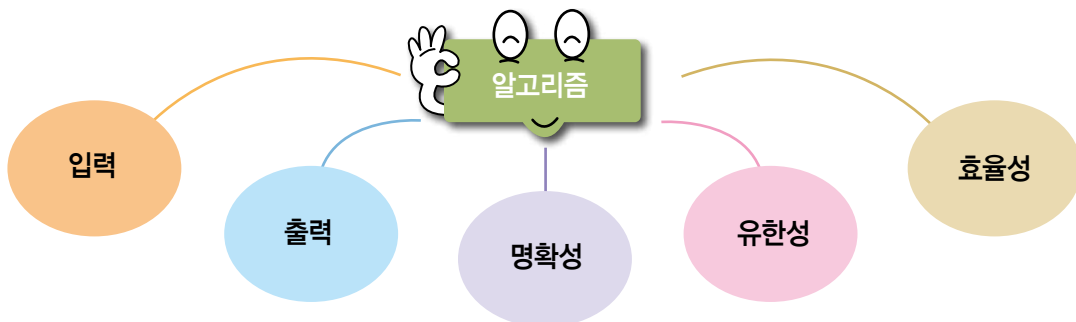
[그림 II-1] 알콰리즈미

알고리즘(Algorithm)은 페르시아의 수학자 알콰리즈미(Al-Khwarizmi)에서 유래한 말로 어떤 문제를 해결하는 방법이나 절차를 의미한다. 우리가 문제를 해결하기 위해서는 적절한 문제 해결 알고리즘을 설계해야 하며, 제작한 알고리즘을 컴퓨터가 이해할 수 있는 명령어 형태로 표현하여 처리하도록 하면 컴퓨터가 해당 문제를 해결할 수 있다.

#### (2) 알고리즘의 조건

일반적으로 알고리즘은 입력, 출력, 명확성, 유한성, 효율성 등의 조건을 만족하여야 한다.

- ① 입력: 외부로부터 입력되는 자료가 0개 이상 존재한다.
- ② 출력: 적어도 1개 이상의 결과가 출력된다.
- ③ 명확성: 수행 과정은 명확하고 모호하지 않은 명령어로 구성되어야 한다.
- ④ 유한성: 유한 번의 명령어를 수행 후 종료되어야 한다.
- ⑤ 효율성: 모든 과정은 명백하게 실행 가능한 것이어야 한다.

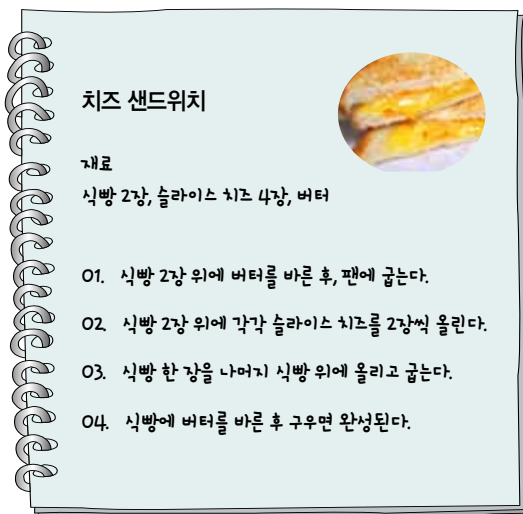


[그림 II-2] 알고리즘의 조건

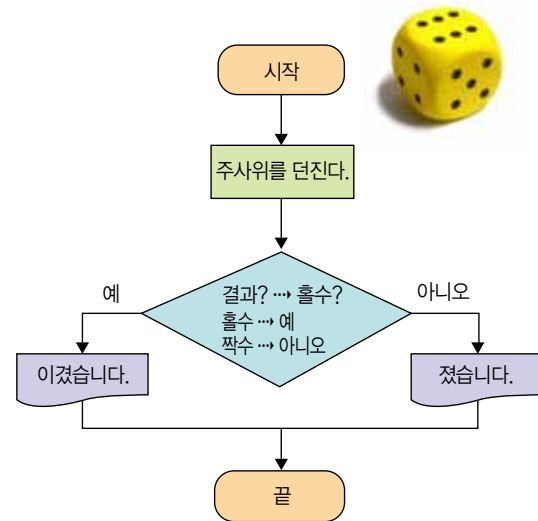
### (3) 알고리즘의 표현 방법

알고리즘을 표현하기 위해서는 다양한 방법들이 존재한다. 그 중에서 자연어, 순서도, 의사 코드, 프로그래밍 언어 등을 주로 사용한다.

자연어 표현 방식은 우리가 사용하는 일반적인 말과 글로 알고리즘을 표현하는 방식이고, 순서도 표현 방식은 약속된 기호와 흐름선을 이용하여 처리 과정을 표현하는 것이다. 의사 코드 표현 방식은 프로그래밍 언어와 유사한 언어로 알고리즘을 표현하는 것이며, 프로그래밍 언어 표현 방식은 컴퓨터에서 실행 가능한 프로그래밍 언어로 알고리즘을 표현하는 것으로 이렇게 표현된 것을 프로그램이라고 한다.



[그림 II-3] 샌드위치 레시피(자연어)



[그림 II-4] 주사위 게임(순서도)

```
sum←0
i←1
begin
    sum←sum + i
    i←i + 1
if i is not 11 goto begin
print sum
```

[그림 II-5] 1~10의 합(의사코드)

```
sum=0
for i in range(1, 11):
    sum=sum + i
print(sum)
```

[그림 II-6] 1~10의 합(파이썬)

## 2. 알고리즘의 성능 평가

어떤 문제를 해결하는 알고리즘은 한 가지만 있는 것이 아니다. 여러 가지 알고리즘 중에서 보다 문제 해결에 효율적인 것을 찾기 위해 알고리즘의 성능을 평가할 필요가 있다. 본 교재에서는 문제 해결에 필요한 실행 시간과 메모리 사용량이라는 2가지 기준을 주로 사용한다.

### 1) 실행 시간

어떤 문제를 해결하는 데 걸리는 시간이 짧은 알고리즘이 보다 효율적이다. 예를 들어 1부터 N까지의 모든 수의 합을 구하는 문제가 있다고 가정하고, 이 문제를 풀기 위한 다음 2가지 코드를 비교해 보자.



〈code II-1〉

```
① n=int(input())
② answer = n*(n+1)/2
③ print(answer)
```

〈code II-2〉

```
① n=int(input())
② answer=0
③ for i in range(1,n+1):
④     answer += i
⑤ print(answer)
```

두 코드 모두 1부터 N까지의 합을 구하는 알고리즘을 표현하고 있다. 하지만 두 코드를 실행해 보면 차이가 있음을 알 수 있다.

두 코드를 실행하고 입력값으로 10000 정도의 작은 값을 입력하면 즉시 동일한 결과를 출력한다. 하지만 10000000을 입력하면 〈code II-1〉은 즉시 결과를 출력하지만 〈code II-2〉는 수십 초의 시간이 흐른 후 결과를 출력함을 알 수 있다.

코드를 잘 살펴보면 〈code II-1〉은 N의 크기가 변해도 실행 시간에 큰 차이가 나지 않지만 〈code II-2〉는 N의 크기가 커질수록 그에 비례해서 실행 시간이 길어진다. 그 이유는 4행의 명령문이 N번 반복되기 때문이다. 만약 중첩된 반복문 내에 명령문이 있다면  $n^2$  번 반복될 것이고, 불과 10000 정도의 입력값을 넣어도 수십 초의 처리 시간이 필요하다. 따라서 알고리즘을 설계할 때 가급적 불필요한 반복을 줄이는 것이 실행 시간 단축에 도움이 된다.

## 2) 메모리 사용량

컴퓨터에서 코드를 수행할 때 사용 가능한 메모리는 제한적이다. 따라서 실행 시간에 큰 차이가 없다면 메모리를 적게 사용하는 알고리즘이 효율적이다.

예를 들어 1에서 10000 사이의 정수 10개를 입력받아 그중 최솟값을 구하는 문제가 있을 때 이 문제를 풀기 위한 다음 2가지 코드를 비교해 보자.

〈code II-3〉

```
① num=[]
② for i in range(10):
③     num.append(int(input()))
④ num.sort()
⑤ print(num[0])
```

〈code II-4〉

```
① min=10000
② for i in range(10):
③     num=int(input())
④     if min>num:
⑤         min=num
⑥ print(min)
```

두 코드 모두 10개의 입력값 중 최솟값을 구하는 알고리즘을 표현하고 있다. 〈code II-3〉은 입력값을 모두 리스트에 저장하고, `sort()`를 이용하여 오름차순으로 정렬한다. 그리고 리스트의 첫 값을 출력함으로써 최솟값을 출력한다. 따라서 입력 데이터의 수  $N$ 이 커지면 리스트에 저장되는 입력값의 수도 증가하므로 메모리 사용량이 늘어난다.

반면 〈code II-4〉는 최솟값을 저장할 `min`과 입력값을 임시 보관할 `num` 2개의 변수만 사용하여 최솟값을 출력하고 있다. 즉,  $N$ 의 개수와 상관없이 늘 2개의 변수 공간만 확보하면 되므로 메모리 사용량이 일정하다. 따라서 메모리 사용량의 측면에서 볼 때 〈code II-4〉가 효율적이다.

또 재귀 방식을 사용하는 알고리즘은 호출하는 깊이가 깊을수록 많은 양의 메모리를 사용하므로 이 또한 유심히 살펴야 한다. 메모리 사용량을 고려하지 않으면 컴퓨터의 가용 메모리를 초과하여 알고리즘 자체에는 이상이 없더라도 프로그램 코드가 실행되지 않을 수 있으므로 주의해야 한다.



## 02



## | 학습 목표 |

## 간단한 문제 해결

1. 조건문을 사용하여 문제를 해결할 수 있다.
2. 반복문을 사용하여 문제를 해결할 수 있다.
3. 리스트를 사용하여 문제를 해결할 수 있다.

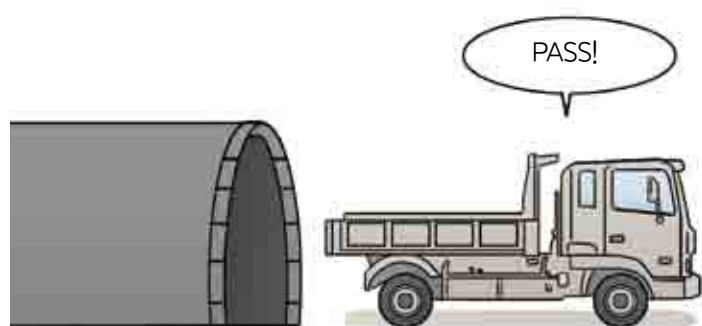


이 단계의 문제는 입·출력문, 조건문, 반복문을 활용하여 해결할 수 있는 문제로, 복잡한 문제를 해결하기 전에 프로그래밍 언어를 활용한 문제 해결 과정을 연습하기 위한 것이다.

## 1. 조건문 활용

## (1) 터널 통과하기

어떤 차의 높이는 170cm이다. 이 차가 목적지에 도착하기 위해서는 3개의 터널을 차례대로 지나게 된다. 터널의 높이가 차의 높이보다 같거나 낮다면 차는 터널과 충돌하여 사고가 날 것이다. 3개 터널의 높이가 공백으로 구분되어 주어졌을 때 터널을 무사히 잘 통과하면 “PASS”를 출력하고, 사고가 난다면 “CRASH”를 출력해 보자. 단, 터널의 높이는 모두 정수이다.





### 1) 문제 분석

문제를 분석하여 필요한 기능이 무엇인지 정리한다.

- 3개의 정수를 입력받아야 함. → `input( )`, `split( )` 함수 사용, 3개의 변수 필요
- 입력값이 차의 높이 이하인지 검사 → 판별 함수 작성
- 판별결과를 문자로 출력해야 함. → `print( )` 함수 사용



### 2) 알고리즘 설계

차와 터널의 충돌여부를 판별하는 프로그램의 알고리즘을 생각하여 자연어로 표현한다.

1. 터널의 높이를 입력받아 a, b, c 변수에 저장한다.
2. 만약  $a \leq 1700$ 이거나  $b \leq 1700$ 이거나  $c \leq 1700$ 이면 "CRASH"를 출력한다.
3. 그렇지 않으면 "PASS"를 출력한다.

### 3) 코딩

위에서 작성한 알고리즘을 바탕으로 파이썬 프로그램을 작성해 보자.

### 4) 테스트/디버깅

프로그램이 완성되었다면 실행시켜 보고, 여러 가지 데이터를 입력하여 잘 동작하는지 확인해 보자.

입력 데이터	예상 결과	확인 및 수정할 사항
170 171 172	CRASH	
171 172 173	PASS	
190 200 169	CRASH	

## (2) 아르바이트 가는 날

주원이는 월, 수, 금, 일요일에 아르바이트를 한다. 요일의 번호를 입력해서 그 날이 아르바이트 가는 날이면 “oh my god”를 출력하고, 쉬는 날이라면 “enjoy”를 출력하는 프로그램을 작성해 보자. 입력하는 요일 번호는 아래와 같다.

월요일(1), 화요일(2), 수요일(3), 목요일(4), 금요일(5), 토요일(6), 일요일(7)



해 보기

### 1) 문제 분석

문제를 분석하여 필요한 기능이 무엇인지 정리한 것이다. 빈 칸을 완성해 보자.

- 1개의 정수를 입력받아야 함. → `input( )` 함수 사용
- 입력된 요일이 아르바이트 하는 날인지 판별해야 함. → 판별 함수 작성
- 

### 2) 알고리즘 설계

아르바이트하는 날인지 판별하는 함수의 알고리즘을 생각하여 자연어로 표현해 보자.

### 3) 코딩

위에서 작성한 알고리즘을 바탕으로 파이썬 프로그램을 작성해 보자.

### 4) 테스트/디버깅

프로그램이 완성되었다면 실행시켜 보고, 여러 가지 데이터를 입력하여 잘 동작하는지 확인해 보자.

입력 데이터	예상 결과	확인 및 수정할 사항
1	oh my god	
2	enjoy	
3	oh my god	

### 5) 개선/발전 방향

아르바이트하는 요일인지 판별하는 함수를 만들 때 조건문이 복잡하지는 않은가? 만약 복잡한 명령문을 사용했다면 개선할 수 있는 방법이 없는지 고민하고 코드를 작성해 보자.

#### 〈힌트〉

아르바이트 가는 날을 표시하는 요일의 공통 점을 찾아보면 된다.



### (3) 30분 전



수호는 시간을 되돌리는 타임머신을 만들고 싶은 사람이다. 수호는 일단 30분 전으로 돌아가는 타임머신부터 설계하려고 한다. 공백으로 나누어진 시간을 입력하여 정확히 30분 전의 시간을 출력하는 프로그램을 작성해 보자.

예)	12 35	→	12 5
	12 0	→	11 30
	21 5	→	20 35

해 보기

#### 1) 문제 분석

문제를 분석하여 필요한 기능이 무엇인지 정리한 것이다. 빈칸을 완성해 보자.

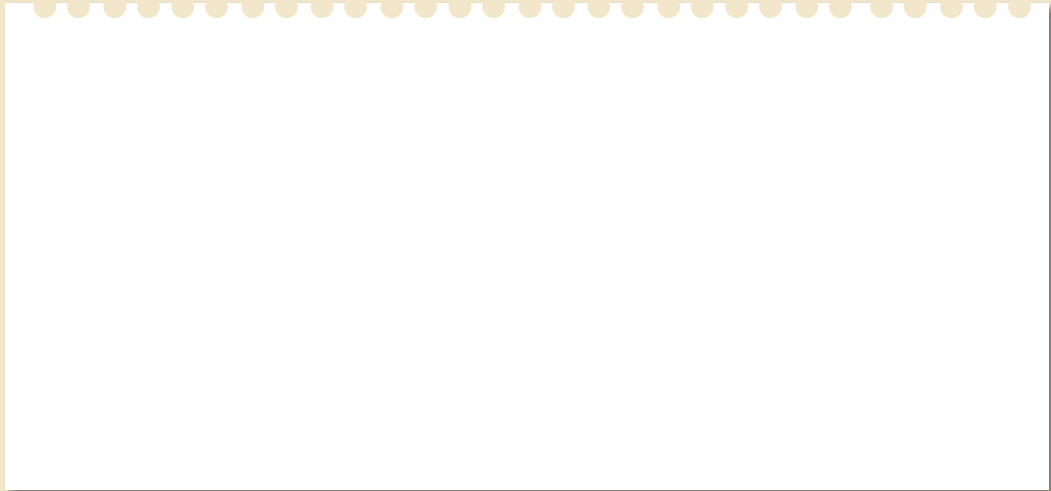
- 
- 입력 시간보다 30분 이전 시간을 계산해야 함.
- 2개의 정수를 출력해야 함. → `print( )` 사용

#### 2) 알고리즘 설계

시와 분을 구할 때 고려해야 되는 조건을 생각하며 알고리즘을 작성해 보자.

### 3) 코딩

위에서 작성한 알고리즘을 바탕으로 파이썬 프로그램을 작성해 보자.



### 4) 테스트/디버깅

프로그램이 완성되었다면 실행시켜 보고, 여러 가지 데이터를 입력하여 잘 동작하는지 확인해 보자.

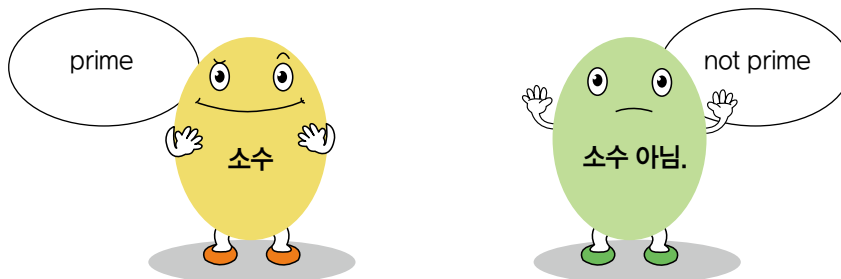
입력 데이터	예상 결과	확인 및 수정할 사항
12 30	12 0	
0 20	23 50	
4 15	3 45	
23 50	23 20	



## 2. 반복문 활용

### (1) 소수 판별

소수란, 약수가 1과 자기 자신뿐인 1보다 큰 자연수를 말한다. 어떤 수가 입력되면 그 수가 소수인지 판별하여 소수이면 “prime”을 출력하고, 소수가 아니면 “not prime”을 출력하는 프로그램을 작성해 보자.



해 보기

#### 1) 문제 분석

- 1개의 정수를 입력받아야 함. → input( ) 함수 사용
- 입력된 수가 소수인지 판별해야 함. → 판별 함수 작성
- 판별 결과를 문자로 출력해야 함. → print( ) 함수 사용

#### 2) 알고리즘 설계

소수인지 아닌지 판별하는 판별 함수의 알고리즘이다. 빈칸을 완성해 보자.

1. 입력값  $n$ 이 2보다 작으면 False 리턴
2.  $d$ 를 2부터  $n-1$ 까지 반복하여 실행  
2-1.  $n$ 을  $d$ 로 나눈 나머지가 0이면  리턴
3.  리턴

### 3) 코딩

위에서 작성한 알고리즘을 바탕으로 파이썬 프로그램을 작성해보자.

### 4) 테스트/디버깅

프로그램이 완성되었다면 실행시켜 보고, 여러 가지 데이터를 입력하여 잘 동작하는지 확인해 보자.

입력 데이터	예상 결과	확인 및 수정할 사항
1	not prime	
2	prime	
7	prime	

### 5) 개선/발전 방향

$n$ 이 아주 큰 자연수라면 소수인지 판별하는 데 오랜 시간이 걸린다. 왜 실행 시간이 오래 걸리는지 생각해 보고, 개선된 프로그램을 작성해 보자.



## (2) 주사위의 합



주사위는 각 면에 1에서 6까지 적혀 있는 정육면체이다. 우리는 두 주사위의 합이  $k$ 가 나오는 경우의 수를 모두 조사하려고 한다. 예를 들어, 주사위 두 개를 굴려 5가 나오는 경우는 1과 4, 2와 3, 3과 2, 4와 1이 있다. 우리가 주사위의 합  $k$ 를 입력하면 프로그램이 모든 경우의 수를 출력하는 프로그램을 작성해 보자.

첫 번째 출력되는 수는 첫 번째 주사위의 숫자이고, 두 번째 출력되는 수는 두 번째 주사위의 숫자이다. 첫 번째 주사위의 숫자가 작은 수에서 큰 순서로 출력한다.

해 보기

### 1) 문제 분석

문제를 분석해보고, 어떤 기능이 필요한지 정리해 보자.

### 2) 알고리즘 설계

이 문제를 해결하는 알고리즘의 일부이다. 빈칸을 완성해 보자.

1.  $k$ 를 입력받는다.

2.  $i$ 를 1부터 6까지 반복

2-1.  $j$ 를  반복

2-1-1. 만약  $i+j$ 가  $k$ 라면  $i$ 와  $j$ 를 출력

### 3) 코딩

위에서 작성한 알고리즘을 바탕으로 파이썬 프로그램을 작성해 보자.

### 4) 테스트/디버깅

프로그램이 완성되었다면 실행시켜 보고, 여러 가지 데이터를 입력하여 잘 동작하는지 확인해 보자.

입력 데이터	예상 결과	확인 및 수정할 사항
3	1 2 2 1	
5	1 4 2 3 3 2 4 1	
9	3 6 4 5 5 4 6 3	



### (3) 아메리카노

유경 선생님은 아메리카노를 매우 좋아한다. 유경 선생님의 단골 커피 전문점에서는 아메리카노를 한 잔 주문할 때마다 쿠폰을 하나 받을 수 있다. 이 쿠폰은 커피 전문점에서 정한 개수(N)가 되면 아메리카노 한 잔과 쿠폰 한 장으로 교환할 수 있다.

유경 선생님이 가진 쿠폰의 개수(K)와 커피 전문점에서 정한 아메리카노 교환에 필요한 쿠폰 개수(N)가 입력되면 선생님이 교환할 수 있는 아메리카노의 최대 잔 수를 출력하는 프로그램을 작성해 보자.



해 보기

#### 1) 문제 분석

문제를 분석해 보고, 어떤 기능이 필요한지 정리해 보자.

## 2) 알고리즘 설계

이 문제를 해결하는 알고리즘을 작성해 보자.

## 3) 코딩

위에서 작성한 알고리즘을 바탕으로 파이썬 프로그램을 작성해 보자.

## 4) 테스트/디버깅

프로그램이 완성되었다면 실행시켜 보고, 여러 가지 데이터를 입력하여 잘 동작하는지 확인해 보자.

입력 데이터	예상 결과	확인 및 수정할 사항
10 3	4	
6 10	0	
100 3	49	
2000 7	333	
1234 5	308	

### 3. 리스트 활용

#### (1) IQ150

**맞히면 아이큐 150!**

4			
6	2		
9	3	1	
19	10	7	?

오늘 신문에 맞히면 아이큐 150이라는 퍼즐이 나왔다.  
?가 있는 격자판에 들어갈 숫자가 무엇인지 맞히면 IQ 150이라는 문제이다. 문제 해결 원리는 조금만 생각해 보면 쉽게 알 수 있다. 이 상황을 좀 더 일반화하여 각 행의 제일 첫 번째 숫자들만 주어진다면 N크기의 모든 격자판 정보를 출력하는 프로그램을 작성해 보자.

첫 줄에 이 삼각격자의 세로 길이 N이 입력되며, ( $2 \leq N \leq 20$ ) 둘째 줄부터 N+1째 줄까지 (k,1)의 격자판의 정보가 입력된다. ( $1 \leq k \leq N$ )

해 보기

#### 1) 문제 분석

문제를 분석해 보고, 어떤 기능이 필요한지 정리해 본다.

입력: 여러 개의 정수 → 저장을 위해 리스트(배열)가 필요함.

처리: 삼각격자판의 규칙을 발견하고, 이를 이용해서 계산을 해야 함.

출력: 삼각격자판 → 배열의 내용을 출력하기 위해 반복문의 사용이 필요함.



## 2) 알고리즘 설계

이 문제를 해결하는 알고리즘이다. 빈칸을 완성해 보자.

1. 20×20 사이즈의 리스트를 준비한다.
2. n를 입력받는다.
3. i를 0부터 n-1까지 반복
  - 3-1. 리스트의 [i][0]에 입력값을 저장한다.
4. i를 1부터 n-1까지 반복
  - 4-1. j를 0부터 i-1까지 반복
    - 4-1-1. 리스트의 [i][j+1] =
5. i를 0부터 n-1까지 반복
  - 5-1. j를 0부터 i까지 반복
    - 5-1-1. 리스트의 [i][j]를 출력한다.
  - 5-2. 줄을 바꾼다.

## 3) 코딩

위에서 작성한 알고리즘을 바탕으로 파이썬 프로그램을 작성해 보자.

#### 4) 테스트/디버깅

프로그램이 완성되었다면 실행시켜 보고, 여러 가지 데이터를 입력하여 잘 동작하는지 확인해 보자.

입력 데이터	예상 결과	확인 및 수정할 사항
4 4 6 9 19	4 6 2 9 3 1 19 10 7 6	
5 2 12 8 15 31	2 12 10 8 -4 -14 15 7 11 25 31 16 9 -2 -27	
7 2 4 6 8 10 12 14	2 4 2 6 2 0 8 2 0 0 10 2 0 0 0 12 2 0 0 0 0 14 2 0 0 0 0 0	

테스트를 할 때 입력 데이터가 길면 불편하다. 이 때 온라인 채점 서버를 이용하면 다양한 입력 데이터를 이용하여 자동으로 프로그램의 이상유무를 점검할 수 있다. 이 문제는 코드업 온라인저지(<http://codeup.kr>) 사이트에서 문제 번호 1508으로 검색하면 테스트해 볼 수 있다.





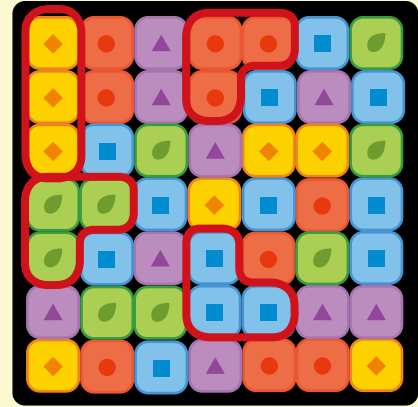
## (2) 캔디팡

최근 캔디팡이라는 스마트폰 게임이 인기를 끌고 있다. 캔디팡은 7×7 모양의 격자판에 같은 색깔이 연속 3개 이상인 부분을 찾아 터치하면 터지면서 점수를 얻는 게임이다. 이때 연속된 부분은 상, 하, 좌, 우만 판단한다.

오른쪽 캔디팡 화면에서 터치하면 터지는 영역은 총 4군데 존재한다. 캔디팡 격자 정보가 주어졌을 때 터치하면 터지는 영역의 개수를 출력하는 프로그램을 작성해 보자.

단, 입력 자료는 7×7의 격자판 색깔을 나타내는 숫자이며, 색상별 숫자는 아래와 같다.

빨강 = 1, 노랑 = 2, 파랑 = 3, 초록 = 4, 보라 = 5



해 보기

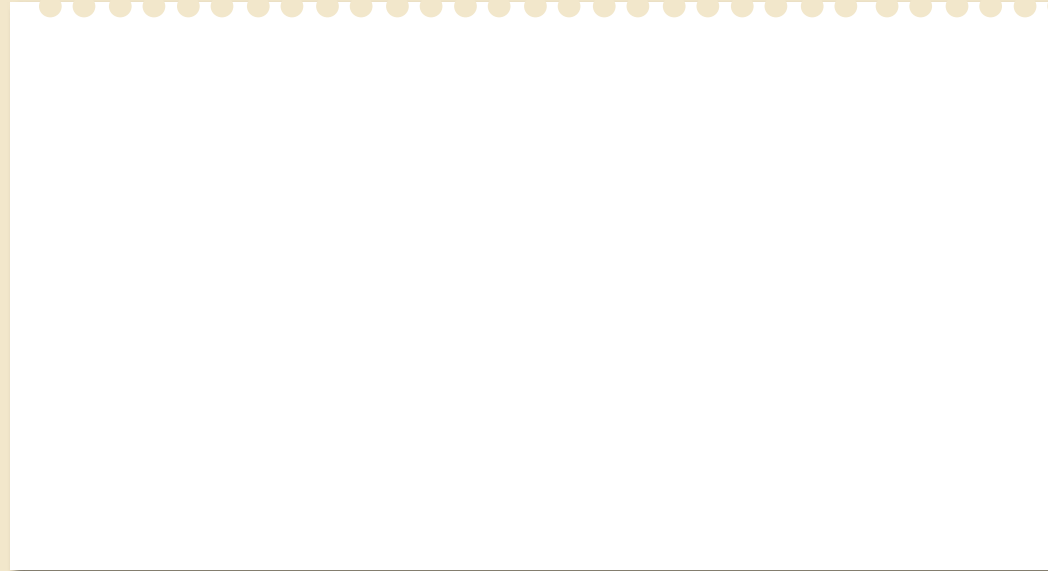
### 1) 문제 분석

문제를 분석해 보고, 어떤 기능이 필요한지 정리해 보자.



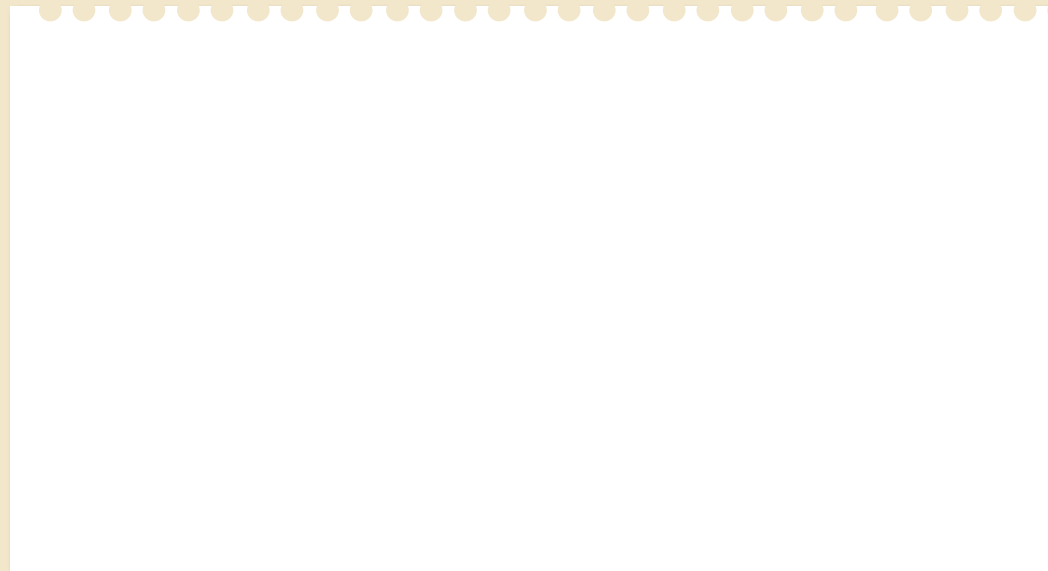
## 2) 알고리즘 설계

이 문제 풀이의 핵심은 같은 색깔이 3개 이상 연결된 구역이 몇 개인지를 판별하는 것이다. 해결을 위해 2중 for문을 이용하여 모든 격자를 차례로 탐색하면서 해당 격자 주변에 같은 색깔의 격자가 몇 개 위치했는지 세어 보는 방식을 택하면 된다. 이 문제를 해결하는 알고리즘을 작성해 보자.



## 3) 코딩

위에서 작성한 알고리즘을 바탕으로 파이썬 프로그램을 작성해 보자.



#### 4) 테스트/디버깅

프로그램이 완성되었다면 실행시켜 보고, 여러 가지 데이터를 입력하여 잘 동작하는지 확인해 보자.

입력 데이터	예상 결과	확인 및 수정할 사항
2 1 5 1 1 3 4 2 1 5 1 3 5 3 2 3 4 5 2 2 4 4 4 3 2 3 1 3 4 3 5 3 1 4 3 5 4 4 3 3 5 5 2 1 3 5 1 1 2	4	
2 1 5 1 1 3 4 2 1 5 2 3 5 3 2 2 2 2 2 2 4 2 4 3 2 3 1 3 4 3 5 2 1 4 3 5 4 4 3 3 5 5 2 1 3 5 1 1 2	1	

이 문제는 코드업 온라인저지(<http://codeup.kr>) 사이트에서 문제 번호 2605로 검색하면 테스트해 볼 수 있다.





# 알고리즘 최적화

## I 학습 목표 I

1. 넓은 범위의 소수를 효과적으로 찾아낼 수 있다.
2. 문제를 같은 해답을 갖는 보다 쉬운 문제로 변환할 수 있다.
3. 효과적인 알고리즘을 찾기 위한 창의적인 생각을 할 수 있다.

기본적인 입·출력문, 조건문, 리스트 등을 사용하면 여러 가지 문제를 해결할 수 있다. 하지만 어떤 문제들은 쉽게 알고리즘을 찾을 수 없거나 찾아낸 알고리즘을 적용하더라도 실행 시간이 너무 오래 걸려 실효성이 떨어질 수도 있다.

이런 경우 문제를 보다 쉬운 문제로 변환하여 해결하거나 보다 실행 시간이 짧아지도록 알고리즘을 개선할 필요가 있다. 이 단원에서는 문제의 변환 또는 알고리즘 최적화를 통해 문제들을 해결해 보자.

## 1. 문제 바꾸기

### (1) 하노이탑

하노이탑 문제는 A, B, C 3개의 기둥이 있고, A 기둥에는 크기가 다른 여러 개의 원판이 크기 순으로 끼워져 있을 때 다음 규칙을 지키면서 A 기둥의 원판을 모두 C 기둥으로 옮기는 것이다.

- 규칙 1** 제일 위의 원판만 이동할 수 있다.
- 규칙 2** 한 번에 하나의 원판만 이동할 수 있다.
- 규칙 3** 이동하는 원판보다 작은 원판 위로는 이동할 수 없다.

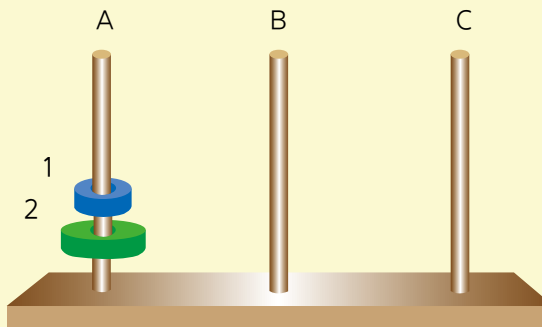
원판의 개수( $1 < n < 10$ )가 주어질 때 원판의 최단 이동 경로를 출력하는 프로그램을 작성해 보자. 출력 형식은 예시를 참고한다.

예시)  $n=2$ 일 때

Disk 1: A to B

Disk 2: A to C

Disk 1: B to C



## 1) 문제 분석

하노이탑 문제는 굉장히 유명한 문제인데, 원판의 개수가 적다면 손으로도 어렵지 않게 풀 수 있지만 원판의 개수가 10개라면 최소한 1023번의 이동을 해야 하므로 틀리기 쉬운 문제이다. 이 문제를 쉽게 풀기 위해 아래와 같이 문제를 분석하였다. 빈칸을 완성해 보자.

- 1개의 정수를 입력받아야 함. → `input( )` 함수 사용
- 결과를 형식에 맞추어 출력해야 함. → `print("{ }".format( ))` 함수 사용
- 문제의 규칙 찾기
  - 만약  $n$ 이 10이라면 문제를 다음과 같이 변형 가능함.
    - ① 9개의 원판을 A에서 B로 옮기는 문제
    - ② 10번 원판을 A에서 C로 옮김.
    - ③ 9개의 원판을 B에서 C로 옮기는 문제
  - 만약  $n$ 이 9라면 문제를 다음과 같이 변형 가능함.

- 만약  $n$ 이 2라면 다음과 같이 3번 만에 풀 수 있음.
  - ① 1번 원판을 A에서 B로 옮김(1개의 원판을 A에서 B로 옮기는 문제 해결).
  - ② 2번 원판을 A에서 C로 옮김.
  - ③ 1번 원판을 B에서 C로 옮김(1개의 원판을 B에서 C로 옮기는 문제 해결).

## 2) 알고리즘 설계

아래의 내용은 분석 단계에서 찾은 규칙을 이용하여 원판의 개수가  $n$ 인 하노이탑 문제를 해결할 수 있는 함수  $f(n)$ 을 정의한 것이다. 빈칸을 채우시오.

함수정의  $f(n, \text{출발지}, \text{경유지}, \text{목적지})$

1. 만약  $n$ 이 1이라면

1-1. 'Disk 1: 출발지 to 목적지'를 출력하고 리턴

2. 문제를 규칙에 따라 3개의 작은 문제로 분리한다.

2-1.  $f(n-1, \text{출발지}, \text{목적지}, \text{경유지})$  호출

2-2.

2-3.

## 3) 코딩

위에서 작성한 알고리즘을 바탕으로 파이썬 프로그램을 완성해 보자.

```
def f(n,a,b,c):
```

```
n=int(input( ))
```

```
f(n, 'A', 'B', 'C')
```

#### 4) 테스트/디버깅

프로그램이 완성되었다면 실행시켜 보고, 여러 가지 데이터를 입력하여 잘 동작하는지 확인해 보자.

입력 데이터	예상 결과	확인 및 수정할 사항
3	Disk 1: A to C Disk 2: A to B Disk 1: C to B Disk 3: A to C Disk 1: B to A Disk 2: B to C Disk 1: A to C	
4	Disk 1: A to B Disk 2: A to C Disk 1: B to C Disk 3: A to B Disk 1: C to A Disk 2: C to B Disk 1: A to B Disk 4: A to C Disk 1: B to C Disk 2: B to A Disk 1: C to A Disk 3: B to C Disk 1: A to B Disk 2: A to C Disk 1: B to C	

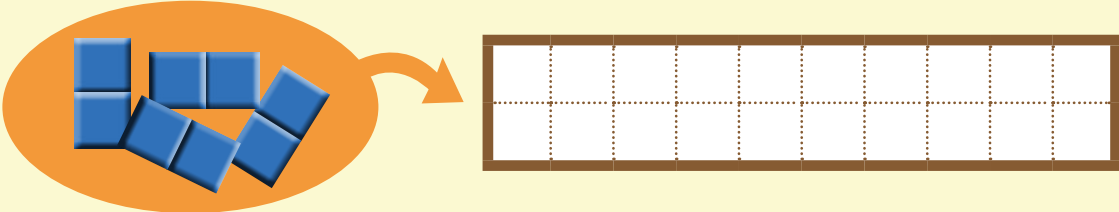
대부분의 경우 어려워 보이는 문제를 규칙에 따라 여러 개의 쉬운 문제로 나눌 수 있다면 재귀함수를 사용하여 쉽게 코딩할 수 있다.





## (2) 블록 채우기

2×1의 직사각형 블록으로 2×n 크기의 직사각형 블록을 완전히 채우려고 한다.



$n(1 \leq n \leq 100,000)$ 을 입력받았을 때 가능한 방법의 수를 구하는 프로그램을 작성해 보자. 단, 2×1 직사각형 블록은 무한정 있다고 가정하고, 입력값에 따라 결과값이 커질 수 있으므로 결과값을 100,000,007로 나눈 나머지를 출력한다.

해 보기

### 1) 문제 분석

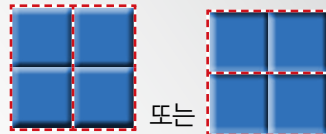
이 문제를 쉽게 풀기 위해 아래와 같이 문제를 분석하였다. 빈칸을 채우시오.

- 1개의 정수를 입력받아야 함. → `input( )` 함수 사용
- 1개의 정수를 출력해야 함. → `print( )` 함수 사용
- 문제의 규칙 찾기

– 만약  $n$ 이 1이라면 직사각형을 채울 수 있는 방법은 다음 그림처럼 세로로 놓는 1가지이다.



–  $n$ 이 2라면 직사각형을 세로로 나란히 놓거나 가로로 나란히 놓는 2가지 방법이 있다.



–  $n$ 이 3이라면 첫 칸에 세로로 놓으면 남은 칸이 2칸이므로  $n$ 이 2인 문제와 결과가 같고, 첫 칸에 가로로 놓으면 남은 칸이 2칸이므로  $n$ 이 1인 문제와 결과가 같다. 즉,  $n$ 이 3일 때는  $n$ 이 1일 때 결과와  $n$ 이 2일 때 결과의 합과 같으므로 3가지 방법이 있다.



-  $n$ 이 4라면

- 일반화하면  $n=N$ 일 때 문제를 해결하는 함수  $f(N)$ 은 다음과 같이 정의된다.

## 2) 알고리즘 설계

분석 단계에서 찾은 규칙을 이용하여 블록 채우기 문제를 우리가 이미 알고 있는 유명한 문제로 변형하였다. 그리고 문제를 풀 수 있는 재귀 함수를 다음과 같이 설계하였다. 빈칸을 완성시켜 보자.

함수 정의  $f(n)$

1. 만약  $n$ 이 1 또는 2라면

1-1.  $n$ 을 리턴

2. 문제를 규칙에 따라 3개의 작은 문제로 분리한다.

2-1.  를 리턴

## 3) 코딩

위에서 작성한 알고리즘을 바탕으로 파이썬 프로그램을 완성해 보자.

```
def f(n,a,b,c):
```

```
n=int(input( ))
```

```
print(f(n))
```

#### 4) 테스트/디버깅

프로그램이 완성되었다면 실행시켜 보고, 여러 가지 데이터를 입력하여 잘 동작하는지 확인해 보자.

입력 데이터	예상 결과	확인 및 수정할 사항
10	89	
40	65580134	
10000	67897762	

#### 5) 개선/발전 방향

① 지금 만든 프로그램은 어떤 문제점들이 있는지 정리해 보자.

② 큰 입력값( $n=100000$ )도 빠른 시간 내에 처리할 수 있도록 프로그램을 개선해 보자.

## 2. 알고리즘 최적화

일반적인 방법으로 풀면 실행 시간이 너무 오래 걸리는 문제들이다. 문제를 잘 읽고 발상의 전환이나 효율적인 알고리즘을 찾는 노력이 필요하다.

### (1) 구간에서의 소수 찾기

지현이는 소수 판별 알고리즘을 활용할 곳이 없을까 고민하다가 새로운 소수 문제를 발견했다. 그 문제는 특정 구간에 있는 소수를 모두 찾아내는 문제였다. 2개의 정수  $a, b$ 를 입력받아 두 수 사이에 포함된 소수들의 합을 출력하는 프로그램을 완성해 보자.

단, 시작값( $a$ )과 마지막 값( $b$ )은 공백으로 분리되어 입력되며, 그 범위는  $2 \leq a < b \leq 1,000,000$ 이다.

해 보기

#### 1) 문제 분석

이 문제를 쉽게 풀기 위해 아래와 같이 문제를 분석하였다. 빈칸을 완성해 보자.

- 2개의 정수를 입력받아야 함. →  사용
- 여러 개의 소수를 한 줄에 출력해야 함. → `print(값, end=" ")` 사용
- 처리: 소수판별 알고리즘 필요함.

#### 2) 알고리즘 설계

소수 판별 함수의 알고리즘을 작성하시오.

### 3) 코딩

위에서 작성한 알고리즘을 바탕으로 파이썬 프로그램을 완성해 보자.

### 4) 테스트/디버깅

프로그램이 완성되었다면 실행시켜 보고, 여러 가지 데이터를 입력하여 잘 동작하는지 확인해 보자.

입력 데이터	예상 결과	확인 및 수정할 사항
18001 20000	3763854	
1 20000	21171191	
98001 100000	17229464	
998001 1000000	38857250	
1 1000000	50399398	

### 5) 개선/발전 방향

① 다른 여러 가지 입력 데이터를 넣어보고, 지금 만든 프로그램은 어떤 문제점들이 있는지 정리해 보자.

- 숫자가 커질수록
- 범위가 넓어질수록
- 숫자와 범위가 동시에 증가하면

② 왜 이런 문제가 생기는지 원인을 생각해보고 정리해 보자.

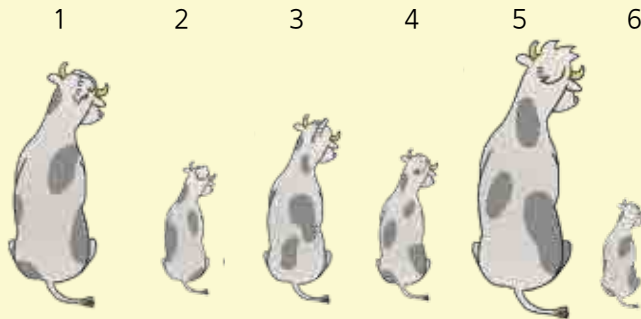
③ 문제 해결 방안에 대해 생각해보고, 다른 프로그램을 작성해 보자.



## (2) 소들의 헤어스타일

농부 존에게는 N마리의 소들이 있다. 소들은 헤어스타일에 예민하기 때문에, 다른 소들의 헤어스타일을 보고 싶어 한다. i번째 소의 키가  $h_i$ 이고, 모든 소들은 일렬로 서 있고 오른쪽으로만 볼 수 있다. i번째 소가 볼 수 있는 다른 소의 헤어스타일은  $i+1, i+2, \dots$  이다. 그런데 자기보다 키가 큰 소가 있으면 그 다음에 있는 모든 소들의 헤어스타일은 보지 못한다.

예)  $N=6, H = \{ 10, 3, 7, 4, 12, 2 \}$ 인 경우



1번 소는 2, 3, 4번 소의 헤어스타일을 확인할 수 있다.

2번 소는 다른 소들의 헤어스타일을 확인할 수 없다.

3번 소는 4번 소의 헤어스타일을 확인할 수 있다.

4번 소는 다른 소의 헤어스타일을 확인할 수 없다.

5번 소는 6번 소의 헤어스타일을 확인할 수 있다.

6번 소는 마지막이므로 다른 소의 헤어스타일을 확인할 수 없다.

따라서 헤어스타일을 확인할 수 있는 총 수는  $3 + 0 + 1 + 0 + 1 + 0 = 5$ 이다.

입력값으로 소의 수  $N(1 \leq N \leq 80,000)$ 과 각 소들의 키를 입력받아 각각의 소들이 헤어스타일을 확인할 수 있는 소들의 수의 합을 출력하는 프로그램을 작성해 보자.

해 보기

### 1) 문제 분석

- 1개의 정수를 입력받아야 함. → `input()` 함수 사용
- 1개의 정수를 출력함.
- 처리: 반복문을 이용해 각 소들이 볼 수 있는 소들의 수를 찾아 누적하기

## 2) 알고리즘 설계

각각의 소가 자신의 오른쪽에 자신과 키가 같거나 큰 소가 나타나기 전까지 헤어스타일을 볼 수 있으므로 2중 반복문을 사용하도록 해결 알고리즘을 설계해 보자.



## 3) 코딩

위에서 작성한 알고리즘을 바탕으로 파이썬 프로그램을 완성해 보자.



## 4) 테스트/디버깅

① 간단한 데이터 입력을 통한 테스트

입력 데이터	예상 결과	확인 및 수정할 사항
4 4 3 2 1	6	
6 10 3 7 4 12 2	5	



## ② 온라인 채점 서버를 통한 테스트

간단한 데이터 입력을 통해 올바른 출력 결과가 나왔다고 프로그램이 잘 동작한다고 안심할 수 있을까? 이 문제의 경우 N의 최댓값이 80,000인데, 80,000개의 데이터를 직접 넣기는 매우 불편하다. 여러분의 프로그램을 제대로 테스트하기 위해서는 온라인 채점 서버를 이용해야 한다. 코드업 온라인 저지(<http://codeup.kr>)에 접속해서 3130번 문제를 검색한 다음 여러분의 프로그램 코드를 제출하고 결과를 확인해 보자.

### 채점 결과의 이해

#### ㉠ 채점 결과가 **정확한 풀이** 일 경우

여러분은 프로그래밍을 통한 문제 풀이의 고수이다. 정보 올림피아드나 코드포스 또는 구글 코드잼과 같은 프로그래밍 대회에 꼭 도전해 보자.

#### ㉡ 채점 결과가 **잘못된 풀이** 일 경우

프로그램 실행 시간이 너무 긴 경우이므로 알고리즘의 최적화가 필요하다.

#### ㉢ 채점 결과가 **시간 초과** 일 경우

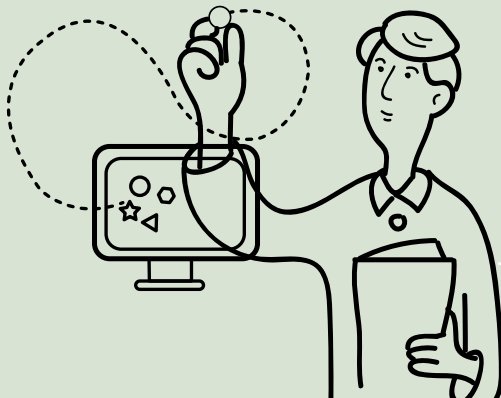
알고리즘이 올바른 결과를 내지 못했으므로 설계 단계부터 다시 고민해 보자.

## 5) 개선/발전 방향

① 지금 만든 프로그램은 어떤 문제점들이 있는지 정리해 보자.

② 많은 수의 데이터도 빠른 시간 내에 처리할 수 있도록 프로그램을 개선해 보자.

- ① 알고리즘은 특정한 문제를 해결하기 위한 논리적인 방법이나 절차를 의미한다. 프로그래밍 언어로 문제를 해결할 때는 문제를 해결하는 적절한 알고리즘을 설계해야 하며, 제작한 알고리즘을 컴퓨터가 이해할 수 있도록 프로그래밍 언어로 표현하는 과정을 거친다.
- ② 알고리즘을 표현하기 위한 방법은 자연어, 순서도, 의사 코드, 프로그래밍 언어 등이 있다. 우리가 일반적으로 사용하는 언어로 표현하는 방식이 자연어이고 약속된 기호로 표현하는 방식이 순서도이다. 의사 코드는 프로그래밍 언어와 비슷한 형태의 알아보기 쉬운 형태로 알고리즘을 표현하는 방식이고 프로그래밍 언어는 특정한 프로그래밍 언어를 사용하여 알고리즘을 표현한다.
- ③ 문제를 해결하기 위한 알고리즘은 여러 가지가 있을 수 있지만 가장 효과적이고 효율적인 알고리즘을 찾는 것이 필요하다. 알고리즘의 성능을 평가하는 기준으로는 실행 시간과 메모리 사용량이 있다. 문제를 해결하는 데 더 짧은 시간이 걸리는 것과 더 적은 메모리 사용량을 차지하는 알고리즘이 더욱 좋은 알고리즘이라고 말할 수 있다.
- ④ 프로그래밍에서 문제 해결 과정은 프로그래밍의 기본 개념인 순차, 반복, 조건을 적절하게 조합하여 진행된다.
- ⑤ 문제 해결 과정에서 여러 자료가 순차적으로 활용된다면 배열 구조로 자료를 저장한 다음 편리하게 불러와 활용할 수 있다.
- ⑥ 복잡한 문제를 해결할 때는 주어진 문제를 보다 간단한 부분 문제로 나누어 풀거나 알고리즘을 최적화하여 해결하여야 한다.
- ⑦ 프로그래밍 문제는 알고리즘이 정확하게 문제를 해결하는지, 얼마나 빠르고 효율적으로 해결하는지 판단하는 시스템이 있다면 보다 편리하게 해결할 수 있다. 프로그래밍 학습을 위해서 다양한 온라인 채점 서버가 활용된다.



# CHAPTER



# GUI 프로그래밍

## 01 계산기 프로젝트

## 02 자판기 프로젝트

프로그래밍 언어의 사용 방법을 익히고 논리적인 문제 해결 과정을 설계할 수 있는 능력을 갖춘 사람은 실생활에서 발생하는 다양한 문제 상황을 소프트웨어를 제작하는 것으로 해결할 수 있다. 실생활 문제를 해결하기 위해서는 실생활에서 일어나는 복잡한 문제에 대해 인식하고, 이를 소프트웨어로 제작하기 위한 방법을 고안하고, 실제 실행되는 소프트웨어로 구현하는 과정을 거친다. 이 단원에서는 파이썬 프로그래밍을 통해 실생활 문제를 해결하는 과정을 이해하고, 그래픽 사용 환경을 제공하는 프로그램을 만드는 과정을 살펴보기로 한다.

# 01



## 계산기 프로젝트

### I 학습 목표 I

1. Qt 디자인너를 이용하여 GUI 화면을 만들 수 있다.
2. 이벤트 처리를 할 수 있다.
3. 실생활 문제를 해결하는 GUI 프로그램을 작성할 수 있다.



### 1. GUI 프로그래밍

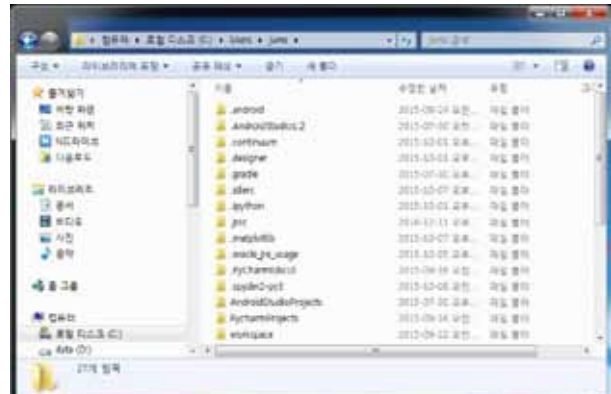
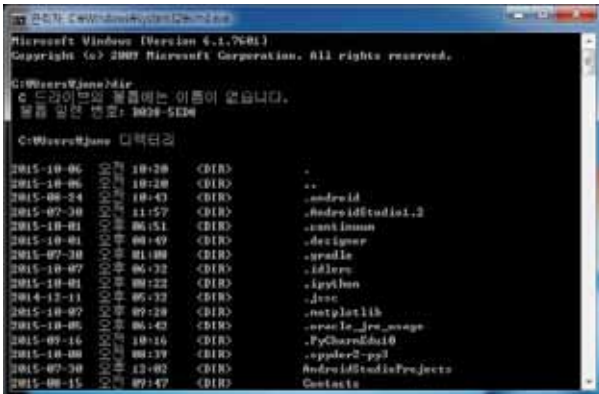
**CUI(Character User Interface):** 문자 기반으로 사용자와 소통하는 방식으로 주로 명령어 입력을 통해 컴퓨터를 제어한다.

**GUI(Graphical User Interface):** 그래픽 요소를 기반으로 사용자와 소통하는 방식



앞에서 우리는 간단한 형태의 프로그램을 제작해 보았다. 우리가 만든 프로그램들은 숫자 또는 문자들을 키보드를 통해 입력받거나 그 결과를 문자의 형태로 모니터에 표시하는 방식이었다. 즉, 사용자와 정보를 주고받는 방식이 텍스트를 기반으로 하고 있다.

하지만 요즘 사용되는 어플리케이션 프로그램의 대부분은 GUI방식으로 되어 있다. GUI 방식은 그래픽 요소를 사용하여 화면을 아름답고 직관적으로 보여주기 때문에 명령어를 입력하는 방식보다 쉽게 사용할 수 있다.



[그림 III-1] CUI와 GUI 소통 방식



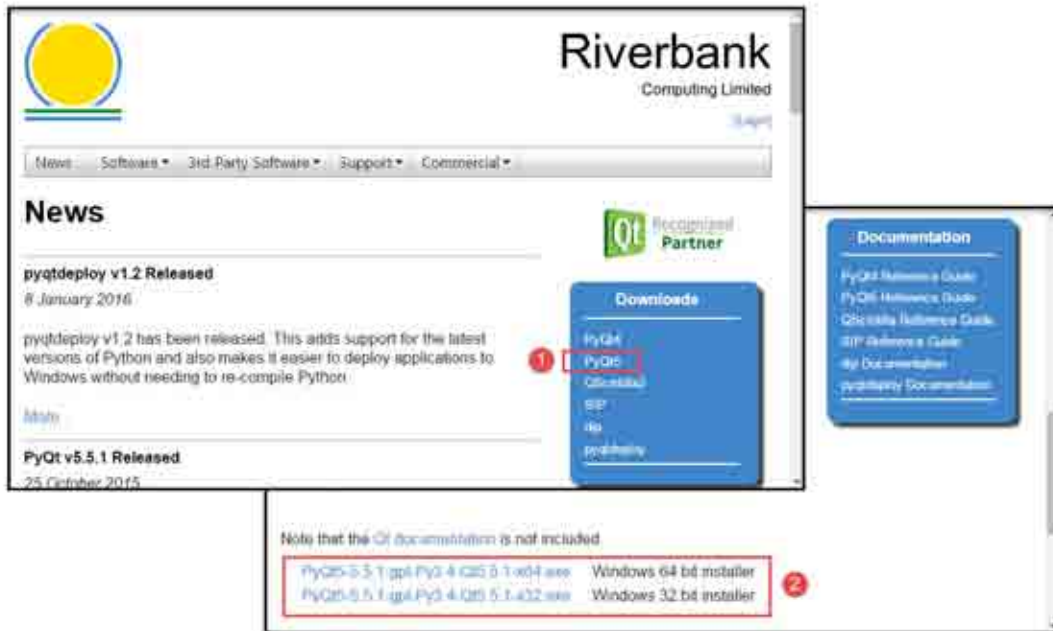
[그림 III-2] GUI 디자인의 사례

## (1) PyQt

### 1) PyQt 설치

파이썬을 이용하여 GUI 프로그래밍을 하려면 GUI 관련 라이브러리가 필요하다. 파이썬에는 기본적으로 GUI를 구현할 수 있는 Tkinter 모듈이 포함되어 있어 이를 이용할 수도 있다. 하지만 우리는 PyQt라는 라이브러리를 설치하여 사용한다. PyQt는 Tkinter에 비해 화면이 더 예쁘고, 화면 디자인을 쉽게 할 수 있는 Qt 디자이너라는 프로그램을 포함하고 있다.

PyQt는 리버뱅크 홈페이지(<https://riverbankcomputing.com/news>)에서 다운로드할 수 있다. 파이썬을 설치할 때처럼 자신의 운영 체제와 파이썬 버전에 맞는 파일을 다운로드하여 설치한다.



[그림 III -3] PyQt 프로그램 다운로드 화면



## 2) 간단한 GUI 프로그래밍

PyQt 설치가 잘 되었다면 먼저 간단한 GUI 프로그램을 작성해 보자. 다음 프로그램은 화면에 윈도우를 띄우고, 윈도우 안에 “Say ‘Hello, World’”라는 버튼을 보여주는 프로그램이다.

```
# 라이브러리 가져오기
import sys
from PyQt4 import Qt

# 애플리케이션 프로그램 생성
myApp = Qt.QApplication(sys.argv)

# 버튼 생성
button = Qt.QPushButton("Say 'Hello world!'",None)
button.show()    # 버튼이 화면에 보이도록 설정
myApp.exec_()    # GUI 프로그램 실행
```

입력이 잘 되었다면 소스 코드를 저장하고 실행시켜 보자. 아래 그림과 같은 결과가 화면에 표시되었다면 첫 번째 GUI 프로그램을 성공적으로 작성한 것이다.



[그림 III-4] 첫 번째 GUI 프로그램 실행 화면

## 3) GUI 프로그래밍 단계

처음 만든 GUI 프로그램을 실행시키면 윈도우 창과 버튼이 표시되는 것을 보았다. 우리가 버튼을 클릭하면 버튼이 클릭되는 애니메이션을 보여주는 하지만 아무런 일도 일어나지 않는다. 이는 버튼이 눌리졌을 때 실행해야 할 명령을 내리지 않아서이다.

이처럼 GUI 프로그램을 작성할 때는 화면을 보여주는 코드만 작성하는 것이 아니라 화면에서 발생하는 이벤트를 처리하는 코드도 작성해야 한다. 즉, GUI 프로그래밍은 2개의 단계로 이루어진다.

### ① 화면 디자인

사용자의 눈에 보이는 화면을 만드는 것으로 기본적으로 윈도우에 버튼, 텍스트 상자, 체크박스, 레이블 등 화면을 구성하는 요소를 적절히 배치하고, 색상이나 크기 등의 속성을 설정하는 작업이다.

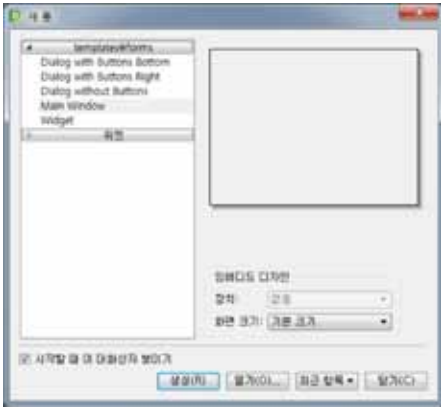
### ② 이벤트 처리

GUI 화면에 어떤 이벤트가 발생했을 때 그것을 처리하는 코드를 작성하는 것이다. 이벤트는 사용자가 GUI 화면에 특정한 조작을 할 때나 프로그램 실행 중 특정한 상황이 발생하는 것을 의미한다. 예를 들어 사용자가 화면에 있는 [종료] 버튼을 클릭한다면 버튼 클릭 이벤트가 발생하게 되고, 버튼 클릭 이벤트가 발생했을 때 처리할 내용을 코드로 작성하는 것이 이벤트 처리이다.

화면 디자인과 이벤트 처리 모두 프로그래머가 신경을 써야 하는 부분이다. 개발해야 하는 프로그램이 복잡한 경우 이 과정을 분담해서 진행할 수도 있다.







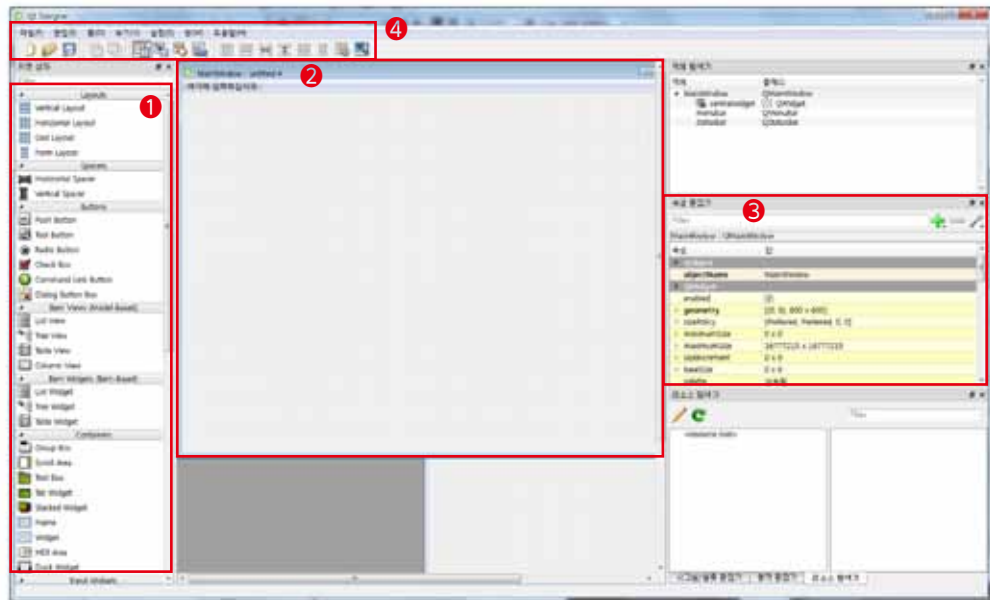
[그림 III-5] 윈도우 창의 종류 선택

## (2) 화면 디자인

파이썬은 다른 프로그래밍 언어에 비해 비교적 짧은 코드로 GUI 프로그래밍이 가능하다. 그러나 윈도우에 포함된 컨트롤의 숫자가 늘어나거나 윈도우의 크기, 표시되는 문자열의 글꼴 및 크기 등 세세한 속성을 지정해서 윈도우를 표시하려고 하면 코드의 길이가 길어지고, 초보자의 경우 쉽게 화면을 작성할 수 없게 된다. 따라서 본 교재에서는 보다 간편한 Qt 디자인너를 사용하여 화면을 작성하는 방법을 알아본다.

[시작]-[모든 프로그램]-[PyQt GPL v4.11.4 for Python v3.4]-[Designer]를 실행시키면 Qt 디자이너 프로그램이 실행된다.

프로그램이 시작되면 윈도우 창의 종류를 선택할 수 있는데, 일반적인 어플리케이션 프로그램은 'Main Window'를 선택하고, [생성] 버튼을 누른다.



[그림 III-6] Qt 디자인너 화면 구성

위젯의 name 속성은 코드에서 위젯에 접근할 때 필요하므로 기억하기 좋은 이름을 지정하는 것이 좋다.



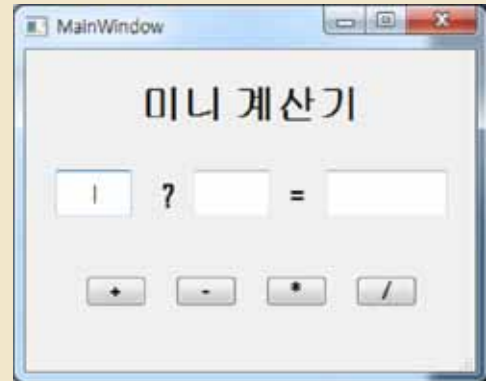
### GUI 화면을 디자인하는 방법

- ① 영역의 위젯 중에서 화면구성에 필요한 위젯을 선택한다.
- ② 영역의 윈도우 창 부분에 드래그 앤 드롭한다.
- 마우스를 이용하여 위젯의 위치나 크기를 조절한다.
- 필요한 경우 ③ 영역의 속성 편집기를 이용하여 위젯의 속성을 설정한다.
- 화면 작성이 끝나면 ④ 영역의 파일 메뉴를 이용하여 화면 파일의 이름을 지정하고 저장한다.



- 아주 간단한 계산기 프로그램을 한 번 만들어 보자. 이 계산기 프로그램은 두 수의 사칙연산을 할 수 있는 프로그램이다.

- ① Qt 디자이너를 이용하여 오른쪽 그림과 같은 화면을 만들고, 'calc.ui'라는 이름으로 저장한다. Qt 디자이너를 이용하여 만든 화면은 'xml' 형식의 파일로 'ui'라는 확장자로 저장되며, 아직 실행할 수 있는 상태는 아니다. 프로그램 실행을 위해 다음 코드를 입력해 보자.



[그림 III-7] 미니 계산기 화면

```
import sys
from PyQt4 import QtCore, QtGui, uic

form_class = uic.loadUiType("calc.ui")[0] # UI 파일읽어오기

class MyWindowClass(QtGui.QMainWindow, form_class):
    def __init__(self, parent=None):
        QtGui.QMainWindow.__init__(self, parent)
        self.setupUi(self)

app = QtGui.QApplication(sys.argv)
myWindow = MyWindowClass(None)
myWindow.show()
app.exec_()
```

- ② 코드를 입력하고 실행해 보자. 화면이 정상적으로 나타나는가?

### (3) 이벤트 처리

GUI 화면을 구성하면 겉으로 보기엔 프로그램이 완성된 것처럼 보이지만 사용자가 프로그램을 사용하기 위해 조작을 하더라도 프로그램이 동작하지 않는다. 이처럼 사용자가 프로그램에 있는 버튼을 누르거나 문자를 입력하는 등의 행위를 하면 프로그램에서는 이를 알아내고, 사용자의 의도에 맞도록 동작하는 코드를 입력해 주어야 한다.



파이썬에서는 프로그램이 감지한 어떤 상황(예를 들어 사용자의 프로그램 조작)으로 발생된 신호를 시그널, 이벤트가 발생할 때 실행되는 코드를 슬롯이라고 한다. 즉, 이벤트 처리를 위해서는 슬롯을 작성하고, 시그널과 연결하는 작업이 필요하다.

슬롯을 작성하기 전에 계산기의 알고리즘을 설명해 보자.

계산기의 알고리즘을 작성해 보자.




앞에서는 계산기의 외형을 설계하였으니 이제 계산기 기능을 설계해 보자.

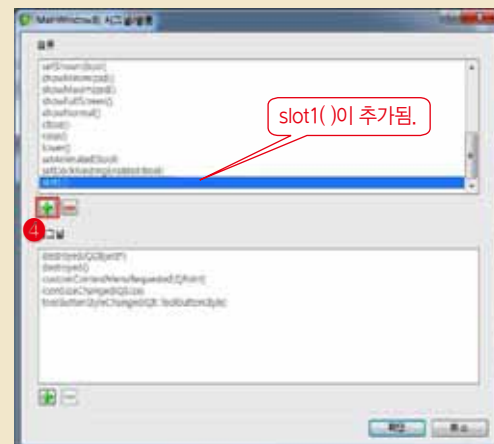
- ① Qt 디자이너에서 'dac.ui' 파일을 열고, [F4] 또는  아이콘을 눌러 시그널/슬롯 편집 모드로 변경한다.
- ②  버튼을 누른 상태로 버튼 바깥쪽으로 드래그 앤 드롭하면 연결 설정 대화상자가 나타난다.



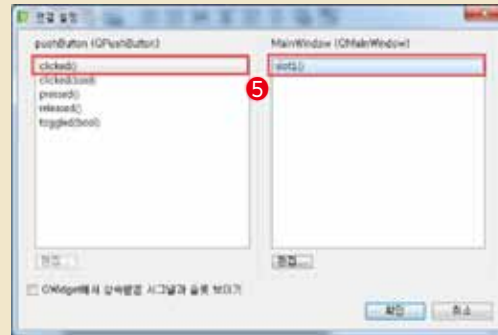
- ③ 오른쪽 편집 버튼을 눌러 시그널/슬롯 대화상자를 표시한다.



- ④ 슬롯 아래의  버튼을 눌러 슬롯을 추가한 다음 [확인] 버튼을 누른다.



- ⑤ 연결/설정 창의 왼쪽 창에서 clicked( )을 선택하고, 오른쪽 창에서 slot1( )을 선택한 다음 [확인] 버튼을 클릭한다. clicked( )는 버튼을 클릭하면 발생하는 시그널이고, 이 시그널을 처리할 슬롯이 slot1( )이라는 의미이다.



- ⑥ 코드 창을 열어 slot1( )을 정의한다. slot1( )은 [+] 버튼이 클릭되는 이벤트가 발생했을 때 처리하는 함수이다.

```
import sys
from PyQt4 import QtCore, QtGui, uic

form_class = uic.loadUiType("calc.ui")[0] # UI 파일 읽어오기

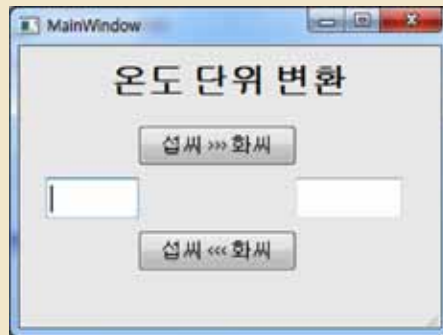
class MyWindowClass(QtGui.QMainWindow, form_class):
    def __init__(self, parent=None):
        QtGui.QMainWindow.__init__(self, parent)
        self.setupUi(self)

    def slot1(self): # slot1() 정의
        result = int(self.lineEdit.text())+int(self.lineEdit_2.text())
        self.lineEdit_3.setText(str(result))

app = QtGui.QApplication(sys.argv)
myWindow = MyWindowClass(None)
myWindow.show()
app.exec_()
```

🔴 나머지 계산 버튼을 눌렀을 때도 계산이 되도록 프로그램을 완성해 보자.

- 아래 그림과 같이 섭씨온도와 화씨온도를 서로 변환하는 프로그램을 만들어 보자.



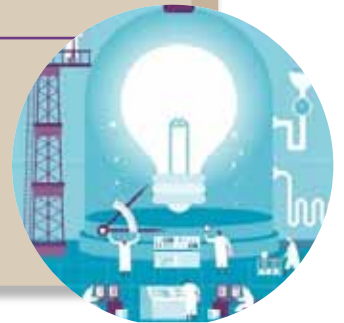
- 화씨온도 = ( 섭씨온도  $\times$  1.8 ) + 32
- 섭씨온도 = ( 화씨온도 - 32 ) / 1.8



# 자판기 프로젝트

## ! 학습 목표 !

1. 요구 사항을 분석하고, 기능을 정의할 수 있다.
2. Qt 디자이너를 이용하여 GUI 화면을 만들 수 있다.
3. 이벤트 처리를 위한 슬롯을 정의할 수 있다.
4. 실생활 문제를 해결하는 GUI 프로그램을 작성할 수 있다.



## 1. 자판기 만들기

다음과 같은 문제 상황을 가정해 보자.



주미가 다니는 학교 매점의 커피 자판기는 커피 한 잔에 200원을 받는다. 그런데, 이 자판기 동작 프로그램이 200원씩 밖에 인식을 못해서 두 잔을 뽑으려면 200원을 넣고 한 잔을 뽑고, 또 다시 200원을 넣고 한 잔을 뽑아야 한다.

자판기 사용이 불편하다고 생각한 주미는 자판기를 교체해 달라고 건의하려고 했다. 그런데, 생각해 보니 자판기의 프로그램만 교체하면 문제가 해결될 수 있겠다는 생각이 들었고, 새로운 자판기 프로그램을 만들어 보기로 했다.

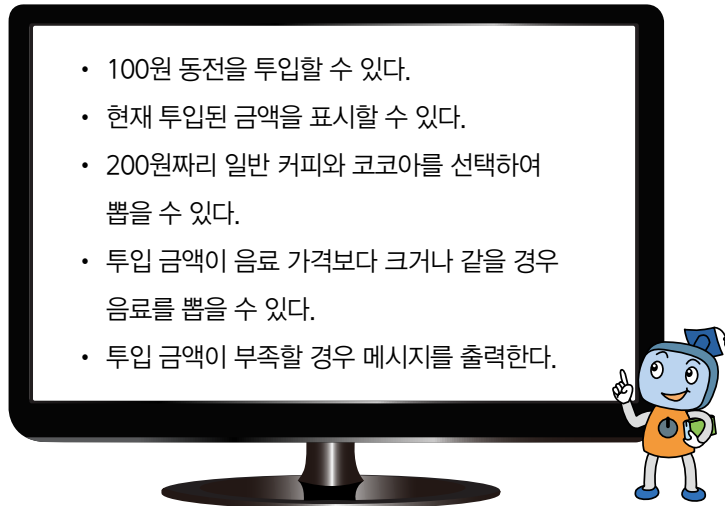


이러한 문제를 해결하기 위하여 다음과 같은 자판기 프로그램을 만들어 보자. 새로운 프로그램은 투입된 금액을 화면에 표시하고, 여러 가지 종류의 커피 또는 코코아를 선택할 수 있다. 또 투입된 금액이 음료의 가격보다 적으면 돈을 더 투입하라고 하며, 그렇지 않은 경우 음료를 뽑거나 잔돈 반환을 결정하도록 한다.

소프트웨어 제작 과정을 바탕으로 주미네 학교 매점의 커피 자판기가 효율적으로 동작할 수 있도록 새로운 자판기 프로그램을 작성해 보자.

## (1) 요구 사항 분석

주미는 프로그램을 작성하기 전에 만들려고 하는 자판기 프로그램에 요구되는 기능들을 분석하고 정리할 필요가 있다. 주미는 기존 자판기 프로그램을 분석하고 새로운 기능을 추가하여 아래와 같이 프로그램의 기능을 정리하였다.



해 보기

- 자판기 프로그램에 더 추가할 기능이 있는지 생각해 보자.

A large empty rectangular box with a scalloped top edge, intended for writing additional requirements.



## (2) 프로그램 설계

실제 자판기를 만들기 전에 자판기 프로그램을 만들어 정확하게 동작하는지 확인해 보자. GUI 프로그램을 작성할 때 크게 화면 부분과 이벤트 처리 부분으로 구분되므로 화면 설계와 알고리즘 설계를 나누어서 진행해 보자.

### 1) 화면 설계

분석 단계에서 정리한 프로그램은 기능을 수행할 수 있도록 화면을 구성하여야 한다. 화면 설계를 할 때에는 기능 수행에 필요한 위젯이 무엇이고, 몇 개가 필요한지를 판단하고, 필요한 위젯을 어떤 형태로 배치하면 사용자의 편의성과 심미적으로 좋을지를 고민한다.

자판기 프로그램의 화면에는 현재 투입 금액을 표시할 수 있는 위젯이 필요하고, 돈을 투입하거나 잔돈을 반환 받으려 할 때, 음료를 선택할 때 사용할 버튼들이 필요하다.

해 보기

- 예시 화면을 참고하여 자판기 프로그램의 화면을 설계해 보자. 화면을 구성할 때 기능 수행에 필요한 위젯이 누락되지 않도록 주의한다.

예시 화면



## 2) 알고리즘 설계

자판기 프로그램이 처리해야 하는 중요한 기능은 다음과 같다.

• 100원 버튼을 눌러 100원을 투입하였을 때

• 일반 커피 버튼을 눌렀을 때

• 코코아 버튼을 눌렀을 때

• 잔돈 반환 버튼을 눌렀을 때



• 100원 버튼을 눌렀을 때

1. 현재 투입금액에 100원을 더한다.
2. 현재 투입금액을 화면에 표시한다.

• 일반 커피 버튼을 눌렀을 때

1. 현재 투입 금액이 200원 이상이면
  - ① 현재 투입 금액에서 200원을 뺀다.
  - ② 현재 투입 금액을 화면에 표시한다.
  - ③ “커피 준비 완료” 메시지를 출력한다.
2. 그렇지 않으면  
“금액 부족” 메시지를 출력한다.

해 보기

● 코코아 버튼을 눌렀을 때 처리할 내용을 작성해 보자.

• 코코아 버튼을 눌렀을 때

● 잔돈 반환 버튼을 눌렀을 때 처리할 내용을 작성해 보자.

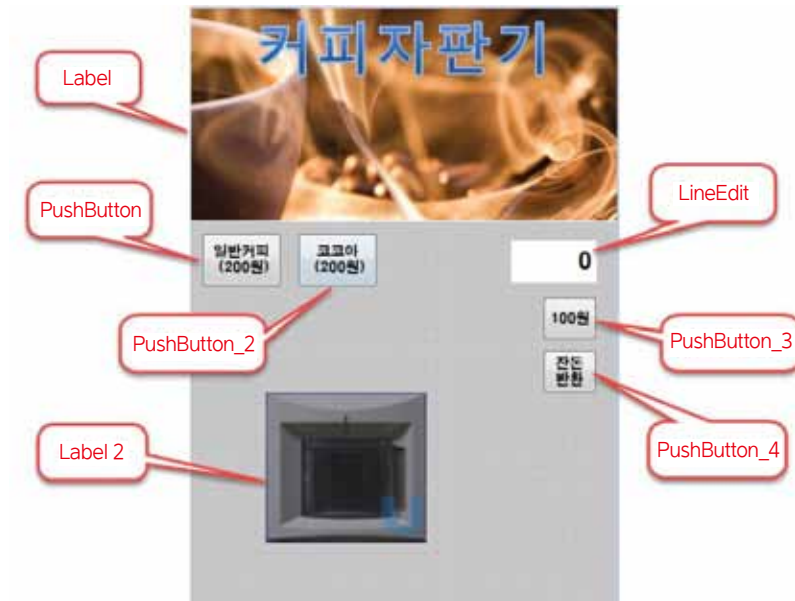
• 잔돈 반환 버튼을 눌렀을 때

### (3) 구현

#### 1) 화면 구현

Qt 디자이너를 이용하여 화면을 작성해 보자. 화면을 작성할 때 프로그램에서 사용할 위젯의 이름을 기억하기 쉽도록 변경하는 것이 좋다. 화면 작성이 끝나면 화면의 이름을 'vendingMachine.ui'로 저장한다.

##### ① 위젯 배치




왼쪽 그림을 참고하여 위젯을 배치한다. 'Label'과 'Label2'의 'pixmap' 속성에 그림 파일을 지정하면 화면에 그림을 표시할 수 있다. 또 화면에 표시되는 글꼴을 변경하기 위해서는 'font' 속성을 수정하면 된다.

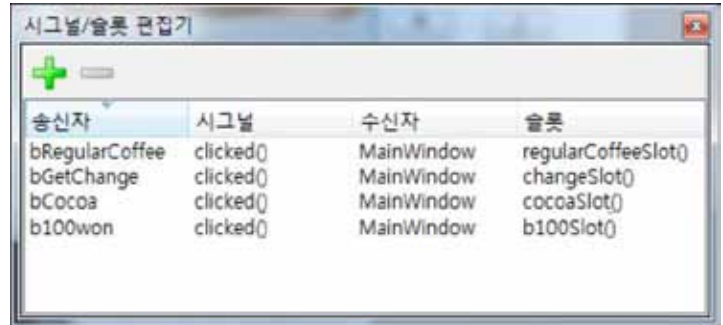
##### ② 위젯 이름 변경

아래 표를 참고하여 위젯 이름을 변경한다. 위젯의 이름은 기본적으로 위젯 종류 뒤에 생성 순서대로 번호가 붙는데, 프로그램에서 사용할 위젯은 기억하기 쉬운 이름을 부여하는 것이 좋다.

구분	변경 전 이름	변경 후 이름	비고
LineEdit	LineEdit	leMoney	투입 금액 표시
PushButton	PushButton	bRegularCoffee	일반 커피 버튼
	PushButton_2	bCocoa	코코아 버튼
	PushButton_3	b100won	100원 투입 버튼
	PushButton_4	bGetChange	잔액 반환 버튼

### ③ 시그널/슬롯 연결

버튼을 클릭하면 적절한 처리를 하기 위해 [F4] 또는  아이콘을 눌러 시그널/슬롯 편집 모드로 변경하고, 오른쪽 그림을 참고하여 시그널과 슬롯을 연결한다. 슬롯의 이름도 의미가 있도록 적절히 바꾸어 준다.



송신자	시그널	수신자	슬롯
bRegularCoffee	clicked()	MainWindow	regularCoffeeSlot()
bGetChange	clicked()	MainWindow	changeSlot()
bCocoa	clicked()	MainWindow	cocoaSlot()
b100won	clicked()	MainWindow	b100Slot()

### ④ 화면 표시 코드 작성

기본적인 GUI 화면을 표시하도록 기본 코드를 작성한다.

해 보기

- ▶ 다음 코드는 'vendingMachine.ui' 파일을 이용해서 화면을 표시하는 코드이다. 코드의 빈 부분을 완성하여 보자.

```
1     import sys
2     from PyQt4 import QtCore, QtGui, uic
3
4     class MyWindowClass(QtGui.QMainWindow, form_class):
5         def __init__(self, parent=None):
6             QtGui.QMainWindow.__init__(self, parent)
7             self.setupUi(self)
8
9     app = QtGui.QApplication(sys.argv)
10
11     myWindow = MyWindowClass(None)
12
13     myWindow.show()
14
15     app.exec_()
```

## 2) 알고리즘 구현

알고리즘 설계를 참고하여 파이썬 코드로 작성한다. 코드를 작성할 때 해당 버튼과 연결된 이름의 슬롯을 함수처럼 정의하면 된다.

- 100원 버튼을 눌렀을 때

```
1 def b100Slot(self):    # 슬롯 정의
2     money = int(self.leMoney.text()) # 현재 투입 금액을 money에 저장
3     self.leMoney.setText(str(money+100)) #money에 100을 더해 화면에 표시
```

해 보기

🔴 다음은 일반 커피 버튼을 눌렀을 때 처리할 슬롯을 구현한 것이다. 코드의 빈 곳을 완성해 보자.

```
1 :    # 슬롯 정의
2     money = int(self.leMoney.text())
3     if money >= 200:
4         self.leMoney.setText(str()) # 커피값을 빼고 표시
5         QMessageBox.information(None, "안내", "일반커피가 준비되었습니다.\n")
6     else:
7         QMessageBox.information(None, "경고", "금액이 {0}원 부족합니다.\n".format
8             (200-money))
```

🔴 코코아 버튼을 눌렀을 때 처리할 슬롯의 코드를 작성해 보자.

🔴 잔돈 반환 버튼을 눌렀을 때 처리할 슬롯을 구현한 것이다. 코드의 빈 곳을 완성해 보자.

```
1  def changeSlot(self):
2       # 현재 투입 금액을 money에 저장
3      if money>0:
4          QMessageBox.information(None,"안내","잔돈 {0}원이 반환됩니다.\n".format
5      (money))
6       # 현재 투입 금액을 0으로 표시
7      else:
8          QMessageBox.information(None,"안내","잔돈이 없습니다.\n")
```

#### (4) 테스트 및 디버깅

프로그램을 작성하고 나서는 언제나 정상적으로 동작하는지 확인하고 오류를 수정하는 과정을 거쳐야 한다. 테스트를 진행할 때는 가능한 모든 경우에 대해 정상적으로 동작하는지를 확인해야 한다.

해 보기

- 완성된 프로그램을 실행해 보고, 잘 동작하는지 확인해 보자. 특히 투입 금액이 제대로 표시되는지, 표시되는 메시지에 이상이 없는지 점검하고 오류가 있으면 수정해 보자.

이상 증상	원인	수정한 부분

다른 사람의 프로그램을 사용해보고 오류가 있는지 확인해 보자.

## (5) 기능 추가

자판기 프로그램에 다음 기능을 추가하려고 한다.

- 500원 동전을 투입할 수 있다.
- 1000원 지폐를 사용할 수 있다.
- 300원짜리 고급 커피를 선택하여 뽑을 수 있다.



- 기존의 화면 설계를 변경하려고 한다. 추가로 필요한 위젯은 무엇이 있는지 표에 적어 보고, 위젯의 종류와 이름을 정해 보자.

종류	이름	용도

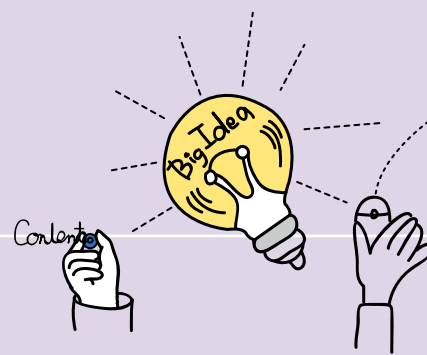


‘vendingMachine.ui’ 파일을 열어 새로운 위젯을 추가하고, 시그널과 슬롯도 추가로 연결해 보자.

기존의 프로그램에 새로운 기능이 동작하도록 코드를 추가해 보자.

그 밖에 추가할 기능이 있는지 생각해 보자. 추가할 기능이 있다면 아이디어를 정리해 보자.

- 1 요즘 사용되는 프로그램은 대부분 그래픽 기반의 유저 인터페이스(GUI)를 가진다. GUI 방식은 CUI 방식보다 사용 방법을 직관적으로 이해할 수 있도록 도와주고, 복잡한 명령어를 입력하지 않고 편리하게 소프트웨어를 활용할 수 있도록 한다.
- 2 파이썬에는 GUI를 구현할 수 있는 다양한 라이브러리가 있다. 그중 PyQt는 디자이너 창과 코딩 영역이 분리되어 있고, 편리하게 화면 디자인을 할 수 있어 많이 활용되는 라이브러리이다.
- 3 GUI 프로그램을 제작하기 위해서는 화면 디자인과 이벤트 처리의 두 단계를 거친다. 화면 디자인은 사용자의 눈에 보이는 프로그램 구조를 설계하는 단계이다. 이벤트 처리는 화면에 배치한 다양한 요소에 이벤트가 발생했을 때 그 이벤트를 적절히 처리하기 위한 코드를 작성하는 단계이다.
- 4 PyQt에서 위젯의 배치는 디자이너 창에서 이루어지며 드래그 앤 드롭으로 진행한다. 위젯을 배치한 후에는 속성 편집기를 이용하여 위젯의 속성을 저장한다. 화면의 작성이 끝나면 화면 파일을 저장한다. 디자이너를 이용하여 만든 화면은 xml 형식의 파일로 'ui'라는 확장자로 저장된다.
- 5 PyQt에서는 프로그램의 특정한 신호를 시그널이라고 부르고, 시그널이 발생했을 때 이벤트 처리를 하기 위해 실행되는 코드를 슬롯이라고 부른다. 이벤트 처리를 위해서는 시그널이 발생하였을 때 처리할 슬롯을 작성해야 한다.
- 6 문제를 해결하기 위한 소프트웨어를 제작할 때는 소프트웨어를 활용해야 하는 요구 사항을 분석한 후 요구 사항을 만족하는 프로그램을 설계한다. 화면 구성과 알고리즘을 설계한 후 이를 프로그래밍 언어를 활용해 컴퓨팅 기기에서 동작하는 소프트웨어의 형태로 변환하고, 소프트웨어의 동작을 테스트한다. 동작이 제대로 되지 않는다면 원인을 파악하여 수정한 후 소프트웨어를 완성한다.



# IV

## CHAPTER



# 앱 프로그래밍

01 주소록 앱

02 안전 알리미 앱

03 게임 앱

우리는 우리의 삶을 더욱 편리하게 만들어주는 다양한 모바일 기기와 함께 생활하고 있다. 우리가 일상적으로 사용하고 있는 모바일 기기에는 다양한 방식으로 활용할 수 있는 여러 가지 센서가 포함되어 있고, 우리가 내리는 명령을 실행시켜 주는 처리 장치가 포함되어 있다. 이 단원에서는 모바일 기기에 담겨있는 다양한 센서를 활용한 프로그래밍을 진행하고, 개발한 모바일 앱을 실제 모바일 기기에서 실행해 보며 소프트웨어의 실행 과정에 대해 이해하도록 한다.



# 주소록 앱



## ! 학습 목표 !

1. 앱 인벤터의 의미와 기능을 설명할 수 있다.
2. 앱 인벤터의 디자이너 창과 블록 창을 활용하여 프로그래밍 할 수 있다.
3. TinyDB를 이용하여 자료를 저장할 수 있다.
4. 리스트뷰어 컴포넌트를 활용하여 사용자 인터페이스를 개선할 수 있다.

## 1. 앱 인벤터

컴퓨팅 기기는 많은 일을 수행할 수 있지만 그러한 작업을 수행하게 하는 명령은 사람이 내리게 된다. 컴퓨터에게 특정한 일을 시키기 위한 소프트웨어는 컴퓨터가 이해할 수 있는 프로그래밍 언어를 이용하여 작성한다. 다양한 프로그래밍 언어 중 앱 인벤터(MIT App Inventor)는 모바일 기기에서 많이 사용하는 운영체제인 안드로이드 운영체제에서 구동되는 어플리케이션(앱)을 제작하기 위한 프로그래밍 언어이다. 일반적으로 안드로이드 앱을 제작하기 위해서는 텍스트 기반의 프로그래밍 언어를 익혀야 하지만, 앱 인벤터는 블록을 결합하는 간단한 프로그래밍 방식을 통해 실제 모바일 기기에서 동작하는 앱을 제작할 수 있는 장점이 있다.

### 앱 인벤터의 특징

- 블록 쌓기 형태의 프로그래밍 언어: 명령어를 가져다 사용할 수 있음.
- 클라우드 방식의 프로젝트 제작 및 저장
- 온라인으로 프로그래밍 가능
- 안드로이드 운영체제가 탑재된 기기에서 실제로 실행되는 앱 제작



### 안드로이드 운영체제와 Java

안드로이드 운영체제는 2015년 현재 스마트폰과 스마트기기에서 가장 많이 사용되고 있는 운영체제이다. 이러한 안드로이드 운영체제의 인기에 힘입어 현재 세계에서 가장 많이 사용되고 있는 프로그래밍 언어 중 하나가 Java이다. 우리가 사용하는 대부분의 안드로이드 앱은 Java 언어를 사용하여 제작된다. 본격적으로 모바일 프로그래밍을 공부하고자 한다면 Java 언어를 학습하는 것이 좋다.

## (1) 앱 인벤터 사용 환경

앱 인벤터는 웹 브라우저를 기반으로 구동되는 클라우드 컴퓨팅 방식을 활용한다. 현재 널리 사용되는 웹 브라우저 중에서 크롬(Chrome), 파이어폭스(Firefox), 사파리(Safari) 브라우저에서 구동된다. 또한 앱 인벤터는 자신이 제작한 프로젝트를 네트워크를 통해 저장하기 때문에 구글(Google) 계정으로 로그인하여 관리해야 한다.

### 앱 인벤터 사용 환경

- 인터넷에 연결된 컴퓨터
- 앱 인벤터가 구동되는 웹 브라우저(크롬, 파이어폭스, 사파리)
- 구글 계정
- 실행 결과를 확인할 수 있는 모바일 기기

### 클라우드 컴퓨팅(cloud computing)

대부분의 소프트웨어나 디지털 자료는 내가 가지고 있는 컴퓨팅 기기에 저장되어 있다. 하지만 클라우드 컴퓨팅은 네트워크로 연결된 다른 컴퓨터로 소프트웨어를 실행시키거나 자료를 저장할 수 있도록 한 시스템이다. 네트워크로 촘촘히 연결된 컴퓨팅 환경이 마치 하늘에 떠 있는 구름과 같이 알 수 없다고 하여 클라우드 컴퓨팅이라 부른다.

## (2) 앱 인벤터 접속하기

앱 인벤터를 사용하기 위해서는 앱 인벤터 홈페이지에 접속해야 한다. 웹 브라우저의 주소창에 앱 인벤터 홈페이지 주소를 입력하여 이동한다.

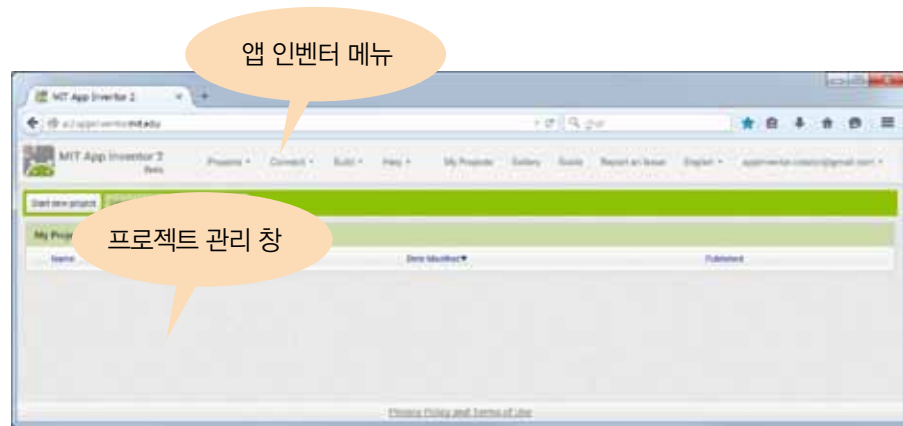
**앱 인벤터 홈페이지 주소 - <http://appinventor.mit.edu>**

앱 인벤터 홈페이지는 앱 인벤터에 대한 최신 소식과 학습에 도움이 되는 많은 자료가 있기 때문에 앱 인벤터에 대한 다양한 정보를 얻으려면 홈페이지를 잘 살펴보는 것이 좋다.

앱 인벤터 홈페이지에서 앱을 제작하기 위한 화면으로 이동하기 위해서는 앱 인벤터 홈페이지 오른쪽 상단의 **Create apps!** 버튼을 클릭한다. 혹은 웹 브라우저에서 아래의 주소를 입력하여 바로 이동할 수도 있다.

**앱 인벤터 앱 제작 주소 - <http://ai2.appinventor.mit.edu>**

처음 앱 인벤터에 접속하였다면 구글 사이트에서 사용하는 아이디와 패스워드로 로그인해야 개발 환경에 접속할 수 있다. 구글 계정이 없다면 계정을 생성하고 로그인 하도록 하자. 앱 인벤터에 접속해서 처음 만나는 화면은 다음과 같다.



[그림 IV-1] 앱 인벤터 초기 화면

앱 인벤터에서는 아래의 화면에 진입하기 전에 구글 계정에 접근할 수 있는 권한을 요구한다. 이때 권한을 허용해 주어야 앱 인벤터를 사용할 수 있다. 또한 서비스 사용 약관도 동의해야 앱 인벤터를 사용할 수 있다.

위의 화면은 앱 인벤터에 접속하면 가장 먼저 나타나는 프로젝트 관리 화면으로 내가 만든 프로젝트를 한눈에 볼 수 있는 곳이다. 앱 인벤터 메뉴에서는 프로젝트 관리(Project), 기기 연결(Connect), 앱 빌드(Build), 도움말(Help), 내 프로젝트(My Project) 등의 메뉴를 사용할 수 있다. 프로젝트 관리 창에서는 새로운 프로젝트를 시작하거나 내가 만든 앱 인벤터 프로젝트를 모두 관리할 수 있다.

### (3) 디자이너 창

간단한 앱을 하나 제작하면서 앱 인벤터의 사용 방법을 익혀 보도록 하자. 우리가 제작할 앱은 '주소록 앱'이다. 주소록 앱의 기능은 다음과 같다.



#### 주소록 앱의 기능

- 이름, 전화번호, 이메일 주소를 입력할 수 있는 공간이 있다.
- '저장하기' 버튼을 누르면 아래쪽 공간에 저장한 내용이 표시된다.
- '초기화하기' 버튼을 누르면 위의 입력 공간의 내용이 모두 지워진다.

앱 인벤터 프로그래밍은 앱의 외형을 디자인한 후, 블록을 결합하여 프로그램을 제작하는 순서로 이루어진다. 다음 해보기를 따라서 앱 제작 과정을 익혀 보자.

1. 먼저 'MyAddressBook'라는 이름으로 프로젝트를 하나 생성해 보자. 프로젝트를 생성하기 위해서는 프로젝트 관리 창에서 왼쪽 위에 있는 **Start new project** 버튼을 클릭한다. 버튼을 클릭하면 오른쪽 화면과 같이 프로젝트를 새로 생성하는 대화창이 나타난다.

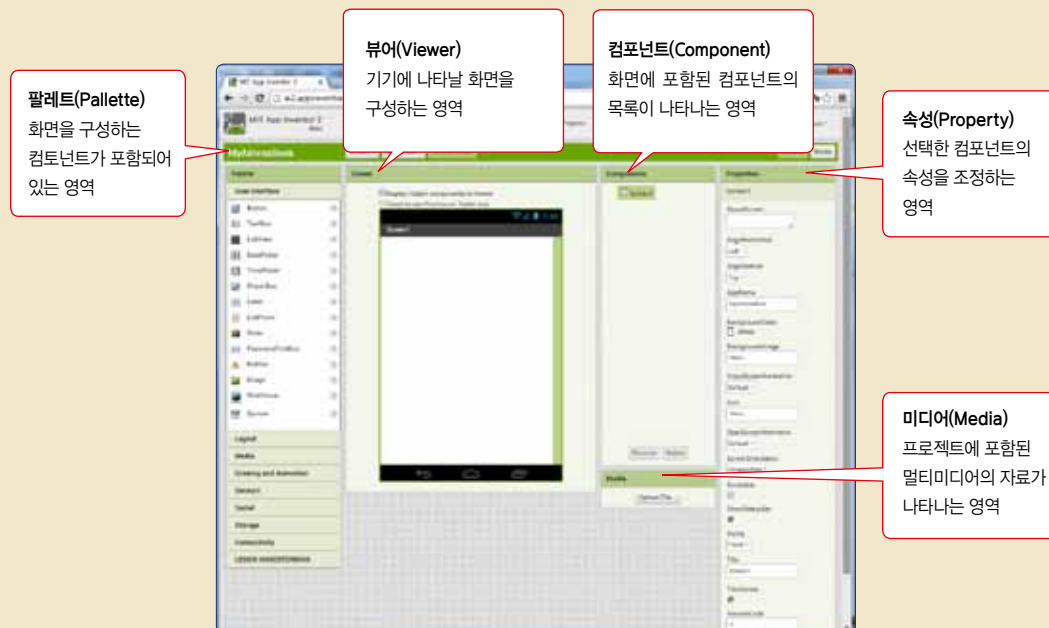


## 보충 학습

## 앱 인벤터 프로젝트 이름

앱 인벤터 프로젝트 이름은 영문자로 시작하여야 하며, 영문자, 숫자, 언더바(\_)만 사용할 수 있다. 또한 프로젝트 이름은 나중에 모바일 기기에 앱을 설치하면 앱의 이름이 되기 때문에 알아보기 쉬운 이름으로 결정해야 한다.

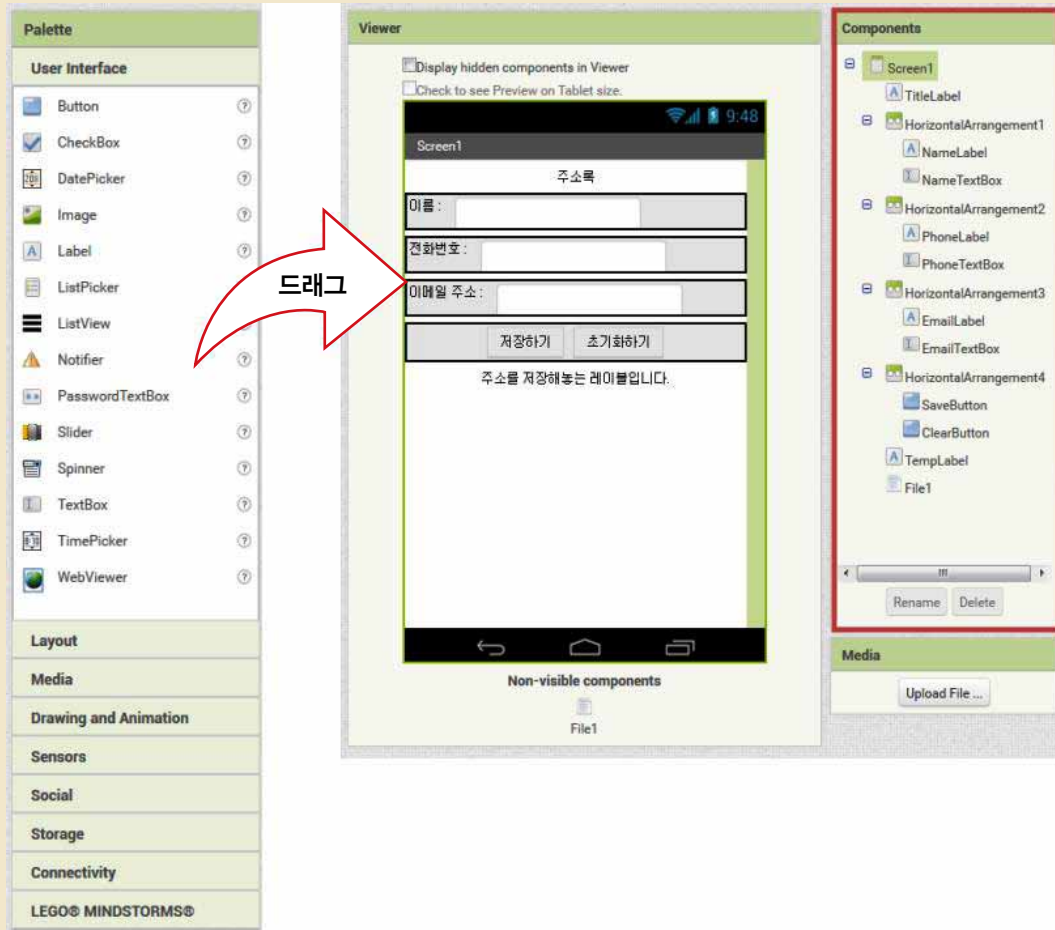
2. 'MyBookAddress' 프로젝트가 생성되면 아래와 같은 화면이 나타날 것이다. 이 화면이 앱 인벤터의 디자이너(Designer) 창이다. 디자이너 창에서는 앱 인벤터에서 사용하는 다양한 컴포넌트(Component)를 모바일 기기의 화면에 맞게 배치하는 작업을 진행할 수 있다.



화면을 구성하기 위해서는 가장 왼쪽의 팔레트 영역에서 뷰어 영역으로 컴포넌트를 드래그해서 배치하고, 오른쪽의 속성 영역에서 컴포넌트의 속성을 수정한다. 디자이너 창은 블록 프로그래밍을 진행하기 전에 앱의 외형을 설계하고 앱에 사용할 다양한 컴포넌트를 결정하는 단계에서 활용한다.



1. 디자이너 창에서 아래와 같이 컴포넌트를 배치해 보자.
2. 컴포넌트 영역에서 배치한 컴포넌트의 이름을 아래와 같이 바꾸어 보자.



### 앱 인벤터의 컴포넌트


안드로이드 어플리케이션은 크게 네 개의 컴포넌트(component)로 구성되어 있다. 사용자 인터페이스를 구성하는 기본 단위가 되는 액티비티(Activity), 사용자의 눈에 보이지 않지만 계속 실행되고 있는 서비스(Service), 시스템에서 발생하는 방송에 귀 기울이고 있다가 신호를 수신하고 전달하는 방송 수신자(Broadcast Receiver), 다른 응용 프로그램을 위해 자신의 자료를 제공하는 자료 제공자(Content Provider)가 그것이다. 앱 인벤터에서는 위의 역할을 담당하는 많은 프로그램 구성 요소를 분류하여 팔레트 영역에 배치해 놓았고, 앱 인벤터에서는 이 구성 요소들을 컴포넌트라고 부른다.

컴포넌트의 속성은 가장 오른쪽의 속성 창에서 변경할 수 있다. 각 컴포넌트마다 가지고 있는 속성은 조금씩 다르지만 공통적으로 가지고 있는 일반 속성은 아래 버튼 컴포넌트의 속성 설명을 참고하여 변경하면 된다.



**Properties**

Button1

BackgroundColor  Default

Enabled ☒

FontBold ☐

FontItalic ☐

FontSize 14.0

FontTypeface default \*

Height Automatic...

Width Automatic...


Image None...

Shape default \*

ShowFeedback ☒

Text Text for Button1

TextAlignment center \*

TextColor  Default

Visible ☒

BackgroundColor: 컴포넌트의 배경 색상을 결정한다.

Enable: 컴포넌트를 사용 가능하게 한다.

FontBold: 컴포넌트에 사용하는 글꼴을 굵게 만든다.

FontItalic: 컴포넌트에 사용하는 글꼴을 기울임체로 만든다.

FontSize: 컴포넌트에 사용하는 글꼴의 크기를 결정한다.

FontTypeface: 컴포넌트에 사용하는 글꼴을 결정한다.

Height: 컴포넌트의 세로 길이를 결정한다. 결정 방식은 네 가지이다.

Automatic - 자동으로 결정한다.

Fill Parent - 컴포넌트가 포함되어 있는 다른 컴포넌트에 딱 차게 길이를 늘린다.

pixels - 픽셀 단위로 크기를 조정한다.

percent - 비율 단위로 크기를 조정한다.

Width: 컴포넌트의 가로 길이를 결정한다.

Image: 컴포넌트에 다른 이미지를 불러온다.

Shape: 버튼의 경우 버튼의 모양을 변경한다.

Text: 버튼에 나타날 글을 여기에 입력한다.

TextAlignment: 글을 정렬한다.

TextColor : 글자색을 결정한다.

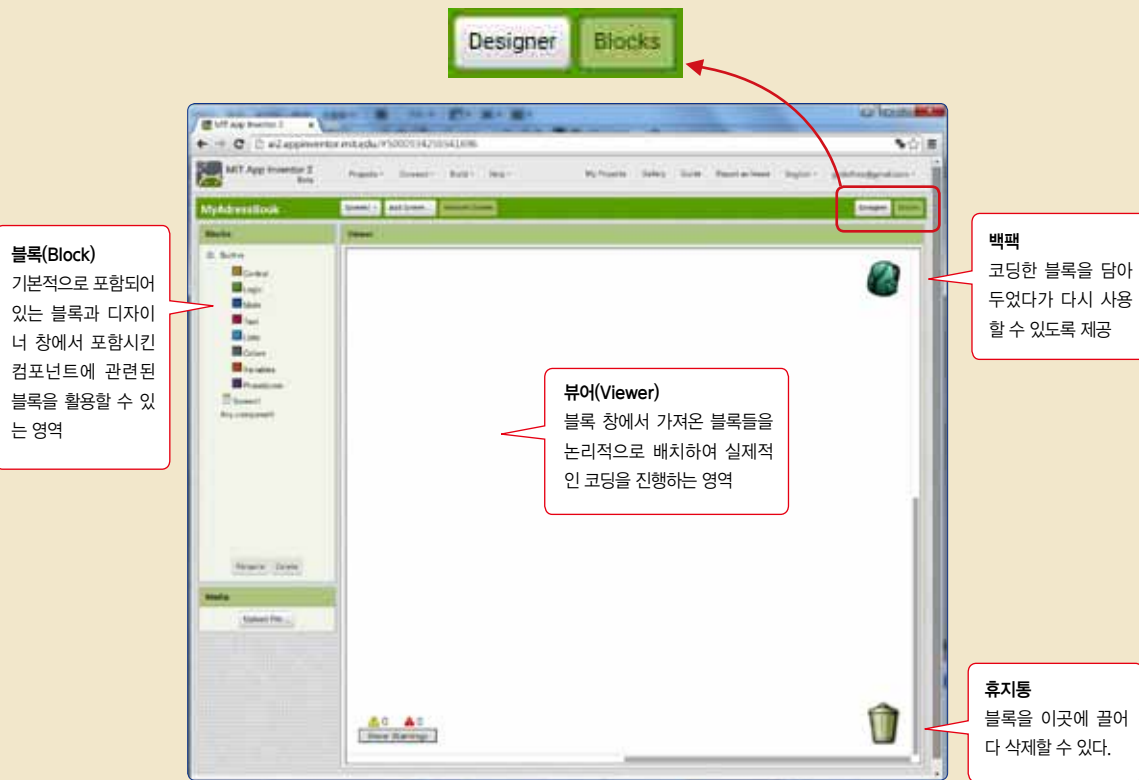
Visible: 컴포넌트를 보이지 않게 할 수 있다.

#### (4) 블록 창

디자이너 창은 앱의 외형을 구성하기 위한 영역이다. 우리가 디자이너 창에서 배치한 컴포넌트에 명령을 내리기 위해서는 블록(Block) 창으로 이동해야 한다.

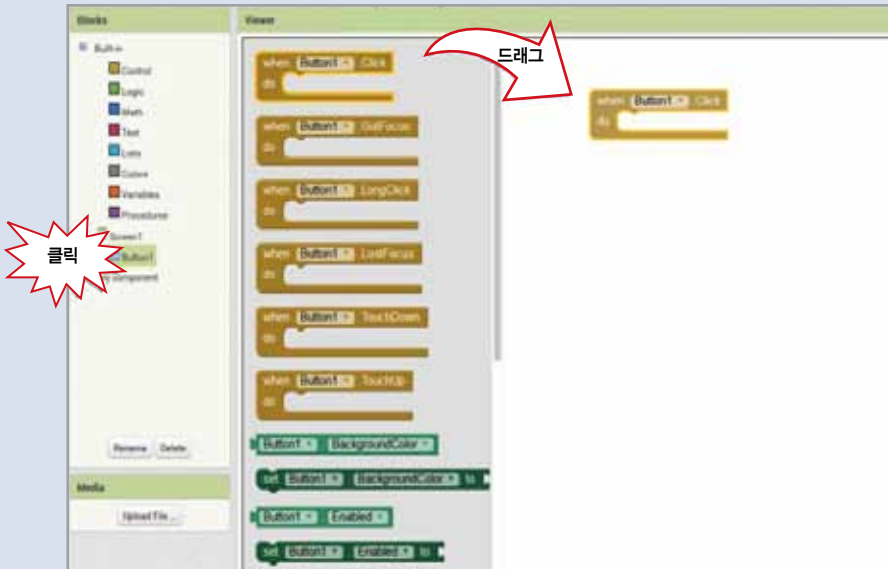
해 보기

디자이너 창의 오른쪽 위에 있는 두 개의 버튼이 디자이너 창과 블록 창을 번갈아가며 이동할 수 있도록 하는 버튼이다. **Designer** **Blocks** 버튼을 눌러 블록 창으로 이동해 보자. 아래 화면이 블록 창이다.



블록 창 왼쪽의 블록 영역에는 앱 인벤터에 기본적으로 내장되어 있는 블록들과 디자이너 창에서 선택한 컴포넌트가 나타난다. 내장 블록이나 컴포넌트를 선택하면 그 컴포넌트가 제공하는 블록이 뷰어 영역의 왼쪽 부분에 나타나게 된다. 필요한 블록을 선택하여 오른쪽 빈 공간으로 드래그하면 그 블록을 사용할 수 있고, 여러 블록을 선택하여 결합시키는 방식으로 프로그래밍이 진행된다.

## 블록 코딩 방법



❶ 블록 창에서 아래와 같이 블록을 배치해 보고 이 블록이 어떤 기능을 수행하게 될지를 설명해 보자.





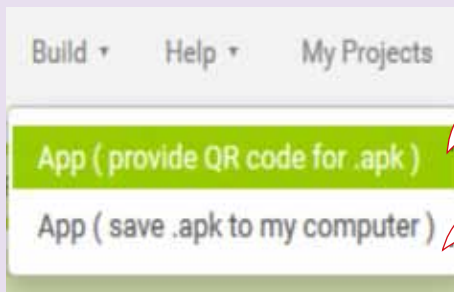
에뮬레이터 활용 방법은  
부록을 참조하도록 한다

## (5) 앱 빌드와 설치

앱 인벤터로 개발한 앱은 컴퓨터에서 구동되는 가상 기기인 에뮬레이터(emulator)에서 실행할 수도 있고, 실제 안드로이드 기기에 설치하여 동작을 확인할 수도 있다. 하지만 에뮬레이터는 모바일 기기의 다양한 기능들을 직접 조작해볼 수 없다는 제약점이 있다. 따라서 이후의 설명은 실제 안드로이드 기기에 앱을 설치하여 동작하는 것을 기준으로 한다.

앱이 완성되면 이 앱을 실제로 안드로이드 기기에 설치할 수 있는 설치 파일 형태로 변환해 주어야 한다. 이렇게 설치 파일을 만드는 과정을 앱 빌드(build)라고 한다. 앱을 빌드하기 위해서는 앱 인벤터 메뉴의 빌드(Build) 메뉴를 사용한다. 앱 인벤터는 두 가지 앱 빌드 방식을 제공한다.

### 앱 빌드 방식



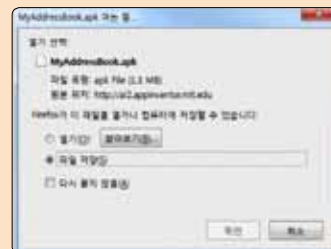
• 앱 빌드 후 QR 코드로 앱 다운로드 제공

• 앱 빌드 후 컴퓨터로 apk 파일 다운로드

### QR 코드로 앱을 빌드했을 때 나타나는 화면



### apk 파일을 저장할 때 나타나는 화면



만들어진 QR 코드를 QR 코드를 인식할 수 있는 앱으로 촬영하면 자동으로 앱을 다운로드하여 설치하는 과정이 진행된다. apk 파일을 다운로드하면 이를 USB나 무선통신을 통해 모바일 기기로 전송하여 설치를 진행할 수 있다. 이 apk 파일을 다른 사람에게 전송하면 다른 사람도 이 앱을 설치할 수 있도록 할 수 있다.

### 안드로이드 설치 파일

안드로이드 운영체제에 설치되는 앱의 확장자는 .apk 이다. 누군가 스마트기기로 특정 파일을 전송하였는데 그 파일의 확장자가 apk라면 파일을 클릭하는 순간 앱의 설치가 진행될 수 있다. 최신 안드로이드 버전에서는 출처가 분명하지 않은 설치 파일을 클릭하여도 바로 설치가 되지 않도록 하고 있지만 의심스러운 링크는 클릭하지 않는 것이 좋다.

### MIT AI2 Companion

앱 인벤토리와 함께 사용할 수 있도록 제공하는 앱이 MIT AI2 Companion 앱이다. MIT AI2 Companion 앱은 앱 인벤토리와 사용하는 모바일 기기를 연결하고, QR 코드를 인식하여 앱을 설치하도록 돕는 역할을 담당한다. MIT AI2 Companion 앱은 구글 플레이스토어에서 내려받을 수 있다.

해 보기

1. QR 코드를 통해 앞에서 만든 앱을 모바일 기기에 설치해 보고 앱이 어떻게 동작하는지 설명해 보자.

2. 초기화하기 버튼을 누르면 입력한 내용이 모두 지워지도록 코드를 작성해 보자.



#### 〈힌트〉

내용이 모두 지워진다 = 포함된  
내용이 하나도 없는 상태이다.

## 2. 데이터베이스에 자료 저장하기

우리는 앞에서 아주 간단한 주소록 앱을 제작하면서 앱 인벤터를 사용하는 방법을 익혔다. 우리가 앞에서 만든 앱은 너무 간단하기 때문에 거의 기능이 없는 것이나 마찬가지이다. 이 주소록 앱이 가지고 있는 문제점은 무엇이고, 어떠한 기능을 더 추가할 수 있을까?

앱의 문제점	앱의 기능 추가

이 앱이 가지고 있는 가장 큰 문제점 중 하나는 앱이 종료되면 저장된 자료가 모두 없어진다는 점이다. 또한 필요한 내용에 대한 검색이 어렵다. 이러한 문제를 해결하기 위하여 앱 인벤터에서는 자체적인 데이터베이스를 사용한다.

### 데이터베이스(Database)

데이터베이스는 대규모의 데이터 집합을 사용자가 편리하게 관련 정보 항목을 검색하고 추출하는 데 사용할 수 있는 추상적 도구로 변환해 주는 시스템이다. 오늘날에는 대단히 큰 규모의 데이터를 저장하고 사용하고 있는데, 이러한 자료를 체계적으로 분류하여 정리하고 필요할 때마다 원하는 종류의 자료를 편리하게 검색하여 추출할 수 있도록 하는 다차원적인 자료 저장 시스템을 데이터베이스라고 한다.

## (1) TinyDB

앱에서 활용하고 있던 내용을 데이터베이스에 저장하기 위해서는 TinyDB 컴포넌트가 필요하다. 앱 인벤터에서 사용하는 TinyDB는 가장 기초적인 형태의 데이터베이스라고 할 수 있다. 데이터를 저장하고 찾기 위한 태그(Tag)만 기억하고 있다면 원하는 자료를 저장하고 찾아올 수 있다. TinyDB에 저장된 자료는 앱이 종료되더라도 없어지지 않아 앱을 다시 실행시켜 불러올 수 있다.

해 보기

● 앱에 TinyDB 컴포넌트를 추가해 보자. [Palette] - [Storage]에서 TinyDB 컴포넌트를 찾을 수 있다.

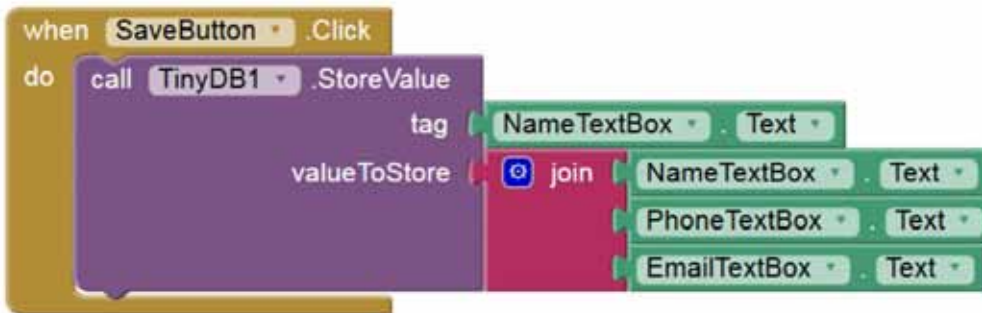


TinyDB 컴포넌트를 추가하면 뷰어 아래쪽에 추가된 것을 확인할 수 있다. 실제로 화면에 보이는 컴포넌트는 위쪽의 스크린 영역에 나타나게 되고 눈에 보이지 않는 컴포넌트는 아래쪽의 보이지 않는 컴포넌트(Non-visible components) 영역에 나타나게 된다.



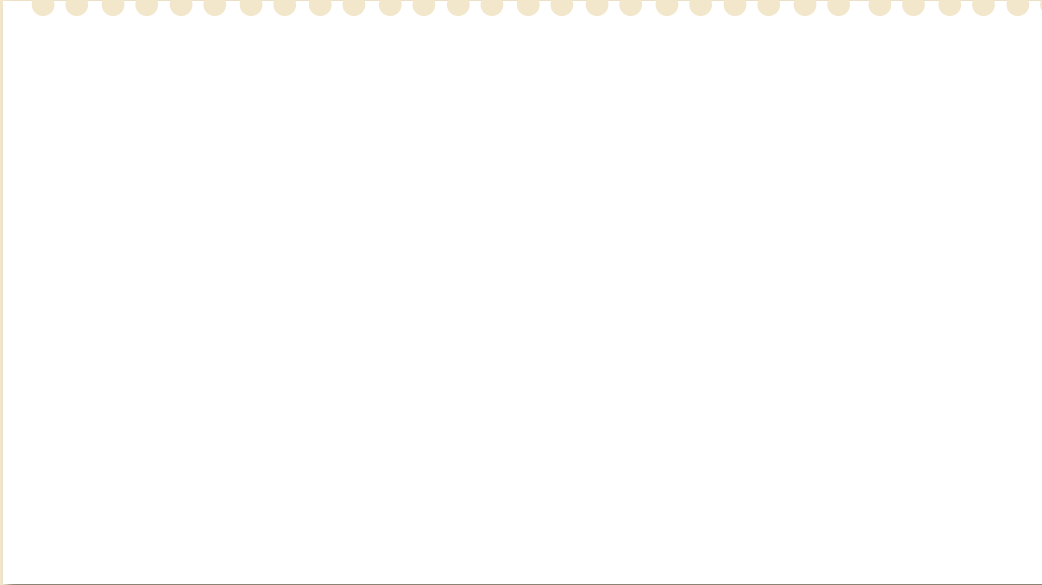
## (2) 코딩하기

컴포넌트를 추가한 다음 앞에서 작성한 코드를 다음과 같이 변경해 보자.





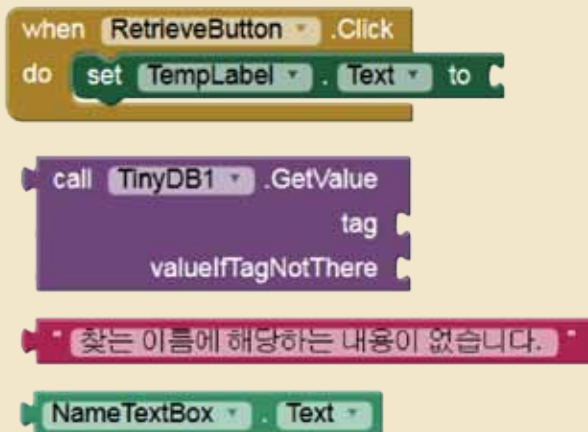
1. 앞의 코드가 어떻게 동작하는지 설명해 보자.



2. 저장해 둔 주소를 불러오는 동작을 추가해 보자.



위의 그림처럼 '불러오기' 버튼을 배치하고 이름을 입력한 후 '불러오기' 버튼을 누르면 데이터베이스에서 해당 이름으로 검색하여 주소를 불러오도록 아래 블록들을 활용하여 코딩해 보자.



### 3. 사용자 인터페이스 개선하기

우리가 앞에서 작성한 앱은 주소를 입력할 수 있게만 제작된 앱이다. 이 앱은 어떤 자료가 입력되어 있는지를 한 눈에 보기 어렵고, 모바일 기기의 특징을 전혀 살리고 있지 못하고 있다. 몇 가지 컴포넌트를 더 추가하여 우리가 사용하기 좋은 형태로 앱을 수정해 보자.

#### (1) 컴포넌트 추가

조금 더 직관적으로 앱을 사용하기 위해서 여기서는 리스트뷰(ListView) 컴포넌트를 활용해 보자. 리스트는 프로그램에서 사용하는 자료의 연속된 묶음이다. 리스트뷰나 리스트피커(ListPicker)는 이 리스트라는 자료 형태를 활용하는 컴포넌트이다.

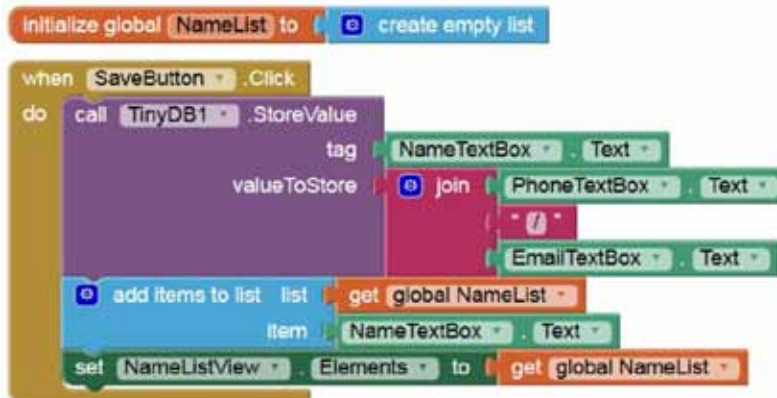
해 보기

- 오른쪽 그림처럼 앱에 리스트뷰 컴포넌트를 추가해 보자. [Palette] - [User Interface]에서 리스트뷰 컴포넌트를 찾을 수 있다.



#### (2) 코딩하기

컴포넌트를 추가한 다음 앞에서 작성한 코드를 다음과 같이 변경한다.



1. 위의 코드가 어떻게 동작하는지 설명해 보자.

2. 위의 코드는 리스트뷰에 이름을 저장하는 코드이다. 아래 블록을 활용하여 리스트에서 이름을 선택했을 때 전화번호와 이메일 주소가 출력되는 코드를 완성해 보자.



리스트뷰 컴포넌트에서 활용할 수 있는 블록은 다음과 같다.

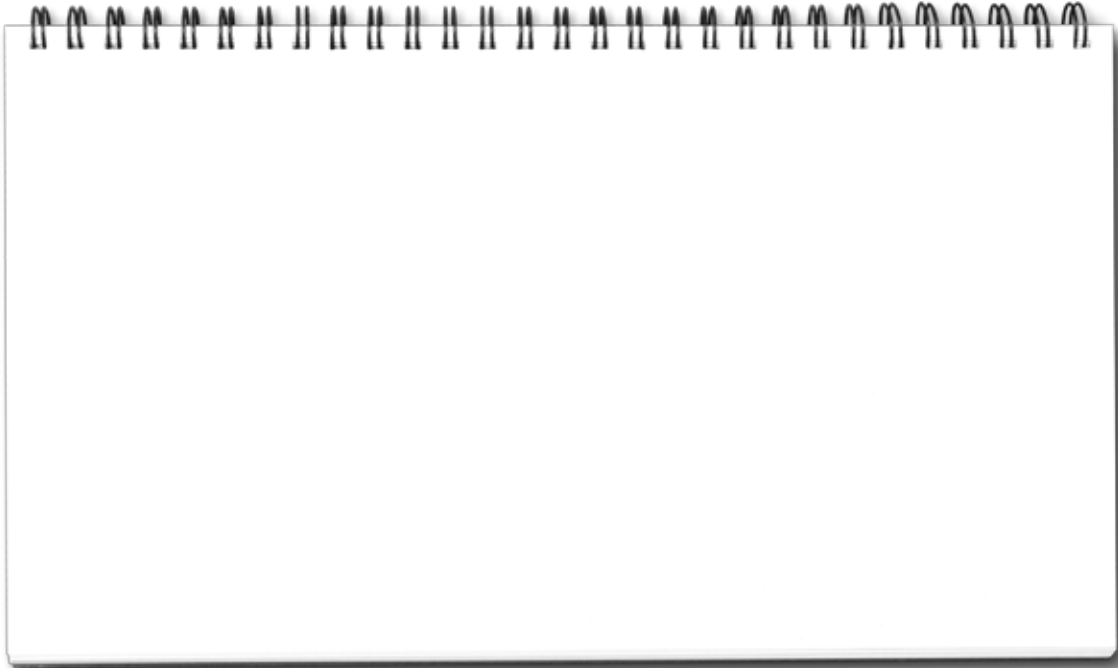
	리스트뷰에서 자료를 선택하고 나서 이벤트를 발생시킬 때 사용
	리스트뷰에 표시할 자료 설정하기 (리스트로 받아오기)
	리스트뷰에 표시할 자료 설정하기 (문자열로 받아오기)
	현재 선택된 원소 현재 선택된 원소의 위치값



## 4. 앱 확장하기

이 앱의 기능을 더욱 확장해 보자. 어떠한 형태의 기능을 추가할 수 있는가?

(1) 앱의 디자인을 개선하기 위한 아이디어를 기록해 보자.

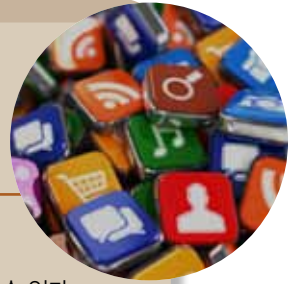


(2) 앱에 추가하고 싶은 기능을 기록해 보고 이러한 기능이 앱 인벤터로 실현 가능할지 프로그래밍해 보자.





# 안전 알리미 앱



## 학습 목표

1. 위치 센서 컴포넌트에 포함된 블록의 기능을 설명할 수 있다.
2. 위치 센서 컴포넌트가 제공하는 위도와 경도 값을 활용하여 앱을 제작할 수 있다.
3. 문자 메시지로 위치 정보를 자동으로 전송할 수 있다.
4. ActivityStarter 컴포넌트로 다른 앱을 실행시킬 수 있다.

## 1. 모바일 기기의 위치 정보

### 생각해 보기



등산을 끝내고 산을 내려오던 나는 발을 헛디뎈 길옆으로 미끄러졌다. 한참 뒤에 정신을 차린 나는 주변이 어두워져 아무것도 보이지 않는다는 것을 알았고, 내가 어디쯤에 있는지도 알 수가 없었다. 내가 가지고 있는 것이라곤 주머니 속에 넣어두었던 스마트폰뿐이었다. 아무리 도움을 요청해도 주변에 나를 도와줄 수 있는 사람은 없었다. 나는 어떻게 이 문제를 해결할 수 있을까?

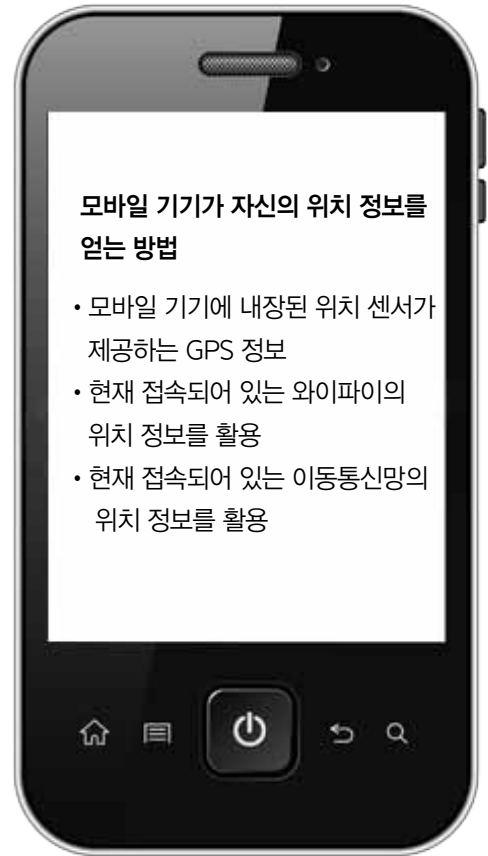
### 해결책 생각해 보기



컴퓨팅 기기에 사용되는 부품의 성능은 시간이 지날수록 향상되고 있고, 크기는 점점 작아지고 있다. 우리가 사용하는 스마트폰이나 태블릿 PC 같은 컴퓨팅 기기도 휴대하고 다닐 수 있을 정도로 작아져서 많은 사람들이 몸에 지니고 다닌다. 이러한 모바일 기기에는 위치 센서(LocationSensor)가 탑재되어 있어 자신의 위치를 파악할 수 있다.



[그림 IV-2] 위치 파악 화면의 예



### 모바일 기기가 자신의 위치 정보를 얻는 방법

- 모바일 기기에 내장된 위치 센서가 제공하는 GPS 정보
- 현재 접속되어 있는 와이파이의 위치 정보를 활용
- 현재 접속되어 있는 이동통신망의 위치 정보를 활용

위의 방법들 중에서 가장 정확한 정보는 모바일 기기에 내장된 위치 센서가 제공하는 GPS 정보이다. 우리는 위치 센서가 제공하는 다양한 위치 정보를 이용하여 유용한 앱을 계획하고 제작할 수 있다.

### GPS(Global Positioning System)

GPS는 우리말로 범지구 위치 결정 시스템으로 부르며 원래 미국 국방부에서 군사용으로 개발한 위치 정보 시스템이다. 지구 주위를 도는 24개의 인공위성에서 발신하는 마이크로파를 GPS 수신기에서 수신하여 자신의 위치값을 계산한다. 어떤 기기여도 GPS 전파를 수신하는 장치만 있으면 GPS정보를 무료로 얻을 수 있다.



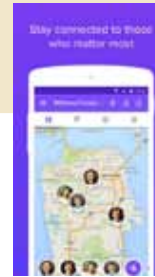
1. 다음 앱 중 하나를 설치하여 각 앱이 어떤 기능을 가지고 있는지 설명해 보자.



내 위치 추적기



지나리 LBS



가족 위치 추적기

2. 위의 앱이 제공하는 기능을 아래에 기록해 보자.

3. 위의 앱이 제공하는 기능을 수행하기 위해서는 어떤 센서나 기능이 있어야 할지 생각해 보자.

## (1) 위치 센서

위치 센서는 GPS 모듈이나 와이파이, 이동통신망에서 제공하는 위치 정보를 수신하고 다른 앱에 전달하는 컴포넌트이다.

### 위치 센서가 제공하는 정보

- 위도(latitude): 지도에서 적도를 기준으로 남쪽과 북쪽의 위치를 나타내는 정보( $90^{\circ}$  N~ $90^{\circ}$  S)
- 경도(longitude): 영국에 있는 본초자오선을 기준으로 동서로 얼마나 떨어져 있는지를 나타내는 위치( $180^{\circ}$  E~ $180^{\circ}$  W)
- 고도(altitude): 평균 해수면을 0으로 놓고 측정한 높이(m 단위)
- 주소(address): 주소 데이터베이스에서 찾은 현재 위치의 주소

### 생각해 보기

앱 인벤터에서 위치 센서가 제공하는 정보를 어떻게 활용할 수 있을지 생각해 보자.

활용안 생각해 보기





다음과 같이 동작하는 앱을 제작해 보자.

앱을 실행하면 앱이 현재 위치를 알려준다.

앱 인벤터의 디자이너 창에서 다음과 같이 화면을 구성해 보자.



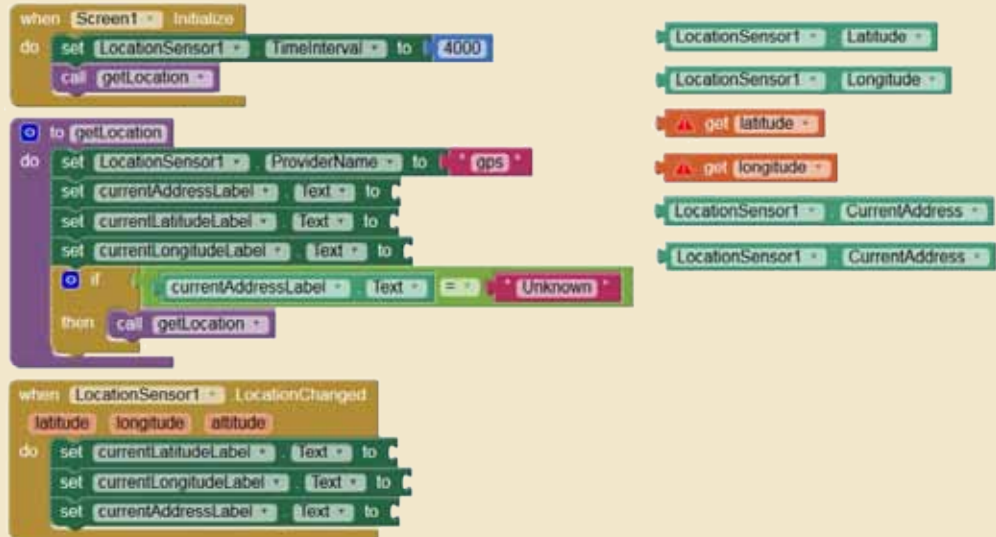
위의 앱 디자인을 조금 더 보기 좋게 꾸며 보자.



## (2) 코딩하기

해 보기

1. 아래의 미완성된 코드를 완성해서 앱이 정상적으로 동작하도록 해보자.



2. getLocation 함수는 앱에서 어떤 역할을 하는지 설명해 보자.

3. 위의 코드를 QR 코드나 apk 파일로 제작하여 모바일 기기에 설치한 다음 나의 위치가 어디인지 확인해 보자. 어느 장소에서 GPS 정보를 잘 얻을 수 있는가?

## 2. 긴급 연락 보내기

생각해 보기



나는 이제 나의 위치를 알 수 있게 되었다. 나의 위치를 효과적으로 알릴 수 있는 방법이 있을까? 스마트폰에서 간단하게 연락할 수 있는 방법은 무엇이 있을까?

모바일 기기가 이동통신과 연결되어 있다면 전화를 걸거나 문자 메시지를 보낼 수 있다. 긴급한 상황에서는 현재의 위치와 긴급 상황임을 알리는 연락을 터치 한 번으로 지정된 번호로 즉시 전송할 수도 있다.

이전까지 만들었던 앱에 다음 기능을 추가해 보자.

- 버튼을 눌러 지정된 번호로 한 번에 전화하는 기능
- 버튼을 눌러 지정된 번호로 나의 위치를 전송하는 기능

### (1) 컴포넌트 추가

연락과 관련된 컴포넌트는 [Palette] - [Social] 메뉴에 포함되어 있다. 전화를 걸기 위해서는 전화(PhoneCall) 컴포넌트를 사용하고, 문자 메시지를 보내기 위해서는 문자(Texting) 컴포넌트를 사용한다.

해 보기

- ▶ 오른쪽 그림과 같이 전화를 걸고, 문자 메시지를 보내기 위한 버튼을 배치해 보자.

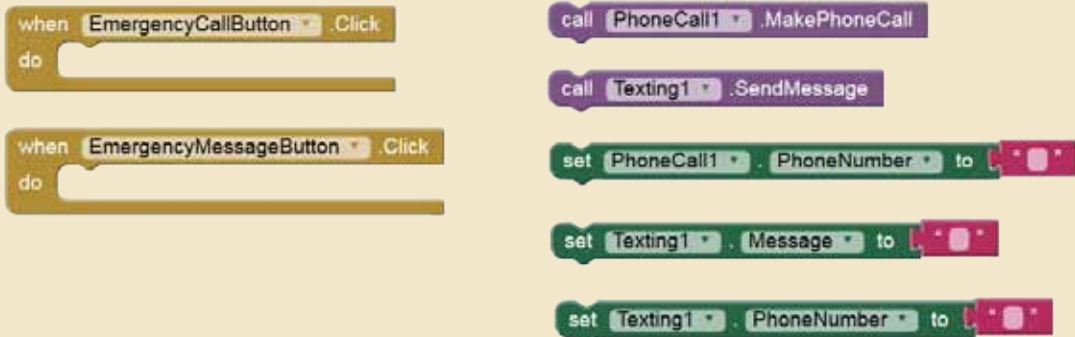


## (2) 코딩하기

전화 걸기와 문자 메시지를 보내기 위해서 공통적으로 필요한 정보는 전화번호이다. 문자 메시지에는 추가적으로 보내고자 하는 메시지 내용이 필요하다.

해 보기

- 아래 그림에 나타난 블록을 배치하여 버튼을 눌러 긴급 전화를 걸거나 긴급 문자를 보내는 코드를 작성해 보자.



- 옆의 그림처럼 긴급번호를 변경할 수 있는 텍스트 박스와 버튼을 배치하고, 연락하는 번호를 변경할 수 있는 코드를 추가해 보자.



### 3. 지도 앱에서 내 위치 확인하기

생각해 보기



긴급 연락을 보내서 나의 위치를 알리고 도움을 요청했다. 조금 정신이 들고 나니 내가 정확히 어디에 있는지 궁금해졌다. 나는 지금 어디에 있는 것일까? 내 위치를 정확하게 확인할 수 있는 방법은 무엇일까?

이제 위치 센서를 통해 나의 위치를 알 수 있게 되었다. 이렇게 얻은 정보는 여러 가지 방법으로 활용할 수 있다. 예를 들어 이 정보를 다른 앱에도 전달할 수 있다. 이전까지 만들었던 앱에 다음 기능을 추가해 보자.

나의 모바일 기기에 설치되어 있는 지도 앱을 실행시켜 나의 위치 확인

#### (1) 컴포넌트 배치

다른 앱을 실행시키기 위해서는 ActivityStarter 컴포넌트가 필요하다. ActivityStarter 컴포넌트는 [Palette]-[Connectivity] 메뉴 아래 위치하고 있다.

해 보기

- 오른쪽 그림처럼 '내 위치 지도에서 확인하기' 버튼과 ActivityStarter 컴포넌트를 추가한다.



## 액티비티(Activity)

안드로이드 앱은 하나 이상의 액티비티(activity)로 이루어져 있다. 액티비티는 자기 자신 위에 다른 컴포넌트를 추가하여 동작하게 만드는 기본 뼈대라고 할 수 있다. 쉽게 말하면 앱을 구성하는 각각의 화면이 액티비티라고 생각하면 이해가 쉬울 것이다. 액티비티 스타터(ActivityStarter) 컴포넌트는 말 그대로 다른 액티비티를 실행시키는 역할을 담당하는 컴포넌트이다. 다른 액티비티를 실행시킨다는 말은 곧 다른 앱을 실행시키게 되는 것이다.

## (2) 코딩하기

해 보기

1. 아래와 같이 블록을 코딩해 보자.

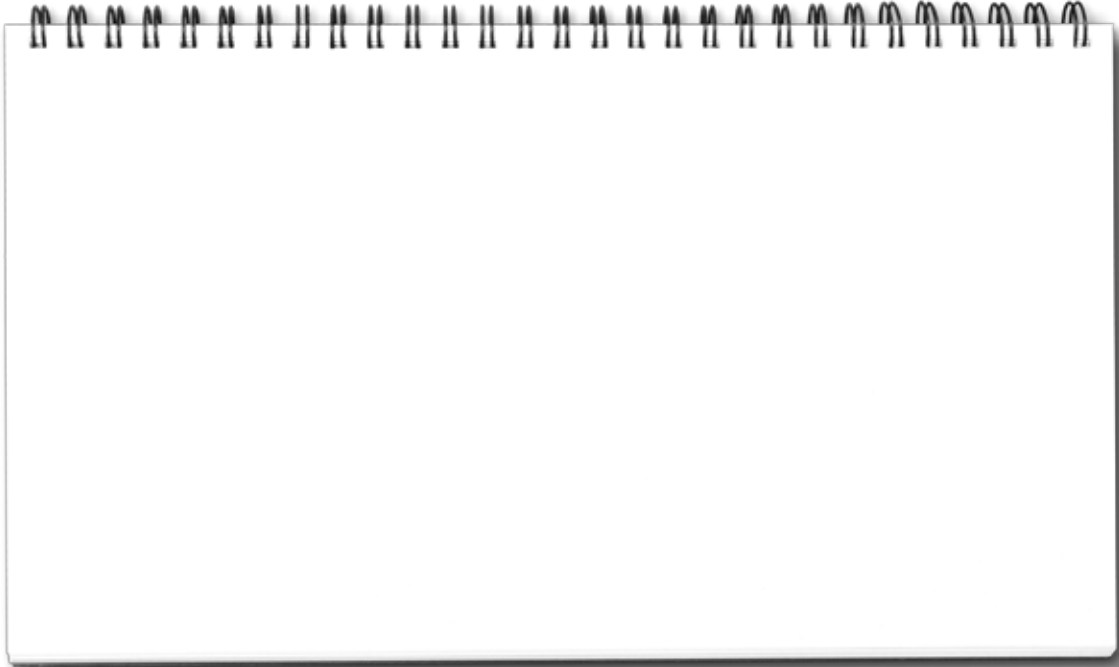


2. 앱을 빌드하여 내 기기에서 실행해 보고 버튼을 눌렀을 때 앱이 실행되는 과정을 설명해 보자.

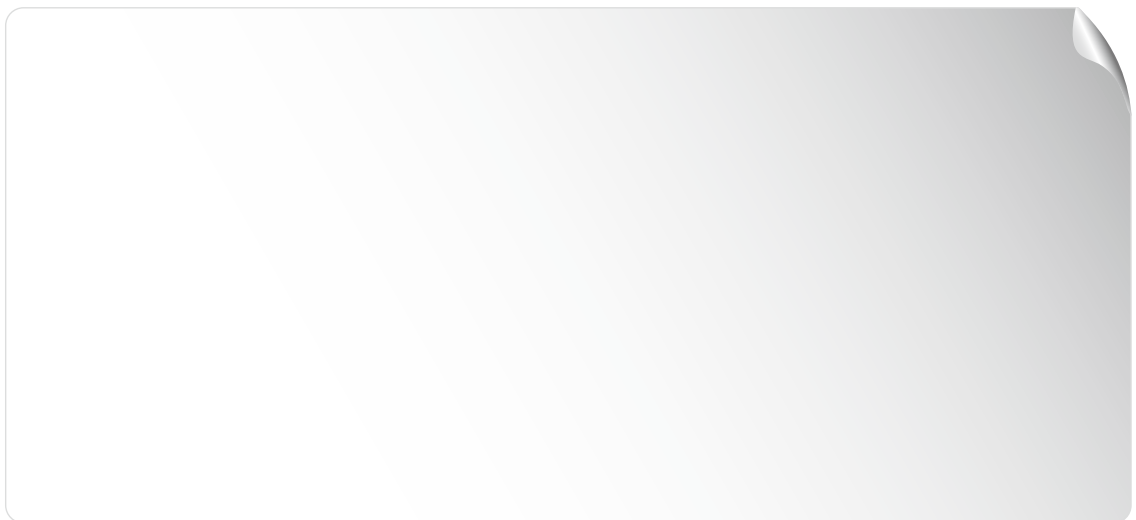
#### 4. 앱 확장하기

이 앱의 기능을 더욱 확장해보자. 어떠한 형태의 기능을 추가할 수 있는가?

(1) 앱의 디자인을 개선하기 위한 아이디어를 기록해 보자.



(2) 앱에 추가하고 싶은 기능을 기록해보고 이러한 기능이 앱 인벤토리로 실현 가능할지 프로그래밍해 보자.





## 게임 앱

### ! 학습 목표 !

1. 캔버스와 터치스크린의 상호작용에 대해 설명할 수 있다.
2. 캔버스와 이미지 스프라이트로 인터랙티브한 게임 화면을 구성할 수 있다.
3. 여러 개의 화면으로 앱을 구성할 수 있다.
4. 다양한 방식의 이벤트 제어를 통해 프로그램을 진행할 수 있다.

### 1. 모바일 기기의 입출력

기존의 컴퓨팅 기기는 일반적으로 키보드와 마우스를 통해 입력을 받고, 모니터나 스피커를 통해 프로그램의 결과를 출력하였다. 현대의 모바일 기기는 이러한 입력과 출력을 하나의 터치스크린을 통해 모두 처리할 수 있다.

이러한 터치스크린을 가장 잘 활용할 수 있는 앱 중 하나가 게임 앱이다. 게임 앱은 터치스크린에서 활용할 수 있는 다양한 방식의 입력을 활용하여 이벤트를 처리할 수 있다.

#### 생각해 보기

내가 자주 실행하는 모바일 게임에 대해 소개하고, 이 게임이 어떻게 동작하는지를 프로그래머의 관점에서 설명해 보자.





## (1) 캔버스 컴포넌트의 활용

게임 앱을 만들기 전에 먼저 터치스크린을 활용하는 방법을 알아보자. 우리는 다음과 같은 앱을 만들 것이다.

- 손가락 터치를 통해 그림을 그릴 수 있는 그림판 앱

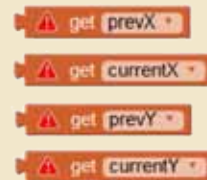
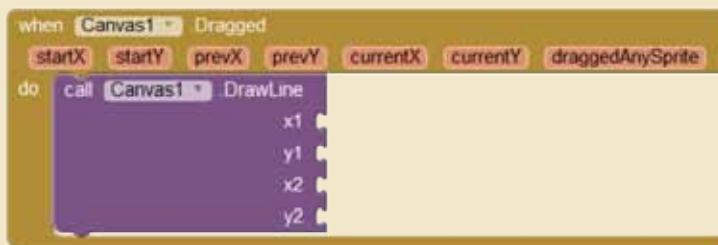
터치스크린을 활용하기 위해서는 앱 인벤터의 캔버스(Canvas) 컴포넌트를 활용한다. 캔버스가 위치한 공간에서는 다른 컴포넌트가 없어도 터치스크린의 입력을 받아들이게 된다. 캔버스 컴포넌트를 사용해서 간단한 그림판 앱을 한 번 만들어 보자.

해 보기

1. 캔버스 컴포넌트는 [Palette] - [Drawing and Animation] 메뉴 아래에 위치하고 있다. 오른쪽 그림과 같이 캔버스 컴포넌트를 배치하고 가로와 세로의 길이를 조절해 보자.



2. 아래의 블록을 활용하여 선을 그릴 수 있는 코드를 완성해 보자.



3. 아래 블록이 어떠한 역할을 담당하는지 생각해 보자.



## (2) 이미지 스프라이트 컴포넌트

캔버스 위에는 다른 컴포넌트를 배치할 수 없다. 오직 이미지 스프라이트 컴포넌트만 캔버스 컴포넌트 위에 배치하여 상호작용할 수 있다. 앞에서 만든 그림판 앱을 활용하여 다음 기능을 추가해 보자.

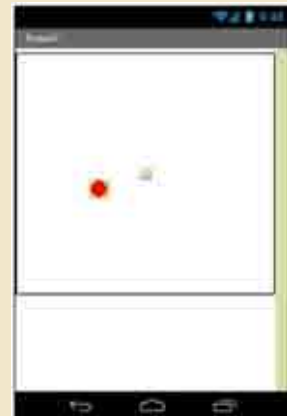
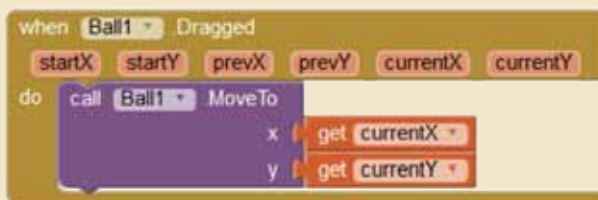
- 손가락 터치를 통해 이미지 움직이기

이미지 스프라이트(ImageSprite) 컴포넌트는 터치 입력을 통해 상호작용할 수 있는 컴포넌트이다. 스프라이트 컴포넌트와 상호작용하기 위해서는 터치를 활용하기 때문에 스프라이트 컴포넌트는 반드시 캔버스 컴포넌트 위에 배치되어야 한다.

이미지 스프라이트 컴포넌트도 [Palette] - [Drawing and Animation] 메뉴 아래에 위치하고 있다. 볼(Ball)은 기본적으로 앱 인벤터에서 제공하는 스프라이트이고, 이미지 스프라이트 컴포넌트를 사용하면 다른 모양의 스프라이트를 활용할 수 있다. 우리는 볼 컴포넌트를 사용하여 스프라이트를 동작시켜 보자.

해 보기

- 오른쪽 그림과 같이 볼 컴포넌트를 캔버스 컴포넌트 위에 위치시키고 아래 그림처럼 블록 창에서 코드를 추가해 보자. 앱을 실행시켜 앱의 동작에 대해 설명해 보자.



## 2. 공 받기 게임

캔버스 컴포넌트와 이미지 스프라이트 컴포넌트를 활용하면 사용자가 직접 조작할 수 있는 인터랙티브한 게임을 제작할 수 있다. 우리는 아주 간단한 게임을 하나 제작해 보면서 게임의 디자인 및 제작이 어떠한 방식으로 이루어지는지를 경험해 보도록 하자. 우리는 아래와 같은 게임을 제작할 것이다.

### 게임 아이템

손가락으로 블록을 움직여 공을 떨어뜨리지 않게 한다.



공은 화면 안을 돌아다닐 것이다.

위의 게임은 손가락으로 블록을 움직여 공이 떨어지지 않게 계속해서 받는 아주 간단한 형태의 게임이다. 우리가 이 게임을 플레이한다면 몇 분 되지 않아 지루하게 생각될 수도 있을 게임이지만, 이러한 간단한 게임도 게임의 진행을 위한 다양한 디자인과 제어가 포함되어 있다.

### (1) 게임 구조 설계

게임 앱을 제작하기 전에 우리는 먼저 게임이 어떤 목적을 가지고 있고, 어떤 동작을 수행할 것이며, 어떤 순서로 진행될 것인지 게임의 전체적인 구조를 설계해야 한다. 우리가 생각해야 할 부분은 게임의 화면 구성, 각 화면의 디자인, 게임의 진행 상황이 있다.

#### 1) 게임의 화면 구성과 디자인

앞에서 우리가 만들어 본 앱은 하나의 화면만을 사용하는 앱이었지만 실제로 우리가 사용하고 있는 여러 앱은 여러 개의 화면을 사용하고 있다.

게임 또한 여러 화면을 활용하여 게임의 진행을 할 수 있다. 우리는 아래 그림과 같이 화면을 구성하여 게임을 진행하도록 해보자.

① 가장 먼저 첫 화면은 게임의 시작을 알리는 화면이다. 이 화면은 게임의 로고와 게임 시작과 종료 버튼, 게임 설명 버튼을 배치할 것이다. 각 버튼을 누르면 해당 명령이 실행된다.

② 게임 화면은 실제 게임이 진행되는 화면이다. 이 화면에서 플레이어는 블록을 이동시키며 공을 떨어뜨리지 않는 게임을 진행하게 된다.

각 화면이 어떠한 역할을 할 것인지를 결정하였다면 각 화면에 들어갈 멀티미디어 자료나 컴포넌트의 배치를 결정한다.

첫 화면



게임 화면



## 2) 게임 동작 설계하기

각 화면에서 사용자가 앱과 상호작용할 때 발생하는 이벤트에 대한 처리를 계획해야 한다. 또한 게임 안에서 자동적으로 진행되어야 하는 다양한 동작에 대한 처리도 계획해야 한다.

해 보기

▶ 다음 빈칸에 각 화면에서 처리해야 하는 다양한 동작을 기록해 보자.

첫 화면	<ul style="list-style-type: none"> <li>• 시작하기 버튼을 누르면 게임을 시작하는 화면으로 전환한다.</li> </ul>
게임 화면	<ul style="list-style-type: none"> <li>• 게임이 시작되면 볼은 계속해서 캔버스 위를 돌아다녀야 한다.</li> </ul>

### 3. 게임 제작하기

앞에서 게임 앱을 어떻게 디자인하고 제작할 것인지를 계획하였다면 계획에 맞게 차례대로 앱을 제작하는 과정을 거치게 된다. 제작 순서는 앱 인벤터의 앱 개발 순서대로 화면 설계, 블록 코딩의 순서로 이루어지게 된다. 제작하는 중간 과정에서도 계획한 대로 앱이 제작되고 있는지를 계속해서 점검하는 것이 필요하다.

#### (1) 화면 추가 및 화면 설계

앱 인벤터에서는 기본적으로 한 개의 화면을 제공한다. 우리는 필요에 따라 새로운 화면을 추가하여 사용할 수 있다. 각 화면은 개별적으로 컴포넌트를 배치하여 설계하여야 하고, 블록 코딩 또한 개별적으로 진행된다.

해 보기

1. 새로운 프로젝트를 하나 생성하고(MovingBlocks), 디자이너 창에서 첫 화면을 디자인해 보자.



2. 아래 그림을 참고하여 게임 실행 화면(PlayScreen)을 추가하고, 캔버스와 이미지 스프라이트(볼, 블록)를 배치하여 실제 게임이 진행될 수 있도록 디자인해 보자.



## (2) 게임 진행하기

프로그램을 진행하는 방식은 다양하지만 대부분의 모바일 앱은 사용자의 외부 입력을 필요로 하고, 외부 입력의 대부분은 터치스크린을 이용한 입력이다. 앱은 사용자의 터치를 인식하면 이를 이벤트(event)로 처리하게 된다. 각 단계마다 나타나는 다양한 이벤트를 통해 게임이 진행될 수 있도록 해보자.

### 1) 게임 시작하기

앱을 실행했을 때 가장 처음 상태는 어떠한가?



해 보기

🔴 위에서 설명한 첫 단계를 앱 인벤터의 동작으로 설명해 보고 실제로 코딩해 보자.

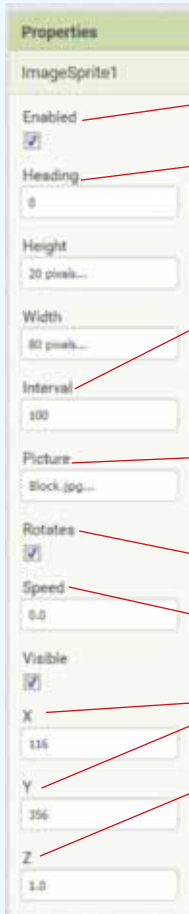
- 게임의 첫 화면이 나타난다.
- 게임 시작하기 버튼을 누르면 게임을 시작한다.

## 2) 공 움직임 처리하기

앞에서 우리는 게임 화면의 개체들이 어떻게 움직여야 하는지를 설계하였다. 이미지 스프라이트의 움직임과 관련된 내용은 다음과 같다.

- 게임이 시작되면 볼은 계속해서 캔버스 위를 돌아다녀야 한다.
- 볼이 벽이나 블록을 만나면 튕겨 나온다.

우리가 이미지 스프라이트를 캔버스 위에서 움직이기 위해서 알아야 하는 몇 가지 속성이 있다.



### 이미지 스프라이트의 속성

- **Enable** : 체크가 되어 있을 때 사용 가능하다.
- **Heading** : 스프라이트가 이동할 방향을 말한다. 숫자는 양의 x축에 대한 각도를 말한다. 옆의 예에서 보는 0은 오른쪽 방향을 나타낸다.
- **Interval** : 위치 정보를 갱신하는 시간을 말한다. 단위는 밀리초(milliseconds)를 사용한다.
- **Picture** : 이미지 스프라이트에 이미지를 불러와 사용할 때 사용한다. 나의 컴퓨터에 있는 이미지를 업로드하여 사용할 수 있다.
- **Rotate** : Heading의 방향에 따라 이미지를 회전할 것인지를 결정한다.
- **Speed** : 이동 속도를 결정한다.
- **X, Y** : 평면에서의 X, Y 좌표를 나타낸다.
- **Z** : 여러 이미지 스프라이트가 겹쳐져 있을 때의 위치를 나타낸다. Z 값이 큰 이미지 스프라이트가 위쪽으로 나타나게 된다.

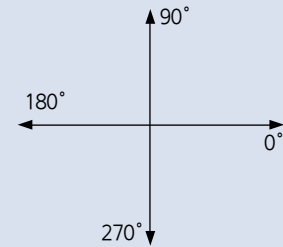
일단 게임이 시작되면 PlayScreen 화면에 있는 공은 자동적으로 움직여야 한다. 따라서 공의 움직임은 공의 속도(speed)와 방향(heading)을 조정하여 조절해야 한다.

- 다음 블록을 수정하여 공을 움직이게 해보자.



### 방향(Heading)

앱 인벤터에서 이동하는 개체는 방향이라는 속성을 가진다. 방향 속성은 0 부터 360까지의 값을 가지는데 개체가 3시 방향으로 움직일 때 개체의 방향이 0도이다. 12시 방향은 90°, 9시 방향은 180°, 6시 방향으로 움직이게 하기 위해서는 270°로 설정하면 된다.



위의 코드를 통해 공이 움직이게 되었다. 하지만 문제가 발생할 것이다. 바로 공이 캔버스의 가장자리에 닿으면 더 이상 움직이지 못하게 되는 문제이다. 공이 벽에 닿게 되면 튕겨 나오는 코드를 작성해 보자.

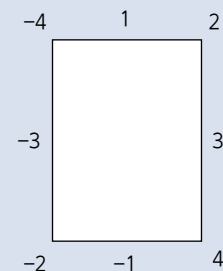
- 오른쪽 블록을 활용하여 공이 벽에 닿게 되면 튕겨 나오도록 코드를 완성해 보자.



### 가장자리(edge)

캔버스 또한 고유한 속성 몇 가지를 가지고 있고 가장자리는 그 중 하나이다. 가장자리는 캔버스를 구성하는 사각형의 각 변과 각 꼭지점을 말하고, 앱 인벤터에서는 이를 아래와 같이 각각의 번호로 구별하게 된다.

북쪽변(1), 북동꼭지점(2), 동쪽변(3), 남동꼭지점(4), 남쪽변(-1), 남서꼭지점(-2), 서쪽변(-3), 북서꼭지점(-4)





### (3) 블록 움직임 처리하기

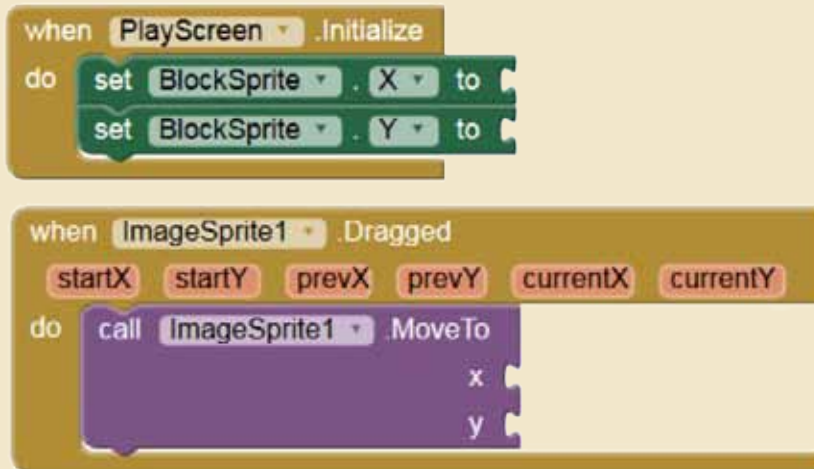
블록의 움직임은 공보다는 간단하다. 우리가 계획한 움직임 중에 블록과 관련된 것은 아래와 같다.

- 블록은 좌우로만 움직이고 사용자가 터치를 통해 이동시켜야 한다.
- 볼이 벽이나 블록을 만나면 튕겨 나온다.

블록은 우리가 손으로 움직일 수 있는 이미지 스프라이트이다. 블록에 대해 설정해 줄 부분은 게임이 시작되었을 때의 초기 위치와 우리가 손으로 드래그했을 때의 움직임이다.

해 보기

1. 아래 블록을 참고하여 블록의 초기 위치를 설정하고, 블록을 드래그했을 때 왼쪽과 오른쪽으로 움직이도록 블록을 완성해 보자.



2. 앞에서 작성한 코드를 참고하여 공이 블록에 닿았을 때 바닥에 닿은 것처럼 튕겨 나오도록 오른쪽 블록을 완성해 보자.



#### (4) 게임 종료 처리

게임은 언제 끝나게 될까? 게임이 끝날 수 있는 조건을 생각해 보자.



게임을 끝낼 수 있는 조건은 다양하게 있을 수 있다. 어떠한 조건이든지 그 조건을 달성하게 되면 게임을 종료시키게 된다. 몇 가지 해보기를 통해 게임을 종료해 보자.

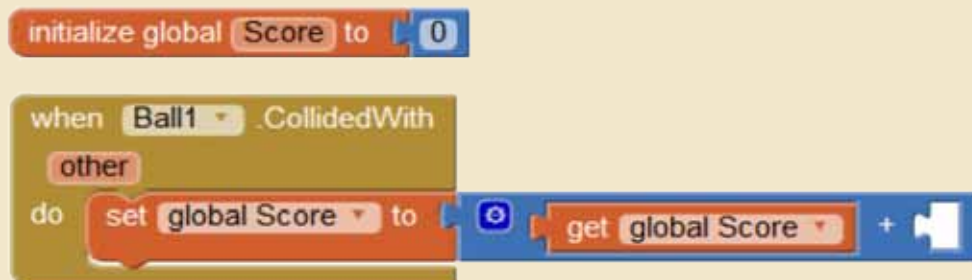
해 보기

##### 1. 실패 조건 - 공이 바닥에 닿는다.

공이 블록에 닿지 않고 바닥에 닿으면 게임에서 진 것으로 판단할 수 있다. 우리는 앞에서 공의 움직임을 설정할 때 바닥에서도 튕겨 나오게 만들었다. 이 코드를 변경하여 공이 바닥에 닿으면 공의 움직임을 멈추고, 게임을 종료하도록 만들어 보자.

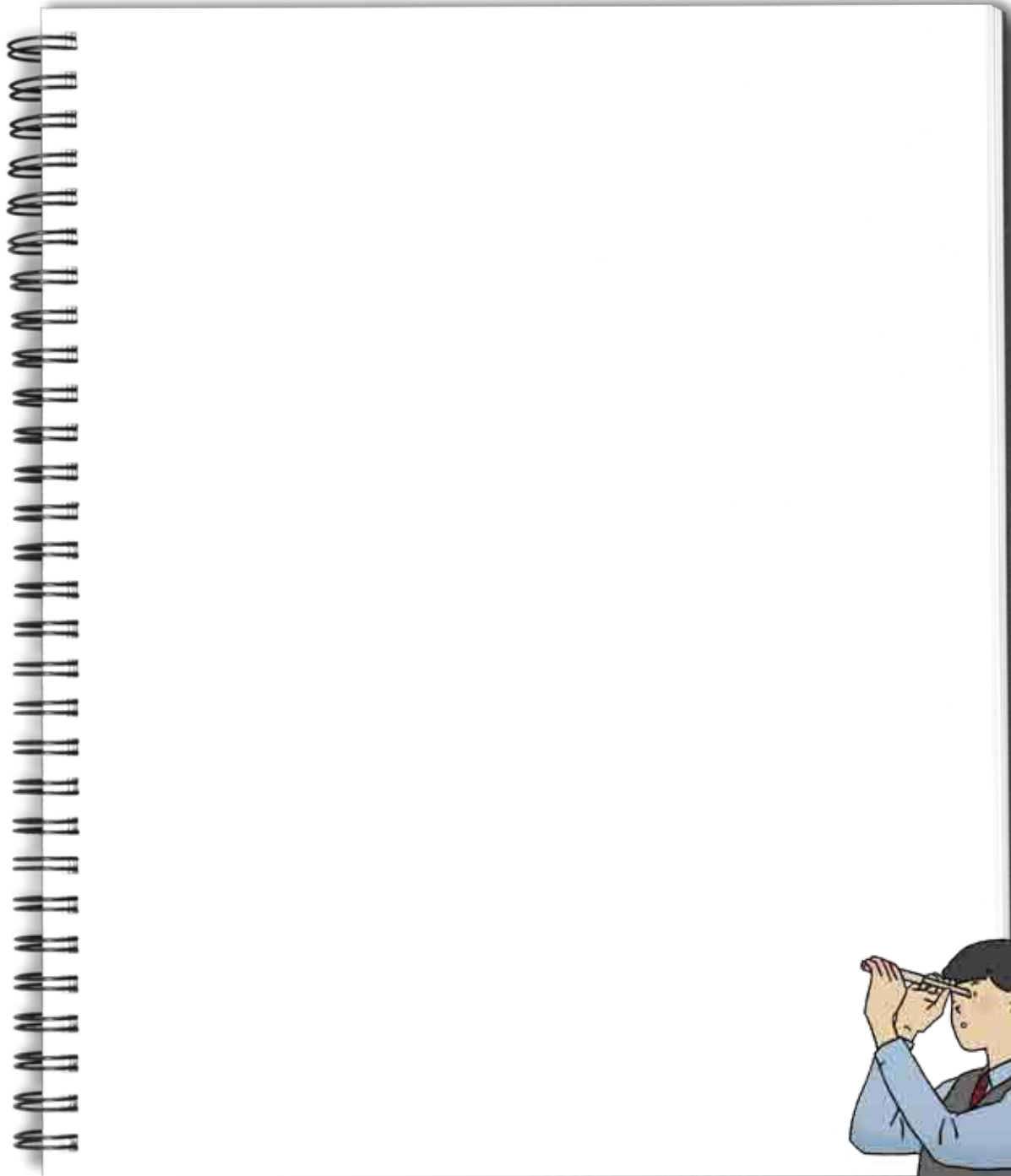
##### 2. 승리 조건 - 공이 블록에 10번 닿으면 게임에서 승리한다.

승리 조건은 다양하게 정할 수 있지만 여기에서는 블록에 닿은 횟수로 결정해보자. 블록에 닿은 횟수를 기록할 수 있는 변수를 하나 만들어서 이 변수가 10이 되었을 때 게임을 멈추고 승리화면을 표시해 보자. 아래의 코드를 활용하여 코딩해 보자.

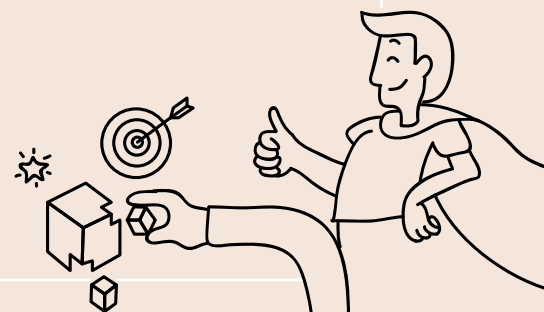


### 3. 앱 확장하기

앞에서 게임의 가장 기본적인 뼈대를 완성하였다. 이 게임을 보다 흥미롭게 만들기 위해서는 어떠한 동작들이 추가되는 것이 좋겠는가?



- ① 앱 인벤터는 안드로이드 운영체제에서 실행되는 모바일 앱을 제작하기 위해 사용되는 프로그래밍 언어이다. 앱 인벤터는 웹 브라우저에서 클라우드 컴퓨팅 방식으로 실행되기 때문에 인터넷이 연결된 환경에서 사용할 수 있다.
- ② 앱 인벤터의 프로그램 개발 환경은 디자이너 창과 블록 창으로 구성되어 있다. 디자이너 창에서는 앱의 외형적인 설계와 필요한 컴포넌트를 배치하는 작업을 진행한다. 블록 창에서는 배치한 컴포넌트와 관련된 블록을 배치하여 내가 원하는 명령을 수행하도록 한다.
- ③ 디자이너 창에서 배치한 컴포넌트는 속성을 변경하여 크기나 상태를 변화시킬 수 있고, 컴변수나 컴포넌트 속성값을 서로 전달할 수도 있다.
- ④ 완성된 앱 인벤터 프로젝트는 실제 모바일 기기에 설치 가능한 파일로 빌드하여 실제 기기에서 구동시킬 수 있다.
- ⑤ 데이터베이스는 대규모의 데이터 집합을 관리하기에 편리하도록 만든 시스템이다. 앱 인벤터에서는 TinyDB라는 데이터베이스를 사용하는데, TinyDB에서는 태그를 활용하여 자료를 저장하고 불러올 수 있도록 도와준다.
- ⑥ 리스트뷰는 여러 값이 묶여 있는 리스트 자료를 한 눈에 보기 쉽게 해주는 컴포넌트로 사용자 인터페이스를 간단하게 만드는 데 사용한다.
- ⑦ 위치 센서는 GPS 모듈이나 와이파이, 이동통신망에서 제공하는 위치 정보를 수신하여 활용할 수 있는 센서이다. 위치 센서가 제공하는 정보는 위도, 경도, 고도, 주소가 있다.
- ⑧ 통신과 관련된 컴포넌트는 전화 컴포넌트와 문자 컴포넌트가 있다. 전화번호와 메시지 등을 미리 지정해 두면 터치 한 번으로 전화를 걸거나 문자를 보낼 수 있다.
- ⑨ ActivityStarter 컴포넌트는 하나의 앱에서 다른 앱을 실행시킬 때 사용하는 컴포넌트이다.



## 집필위원

---

이영준 한국교원대학교 컴퓨터교육과 교수(집필 책임)  
최정원 석정중학교 교사  
안상진 동래중학교 교사  
전준호 평촌경영고등학교 교사

## 기획위원

---

서석진 미래창조과학부 소프트웨어정책관  
최미정 미래창조과학부 소프트웨어교육혁신팀장  
손종걸 미래창조과학부 소프트웨어교육혁신팀 사무관  
장영록 한국과학창의재단 과학수학교육단장  
이환철 한국과학창의재단 수학컴퓨팅교육실장  
이현숙 한국과학창의재단 수학컴퓨팅교육실 연구원

## 발행 협조

---

편집·디자인 정은영  
사진·삽화 김윤경

## 고등학교 소프트웨어와 친해지기

2016. 01. 15. 초판 발행  
저작권자 : 한국과학창의재단  
발행자 : 한국과학창의재단  
인쇄 및 보급처 : 주)서울교과서[경기도 파주시 회동길 325-22 세화빌딩 3층]

정가 원

- 저작물 보상금 지급 : (사)한국복제전송저작권협회(02-2608-2800, [www.korra.kr](http://www.korra.kr))
- 교과서 관련 문의사항 : 교과서민원바로처리센터(1566-857, <http://www.textbook114.com> 또는 <http://www.교과서114.com>)
- 구입 관련 문의 : (주)서울교과서(전화 : 02-322-1350, 전송 : 02-322-6913, [www.seoulbooks.co.kr](http://www.seoulbooks.co.kr))





