

# 앱인벤터, ESP32로 사물인터넷 디자인하기

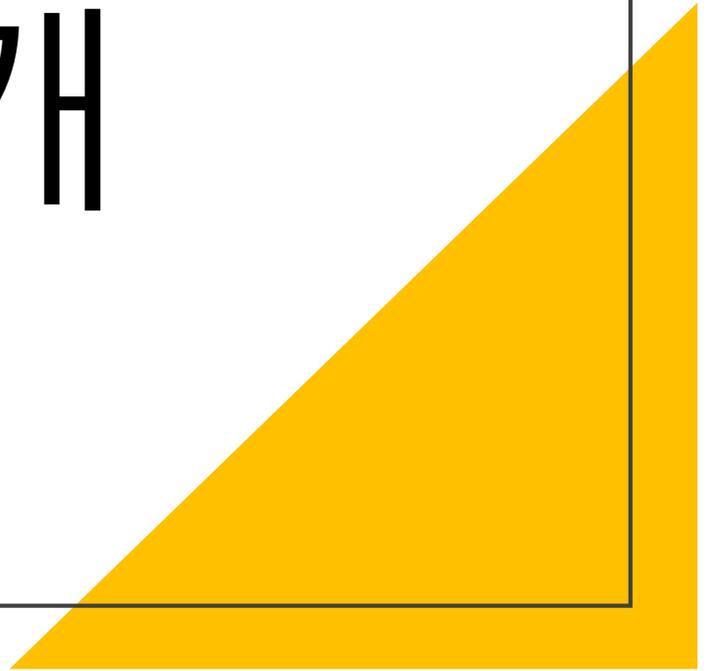
---

코딩거점학교

# 차례

1. 앱 인벤터 소개
2. 기본적인 앱 만들기
3. 게임 앱 만들기
4. 번역 앱 만들기
5. 영단어 학습 앱 만들기
6. 마스크 분류 앱 만들기

# 1. 앱 인벤터 소개



# 스마트폰 앱?

- APP = Application  
= Application Software  
(응용 소프트웨어)
- 스마트폰에서 실행되는 모든 프로그램



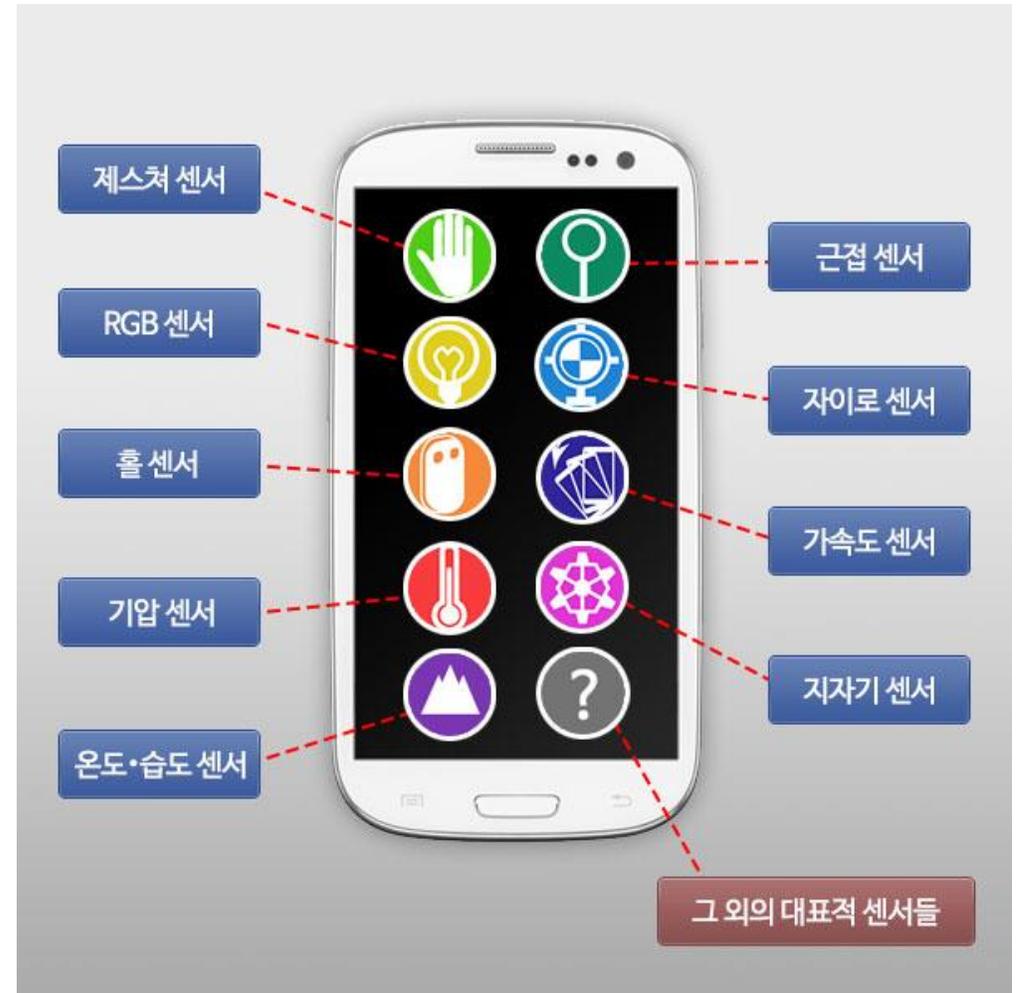
# 스마트폰 운영체제?

---

- OS = Operating System = System Software (시스템 소프트웨어)
- 사용자가 스마트폰을 쉽게 다룰 수 있도록 해주는 소프트웨어

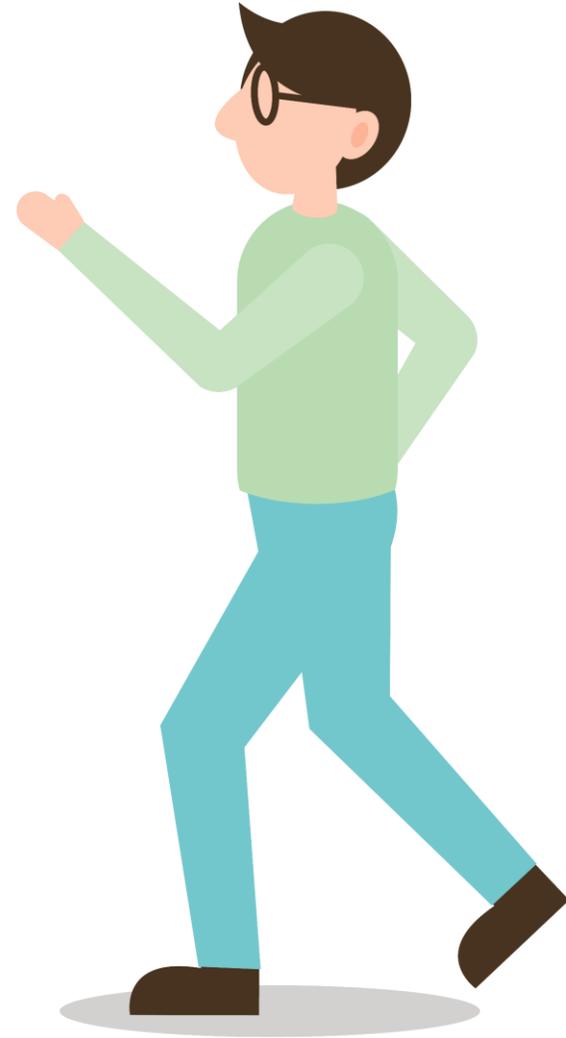


# 스마트폰의 다양한 센서들



출처: 에듀넷

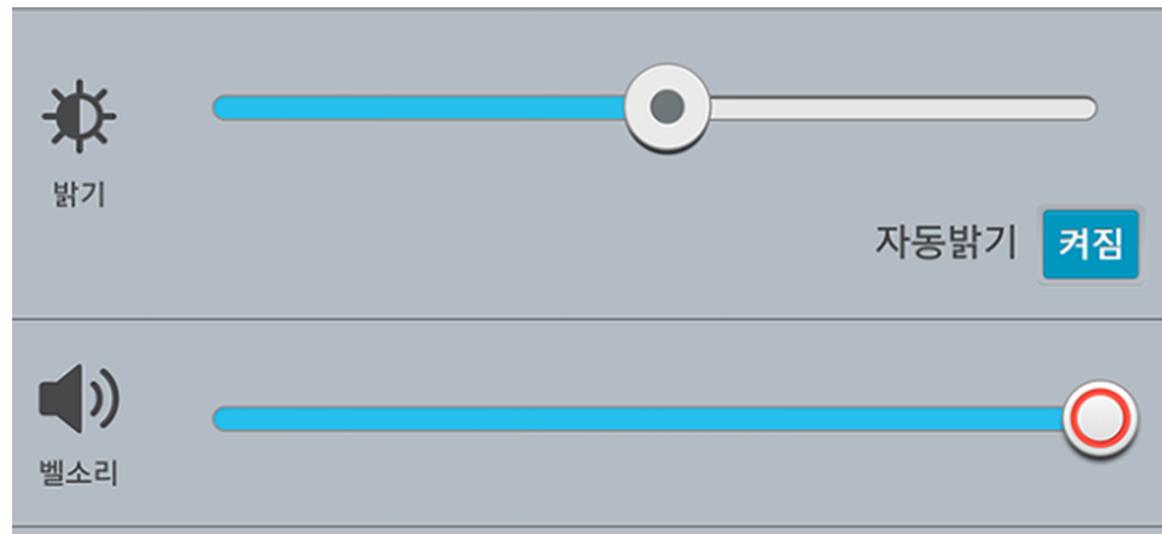
# 가속도 센서



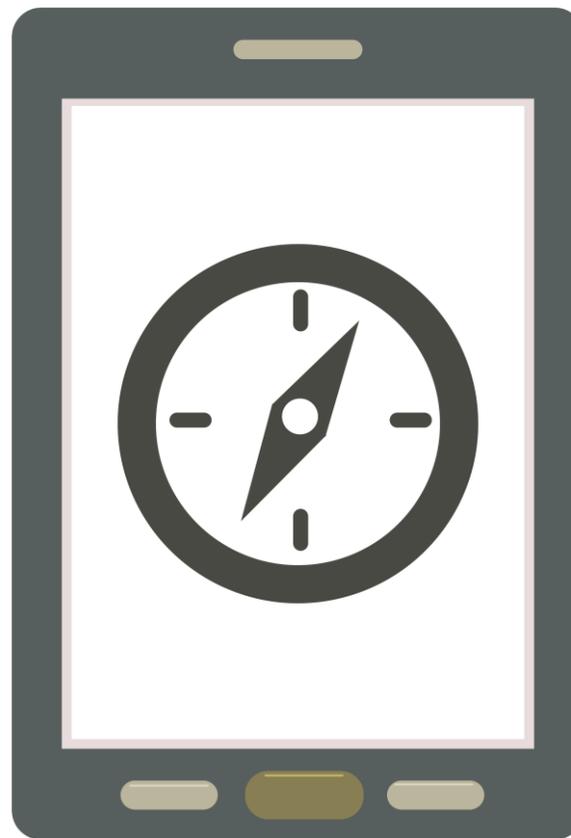
# 근접 센서



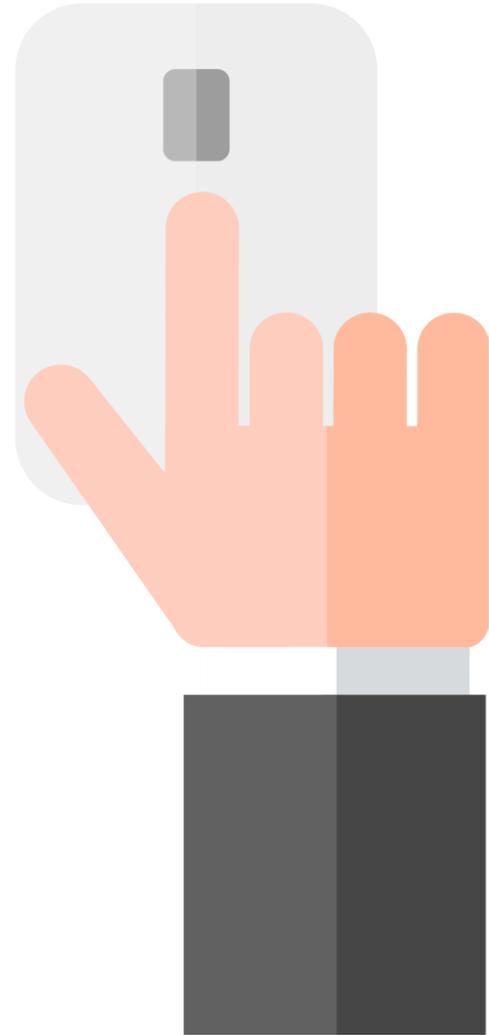
# 조도 센서



# 지자기 센서



# 지문인식 센서



# 스마트폰 운영체제

---



android





# 스마트폰 앱 개발 환경

앱 인벤터는 안드로이드  
앱 개발 전용 툴입니다.

아이폰은 Swift라는 개발 환경이  
따로 있습니다.



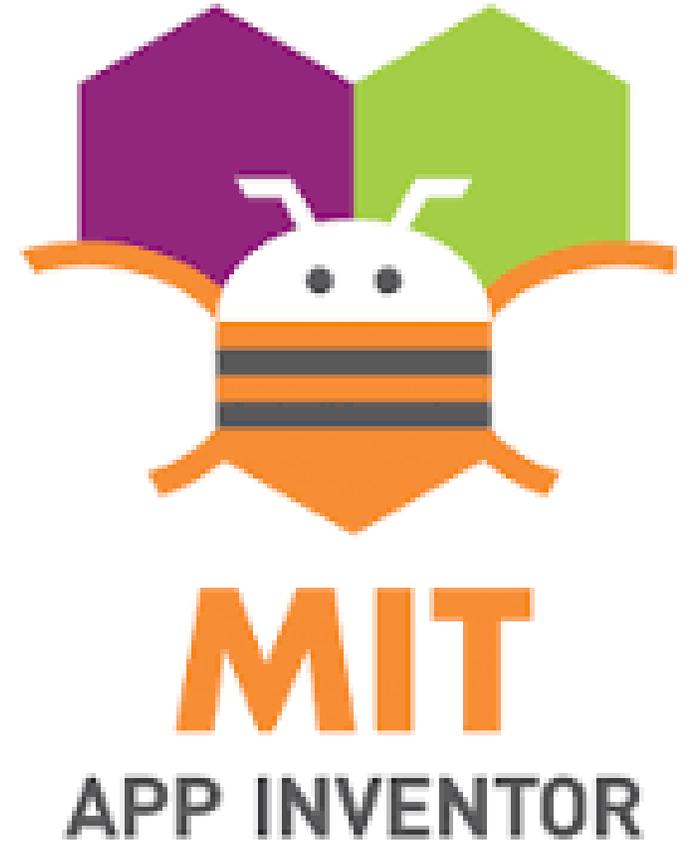
**MIT**  
APP INVENTOR



# 앱 인벤터란?

---

- 구글과 MIT가 공동으로 개발을 주도하고 MIT에서 웹(web) 서비스로 제공되고 있는 안드로이드 웹 개발 도구



# 안드로이드 앱 개발 과정

앱인벤터를  
통한 앱 개발

APK 파일 생성

Google Play  
Store에 앱 등록

# 앱 인벤터는 블록 코딩 툴입니다.

---

- 엔트리와 스크래치를 경험해보았다면 빠르게 적응할 수 있습니다.

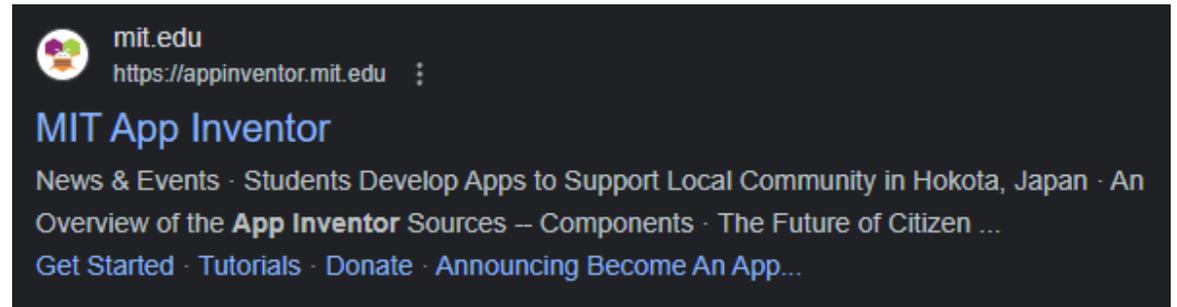


# 앱 인벤터 사용을 위한 준비물



구글 검색창에 '앱 인벤터' 검색

# 앱 인벤터 사이트 접속하기



# Create Apps 클릭



Create Apps!

About

Educators

News

Resources

Blogs

Donate

ENHANCED BY GOC



Anyone Can Create iPhone and Android  
Apps with Global Impact

Get Started



# 구글 로그인

---

 Google 계정으로 로그인

로그인

[App Inventor Authentication](#)(으)로 이동

[이메일을 잊으셨나요?](#)

[계정 만들기](#) [다음](#)



join " MIT App Inventor "

" Release "

nb187

**Release nb187 Out**  
[Read About It](#)

Got an Android phone or tablet? Find out how to  
[Set up and connect an Android device.](#)

Continue

Do Not Show Again

GET STARTED WITH SOME TUTORIALS

HELLO PURR



HelloPurr is a simple app that you can build in a very short time. You create a button that has a picture of a cat on it, and then program the button so that when it is clicked a "meow" sound plays.

GO TO TUTORIAL

TALK TO ME



Text to Speech is surprisingly fun. Find out for yourself with this starter app that talks.

GO TO TUTORIAL

TR

Quickly translate English to other languages too!) You can build an app that could act as a translator for people who need a little extra help in situations. Inspired by [this app](#) to be a Translator.

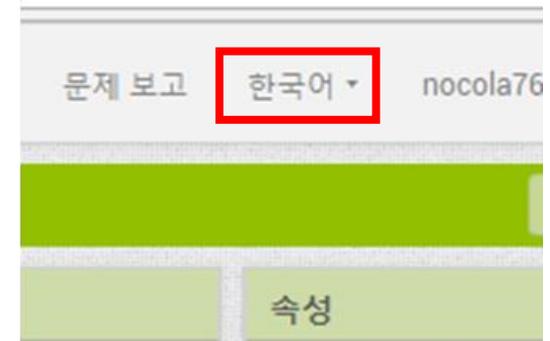
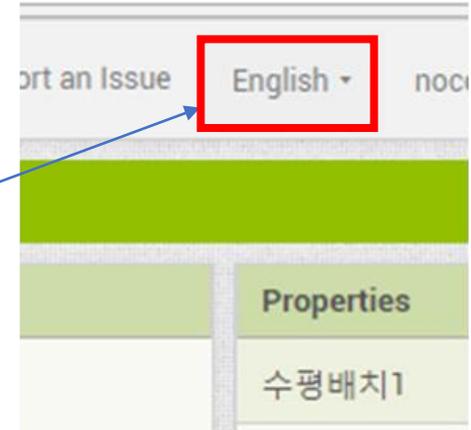
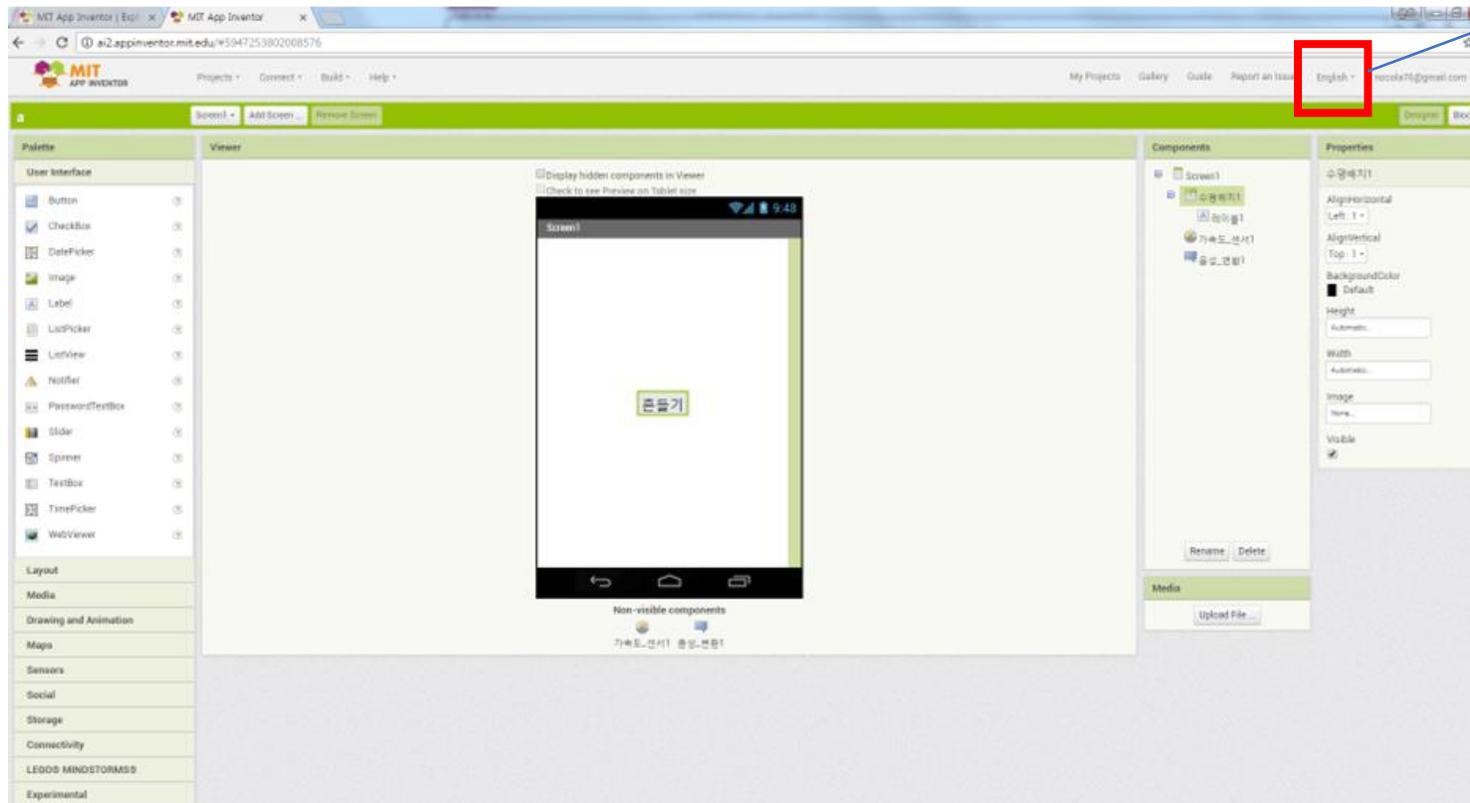
GO

START A BLANK PROJECT

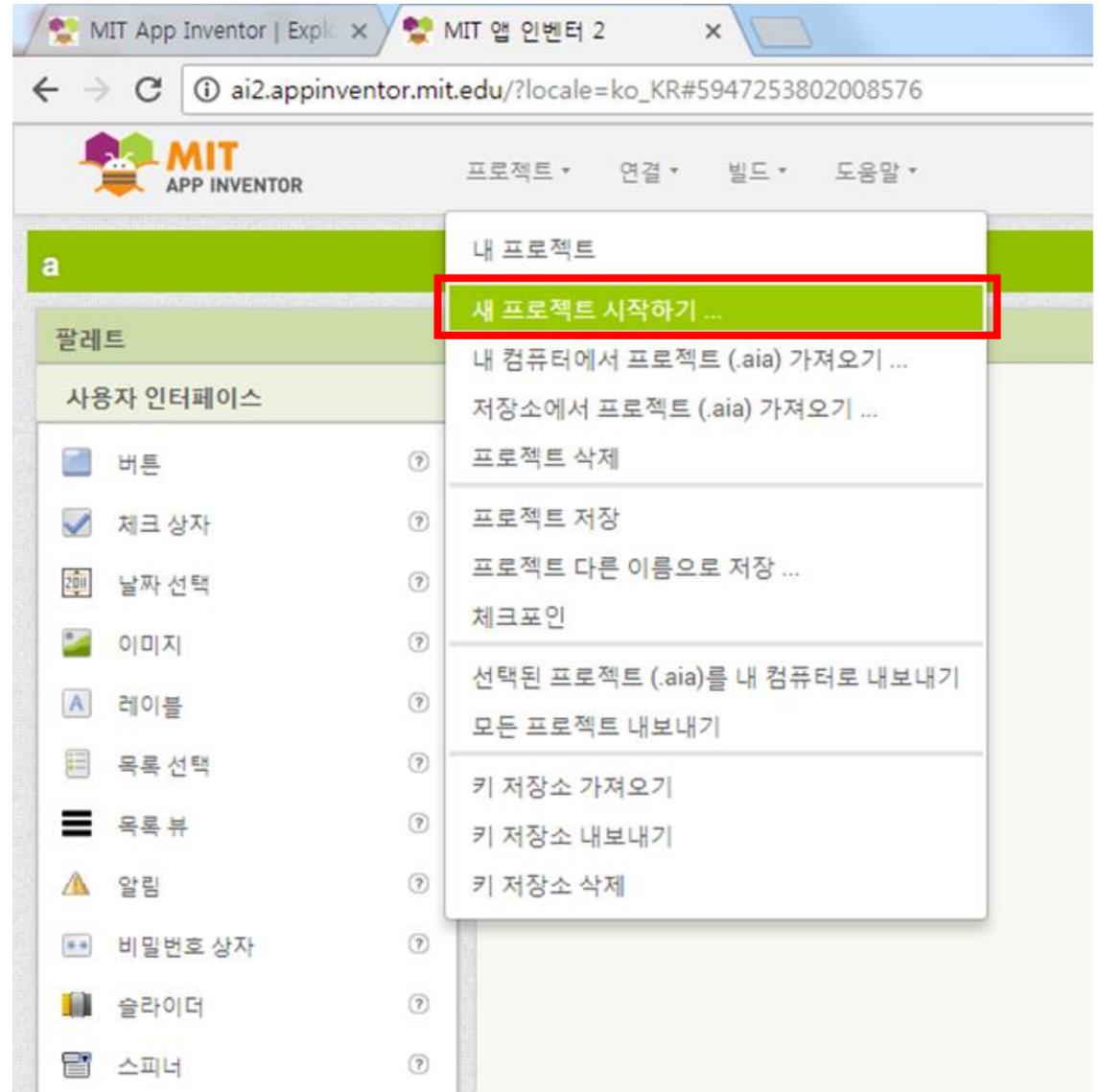
CLOSE

다음 절차에 따라 진행

# 한국어로 변경



# 새 프로젝트 시작하기



# 프로젝트 이름 정하기

새로운 앱인벤터 프로젝트 만들기

프로젝트 이름:

talk

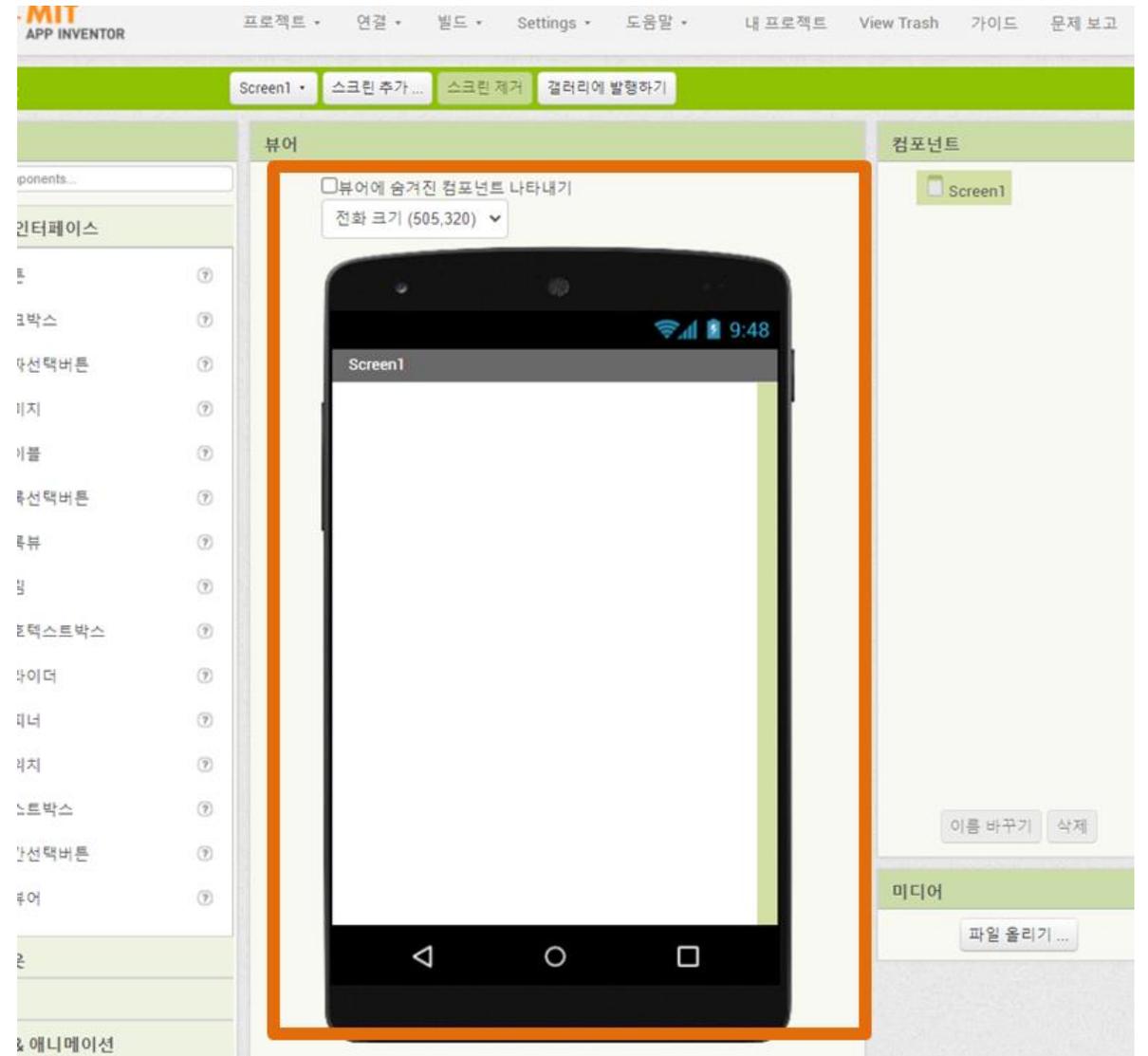
# 팔레트

- 컴포넌트를 모아 놓은 곳
- 쓰고자 하는 컴포넌트를 찾아서 뷰어에 드래그&드롭



# 뷰어

- 컴포넌트를 옮겨다 놓는 곳
- 실제 앱이 휴대폰에서 어떻게 보이는지 확인



# 컴포넌트

- 팔레트에서 가져온 것들을 모아 놓은 공간
- 각자의 구조를 확인 가능
- 이름 변경 및 삭제 가능





# 속성

색, 크기, 간격 등  
컴포넌트의 속성 변경

기본적으로 스크린1이며 스크린 추가를 통해 여러 개의 스크린을 만들 수 있음

프로젝트 이름



현재 화면이 '디자이너'이며 '블록'을 눌러 코딩할 수 있는 화면으로 전환 가능하다

MIT APP INVENTOR

math7

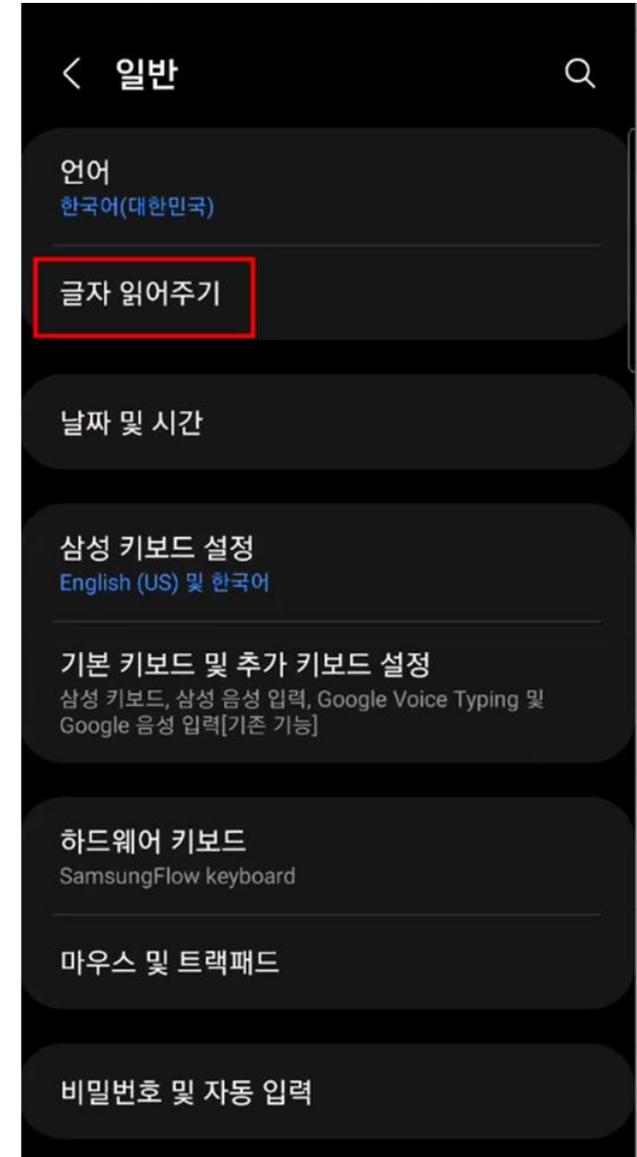
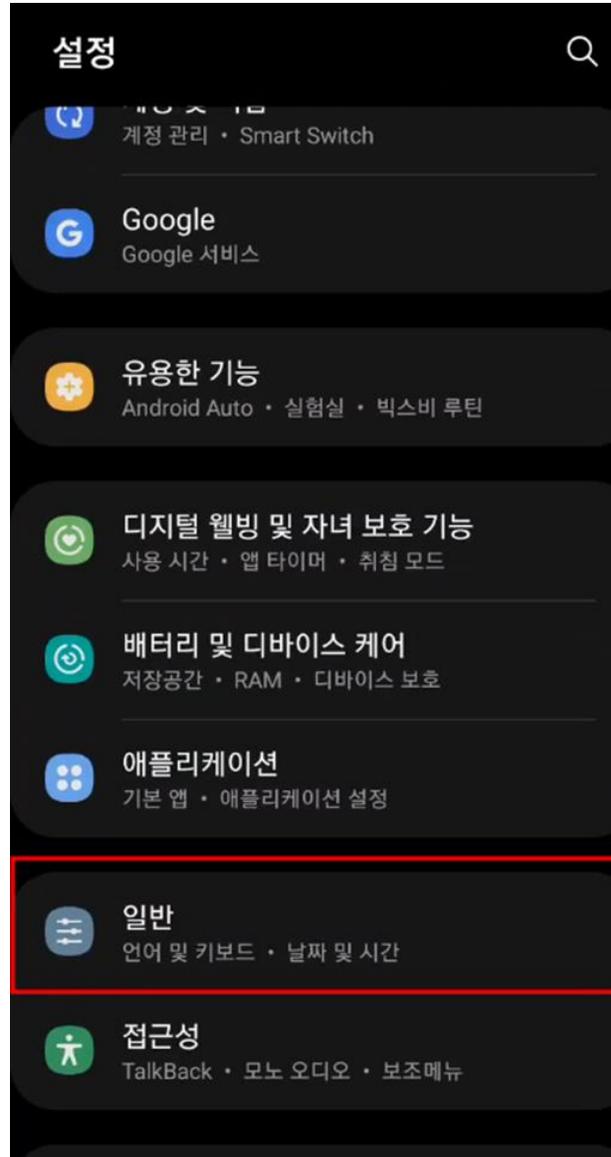
1단계 - 앱 디자인

MIT APP INVENTOR

math7

2단계 - 블록 코딩

# 언어팩 설치하기 (갤럭시 S6 Lite)



## < 글자 읽어주기

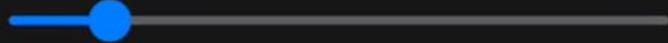
기본 엔진  
삼성 TTS 엔진



언어  
시스템 언어 사용

읽어주는 속도

느리게 빠르게



목소리 톤

낮음 높음



재생

초기화

## < 삼성 TTS 엔진 설정

음성 데이터 설치

버전

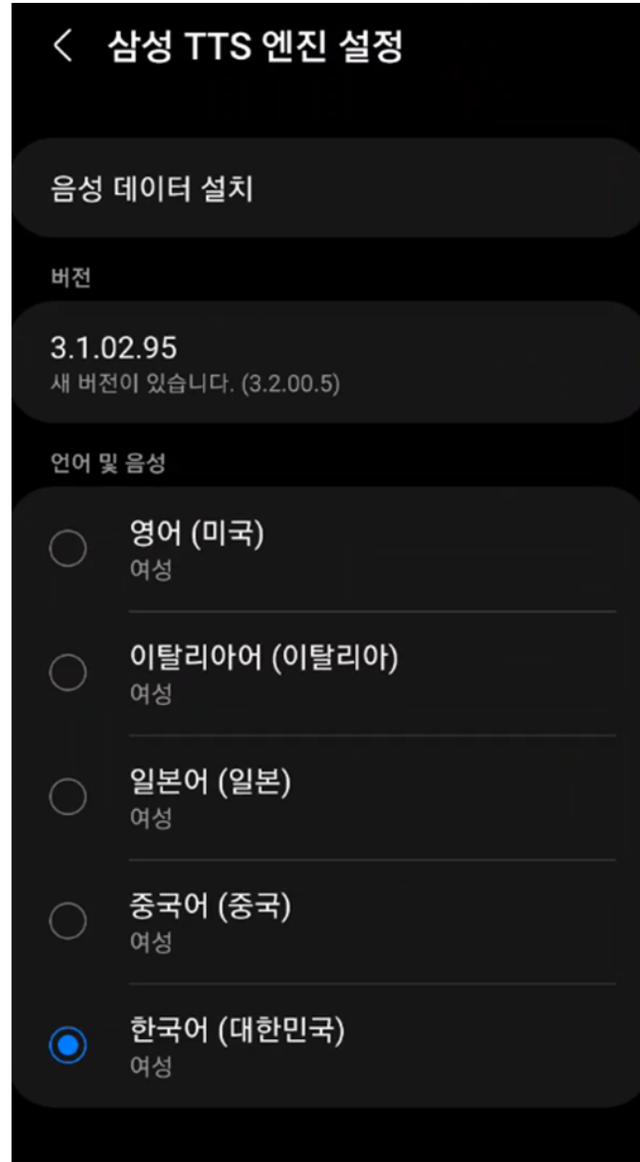
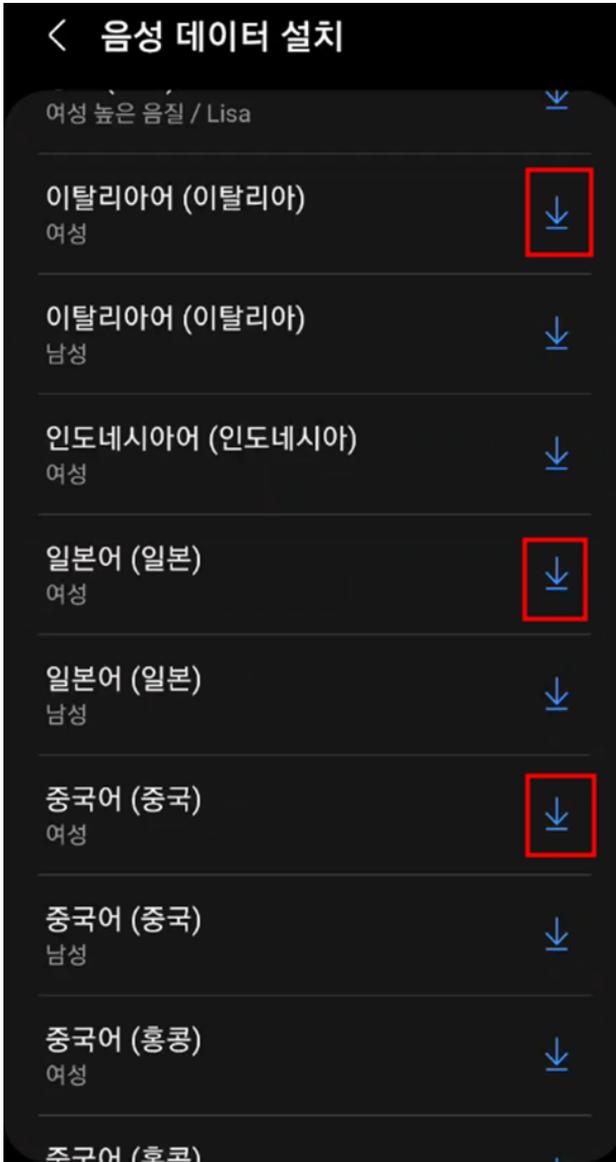
3.1.02.95

새 버전이 있습니다. (3.2.00.5)

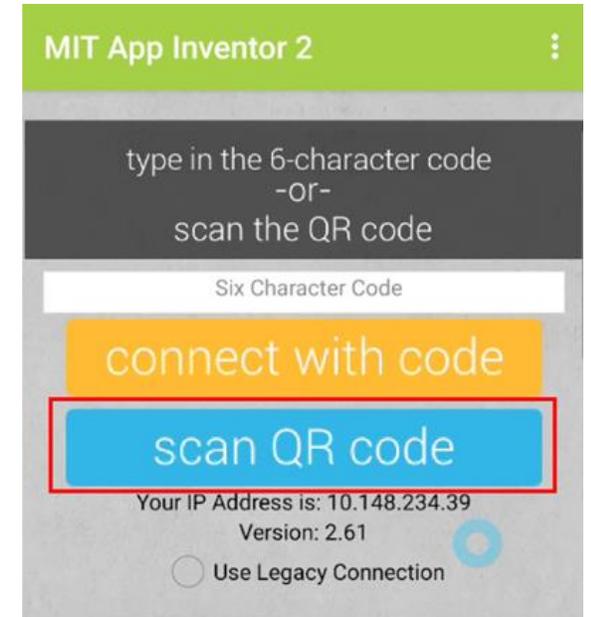
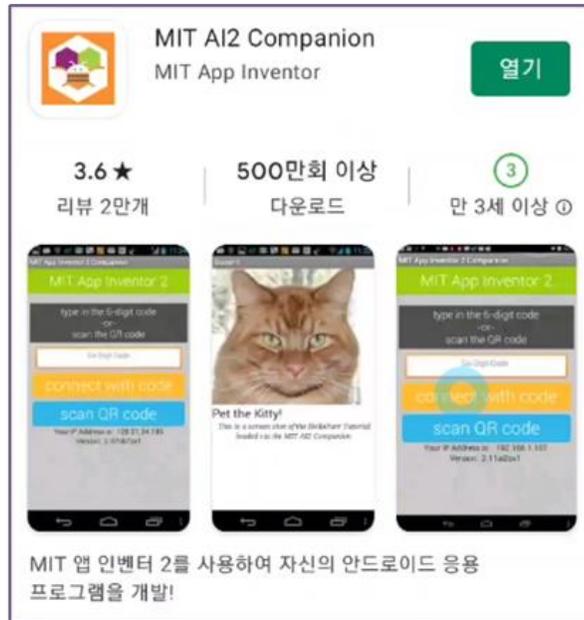
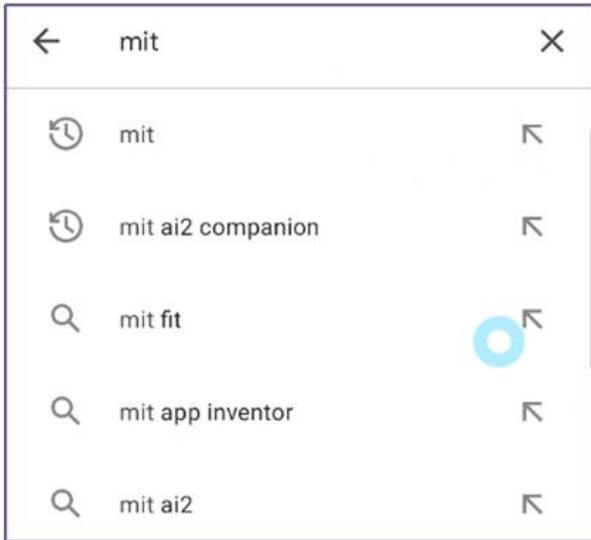
언어 및 음성

영어 (미국)  
여성

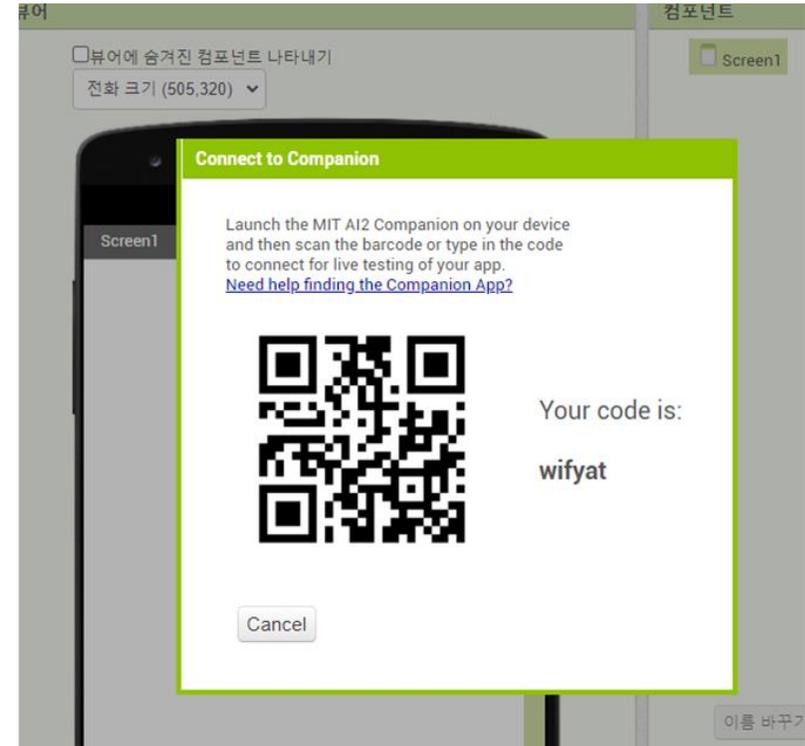
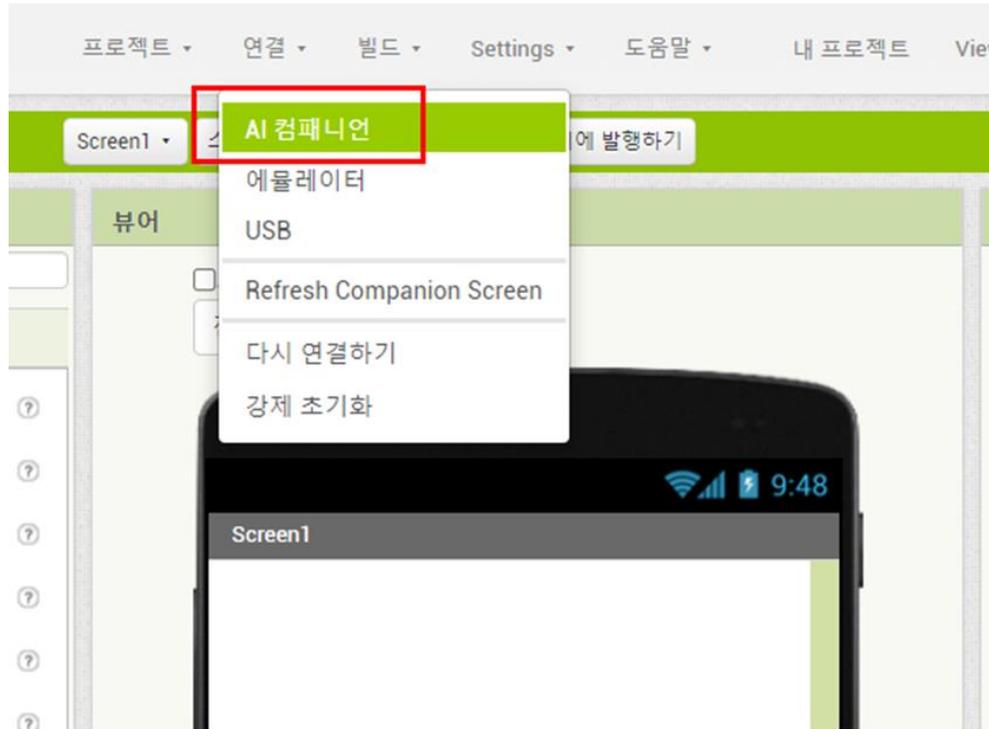
한국어 (대한민국)  
여성



# 'MIT AI2 Companion' 설치



# 스마트폰과 동기화



# 이벤트 핸들러

---

앱을 작동시키는 조건  
(버튼, 흔들기, 시간 흐름 등)

---

# 앱 인벤터 기본

**버튼**  
→ 대표적인 이벤트  
핸들러

Screen1 속에  
넣는다.

Screen1 : 부모  
버튼1 : 자식





# 앱 인벤터 기본

Screen1

버튼1 의 텍스트

스마트폰 앱에서도 실시간으로 동일하게 바뀌는 걸 확인할 수 있다.

# 앱 인벤터 기본

부모인 Screen1에서 수평정렬을 '가운데'로 하면 모든 자식들이 가운데로 정렬된다.



# 앱 인벤터 기본



컴포넌트의 색깔, 글자,  
모양 등을 바꾸고 싶을  
때는 속성에서 바꾸면 됨

# 앱 인벤터 기본



버튼을 눌렀을 때 어떤 변화(이벤트)를 주고 싶다면 '블록' 으로 넘어감

# 앱 인벤터 기본



기본적으로 제공되는 블록들

디자이너에서 활용한 컴포넌트

# 앱 인벤터 기본



클릭하거나 드래그한다.

# 앱 인벤터 기본



버튼을 클릭하면  
이 안에 넣을 블록들이 실행된다

An orange arrow points from the '실행' area of the event listener configuration to this text box. The text box has a black border and contains two lines of text explaining the purpose of the configuration.

# 앱 인벤터 기본

버튼을 클릭하면  
Screen1의 배경색이 바뀌도록  
해보자

The screenshot displays the Android Studio interface. On the left, the Outliner panel shows a tree view of components for 'Screen1', with 'Screen1' highlighted. The main visual editor shows a button with the text '실행' (Run). The Properties panel on the right shows the properties for the selected button, with the 'Background Color' property highlighted in red. An orange arrow points from the button in the Outliner to the text box above.

# 앱 인벤터 기본

언제 버튼1 ▾ .클릭했을때

실행 지정하기 Screen1 ▾ . 배경색 ▾ 값

이 뒤에 물리는 블록의 색깔이  
Screen1의 배경색이 된다.

# 앱 인벤터 기본

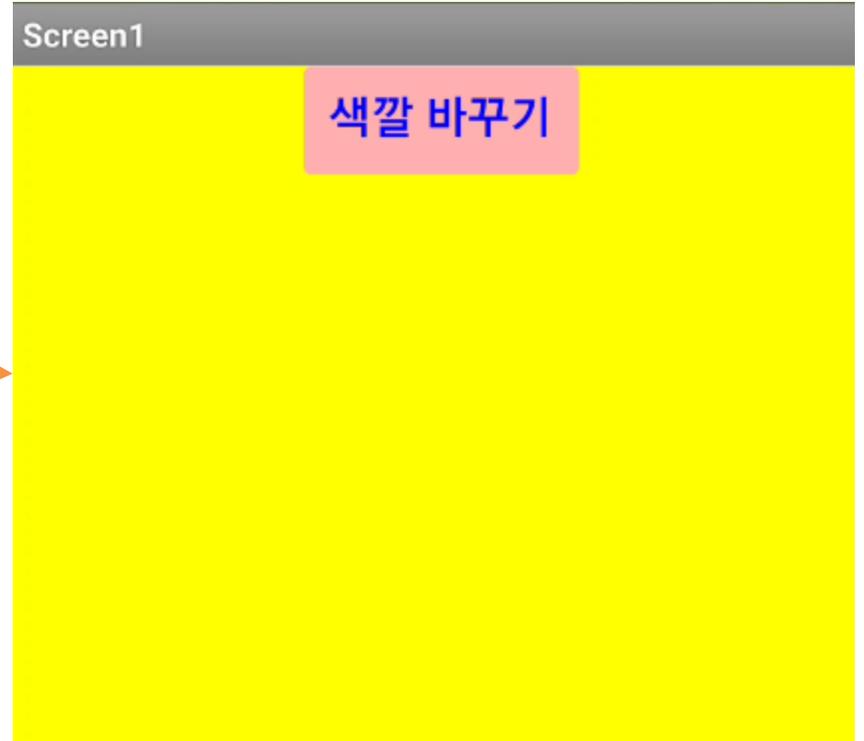
언제 **버튼1** .클릭했을때

실행 **지정하기** Screen1 . **배경색** 값

공통 블록에서 '색상'을 클릭하여  
원하는 색깔 블록을 가져와 물린다.

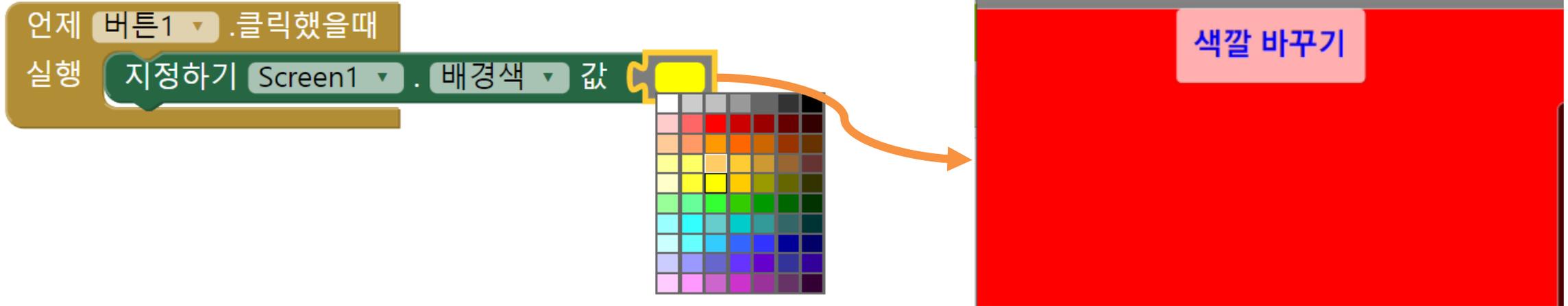


# 앱 인벤터 기본



스마트폰에서 버튼을 클릭해 보면  
지정한 색과 같은 색깔로 Screen1  
배경색이 바뀌는 것을 확인할 수 있다.

# 앱 인벤터 기본



색 부분만 클릭하면 다양한 색을 선택할 수 있다. 색 선택 후 스마트폰에서 버튼을 누르면 해당 색으로 바뀐다.

# 앱 인벤터 기본

언제 버튼1 ▾ .클릭했을때

실행

지정하기 Screen1 ▾ . 배경색 ▾ 값

색상 만들기

리스트 만들기

255

0

0

이 숫자는 위에서부터 R, G, B를 나타낸다.  
0~255사이의 숫자로 밝기를 조정할 수 있다.  
색깔을 적절히 섞으면 다양한 색을 표현할 수 있다.

# 앱 인벤터 기본

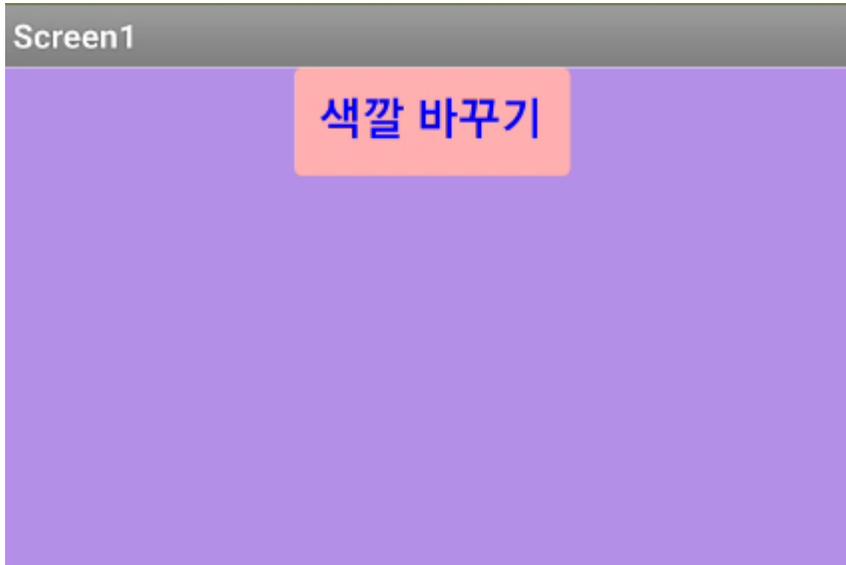


버튼을 누를 때마다 아무 색깔이나  
랜덤으로 나오게 하고 싶다면

'수학' 블록에 있는  
'임의의 정수' 블록을  
사용할 수 있다.

시작을 0, 끝을 255로 해서  
R, G, B에 각각 끼우자

# 앱 인벤터 기본



스마트폰에서 버튼을 누를 때마다  
임의의 색깔이 연출된다.

# 앱 인벤터 기본

The screenshot shows the Android Studio interface for designing an app screen. The top bar includes the 'chatbot' title, a dropdown menu for 'Screen1', and buttons for '스크린 추가...', '스크린 제거', and '갤러리에 발행하기'. A red box highlights the '디자이너' (Designer) button in the top right corner. The main workspace is divided into four panels: '팔레트' (Palette) on the left, '뷰어' (Viewer) in the center, '컴포넌트' (Components) on the right, and '속성' (Properties) on the far right. The '팔레트' panel has a search bar and a list of categories: '사용자 인터페이스', '레이아웃', '미디어', '그리기 & 애니메이션', '지도', and '센서'. The '센서' category is highlighted with a red box, and the '가속도센서' (Accelerometer) component is selected. A red arrow points from this component to the '뷰어' panel, which displays a smartphone screen with a pink button labeled '색깔 바꾸기'. The '컴포넌트' panel shows a list of components: 'Screen1' and '버튼1'. The '속성' panel shows the properties for 'Screen1', including a '스크린설명' (Screen description) field. A text box on the right contains the instruction: '디자이너에 가서 센서를 클릭 가속도 센서를 화면 가운데 놓는다.' (Go to the designer, click the sensor, and place the accelerometer sensor in the center of the screen.)

디자이너에 가서  
센서를 클릭  
가속도 센서를 화면 가운데  
놓는다.

# 앱 인벤터 기본



센서는 눈에 보이지 않으므로  
'보이지 않는 컴포넌트'로  
들어간다.

# 앱 인벤터 기본



블록으로 가보면  
디자이너에서 넣은  
가속도 센서가  
나타난 것을 볼 수 있다.  
이처럼 코딩을 하고 싶으면  
일단 디자이너에서 뷰어로  
컴포넌트를 끌고 와야 한다.

# 앱 인벤터 기본

공통 블록

- 제어
- 논리
- 수학
- 텍스트
- 리스트
- Dictionaries
- 색상
- 변수
- 함수

Screen1

- 버튼1
- 가속도센서1

모든 컴포넌트

언제 가속도센서1 .가속도가변화했을때  
X가속도 Y가속도 z가속도  
실행

언제 가속도센서1 .흔들렸을때  
실행

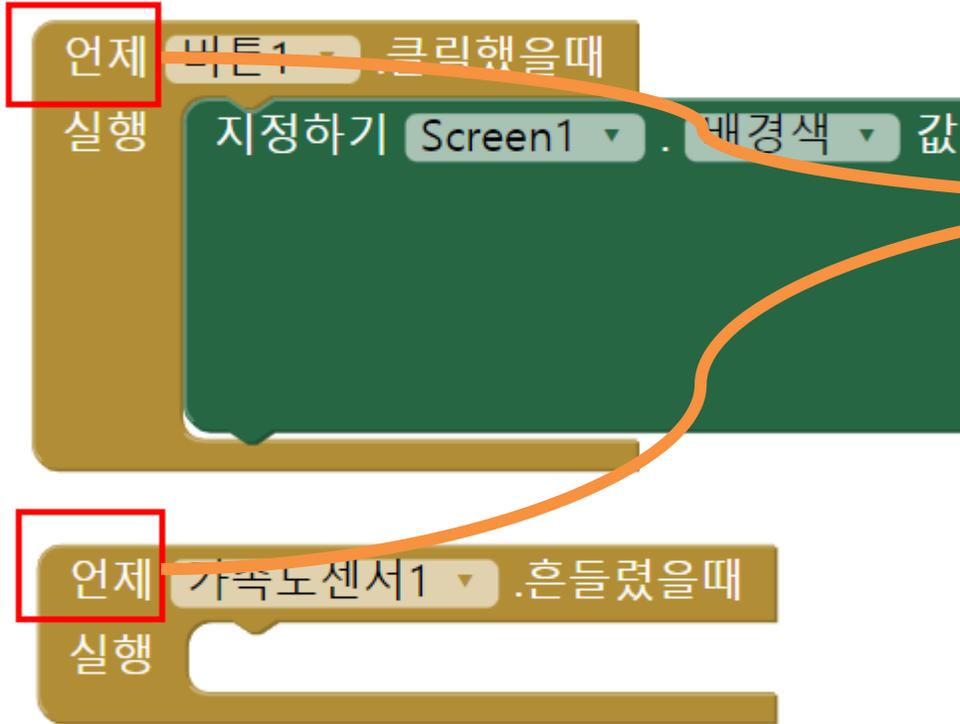
가속도센서1 .사용가능

가속도센서1 .활성화

지정하기 가속도센서1 .활성화 값

언제 가속도센서1 흔들렸을 때  
블록을 가져온다

# 앱 인벤터 기본



이벤트 핸들러 블록은 앞에  
'언제' 가 붙은 것을  
확인할 수 있다.

# 앱 인벤터 기본



이벤트 핸들러를 교체하면 이제 버튼이 아니라 흔들어야 색깔이 바뀌는 것을 확인할 수 있다.

# 앱 인벤터 기본



다음으로 넘어가기 위해 화면을 하얗게 만든다.

## 앱 제작의 실제1 – TALK

- 목감기에 걸려서 목소리가 안 나온다.
- 나 대신 누가 말해줄 수 없을까?



## 앱 제작의 실제1 – TALK

① 화면에 글자를 입력하는 빈 칸이 있다.

② 말하고 싶은 내용을 입력한다.

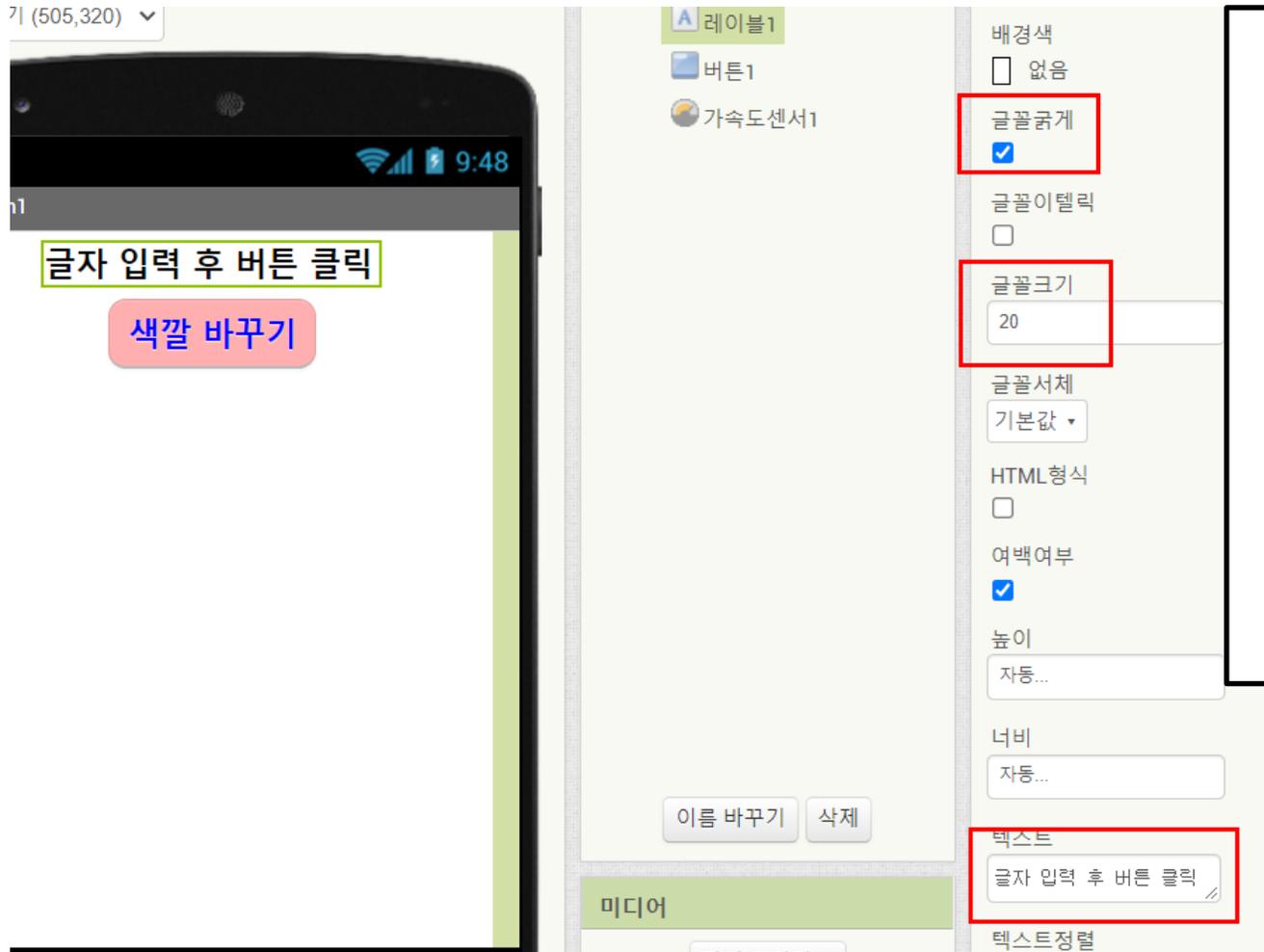
③ 버튼을 누르면 글자를 말해준다.

# 앱 제작의 실제1 – TALK



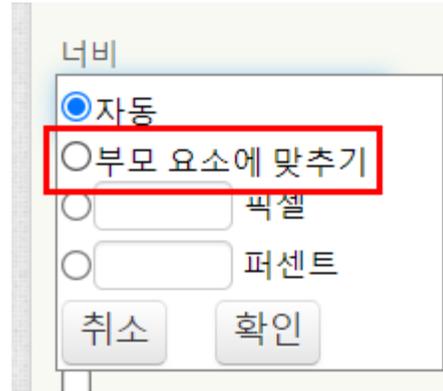
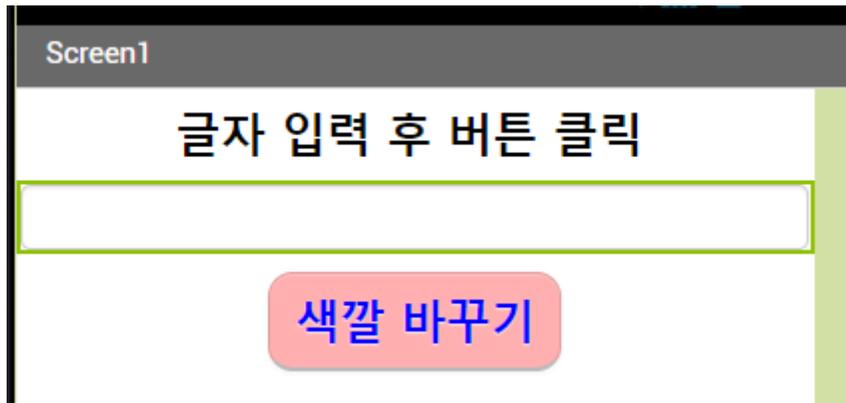
앱 인벤터에서는 '글자' 대신  
'텍스트'라는 용어를 씁니다.  
텍스트를 출력하기 위해서  
'레이블'을 사용합니다.

# 앱 제작의 실제1 – TALK



레이블의 텍스트 색깔이나 크기,  
내용 등을 바꾸고 싶을 땐  
'속성'에서 할 수 있다.

# 앱 제작의 실제1 – TALK



텍스트 박스를 가져와서  
너비를 '부모 요소에 맞추기' 로  
하면 좌우 너비가 부모인  
Screen1에 맞춰 늘어난다.

# 앱 제작의 실제1 – TALK

힌트

여러줄

글자 입력 후 버튼 클릭

색깔 바꾸기

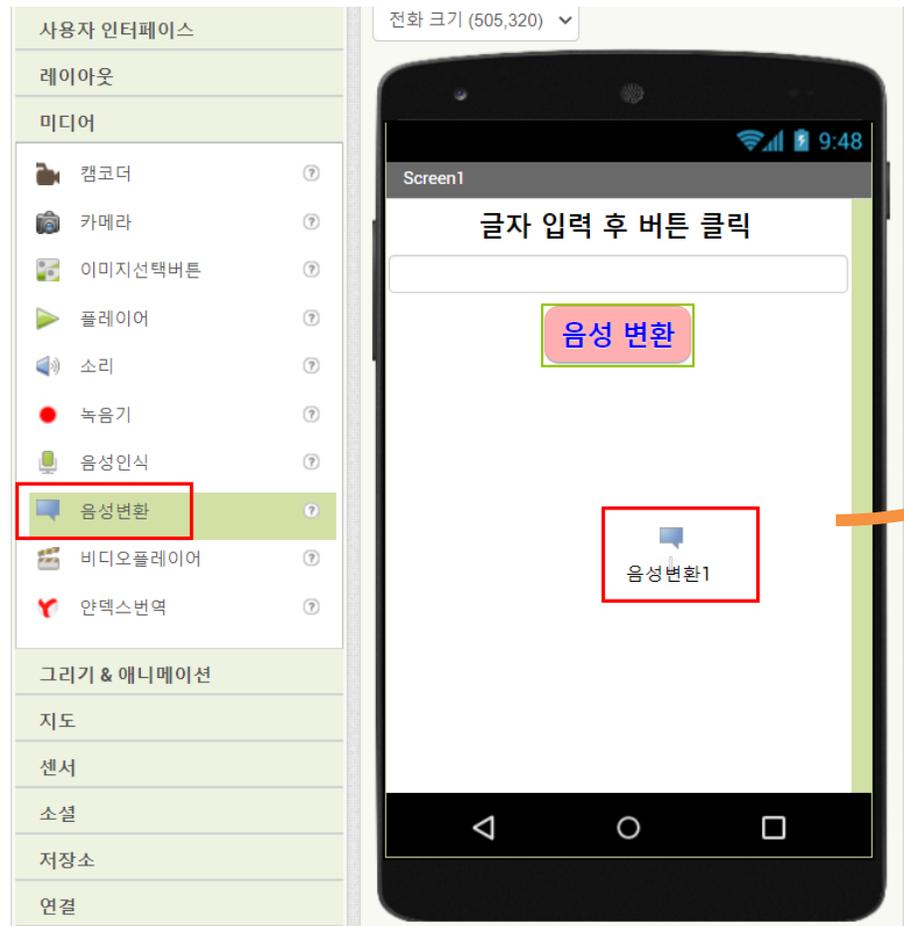
힌트 안에 있는 글자를 지우면  
스마트폰의 텍스트박스 안에  
있던 힌트 글자가 사라진다.

# 앱 제작의 실제1 – TALK



버튼 안의 글자를 바꿀 때는  
속성에서 '텍스트'의 글자를  
바꾸면 된다.

# 앱 제작의 실제1 – TALK



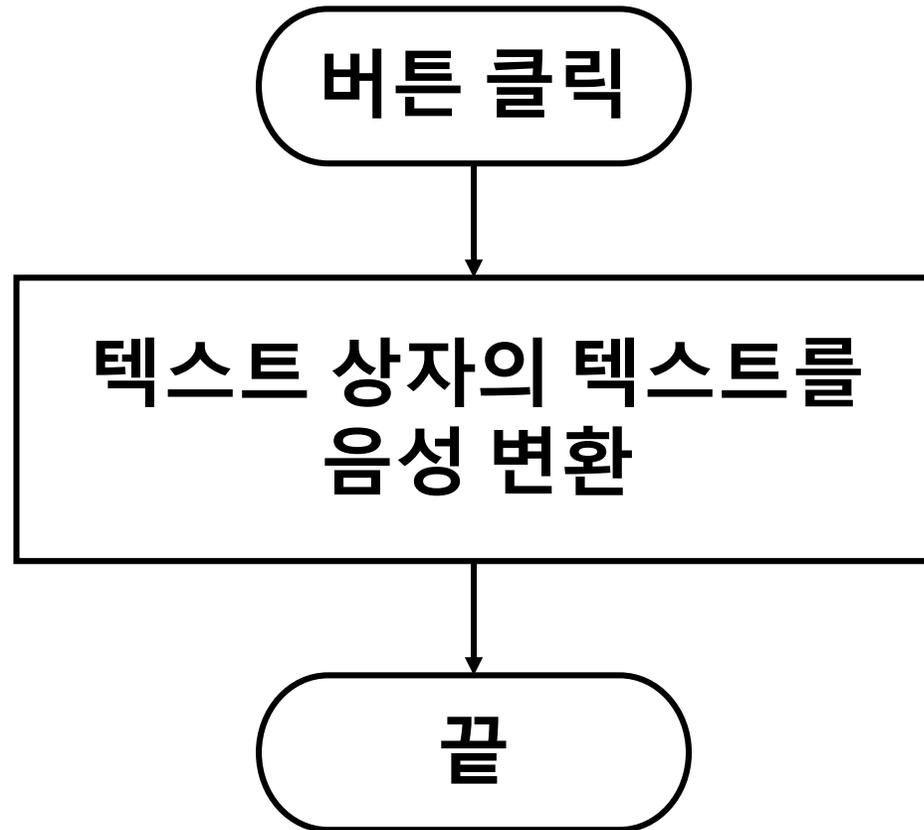
글자를 소리로 바꾸기 위해  
서는 '미디어'의 '음성변환'  
기능이 필요하다.

# 앱 제작의 실제1 – TALK



줄 간격을 벌려 주고 싶다면  
레이블을 넣고  
글자 색깔을  
'없음'으로 하면 된다.

# 앱 제작의 실제1 – TALK

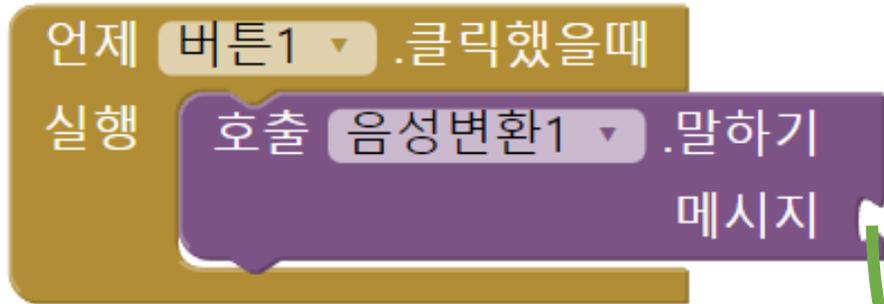


## 앱 제작의 실제1 – TALK

언제 `버튼1` .클릭했을때  
실행

이벤트 핸들러를 놔두고  
나머지 블록은 삭제한다.  
휴지통에 넣거나  
선택 후 delete 키를 누르면 된다.

## 앱 제작의 실제1 – TALK



버튼을 누르면 텍스트를  
음성으로 바꿀 것이므로  
'음성변환' 블록을 가져온다.

여기에 물리는 블록을 음성으로  
표현해 준다.

# 앱 제작의 실제1 – TALK



아무 글자나 쓰고 싶을 때는  
텍스트 블록의 맨 위에 있는  
빈칸 블록을 이용하면 된다

## 앱 제작의 실제1 – TALK



빈칸에 글자를 쓰는 대로  
스마트폰에서 버튼을 누르면  
그대로 말한다.

# 앱 제작의 실제1 – TALK

언제 버튼1 ▾ .클릭했을때

실행

호출 음성변환1 ▾ .말하기

메시지

텍스트박스1 ▾

. 텍스트 ▾

Screen1

글자 입력 후 버튼 클릭

안녕하세요

음성 변환

우리는 텍스트 박스 안에 있는 텍스트를 말하게 할 것이므로 '텍스트박스1 텍스트' 블록을 끼워 넣는다.

# 앱 제작의 실제1 – TALK

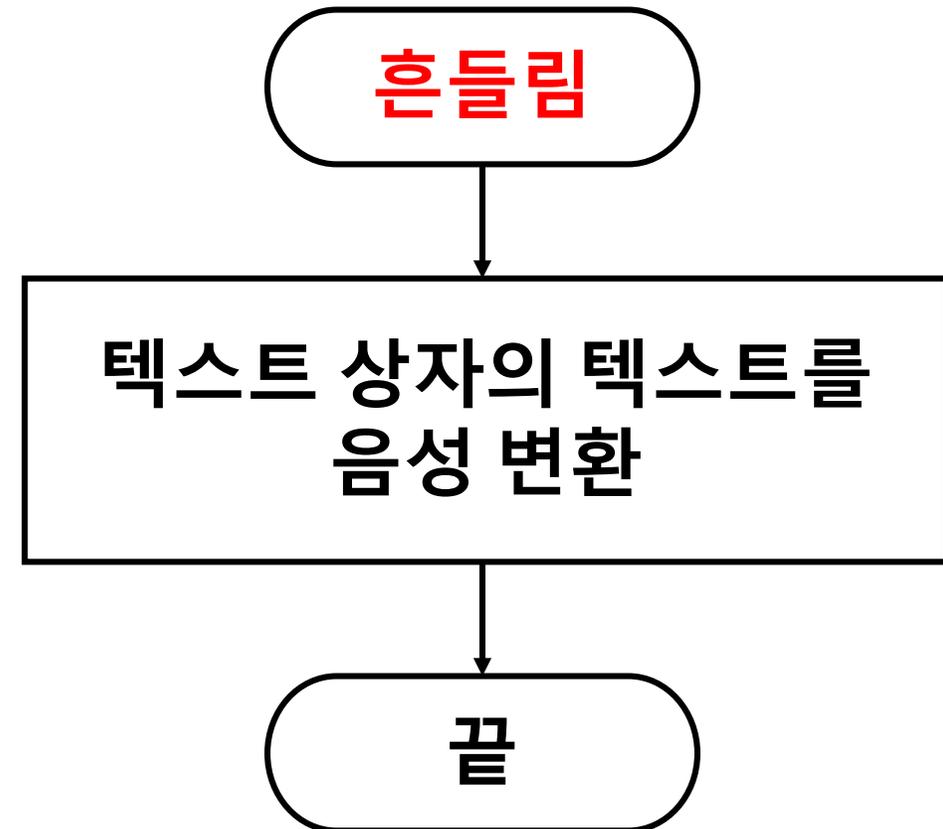
## QUIZ1

**버튼을 누르지 않고 스마트폰을 흔들어도 소리가  
나도록 해보자 (힌트: 센서-가속도 센서 활용)**

# 앱 제작의 실제1 – TALK

아래 블록을 활용해서 만들어 보자.

언제 `가속도센서1` .흔들렸을때  
실행



# 앱 제작의 실제1 – TALK

언제 버튼1 ▾ .클릭했을때

실행 호출 음성변환1 ▾ .말하기  
메시지

텍스트박스1 ▾ .배경색 ▾

언제 가속도센서1 ▾ .흔들렸을때

실행 호출 음성변환1 ▾ .말하기  
메시지

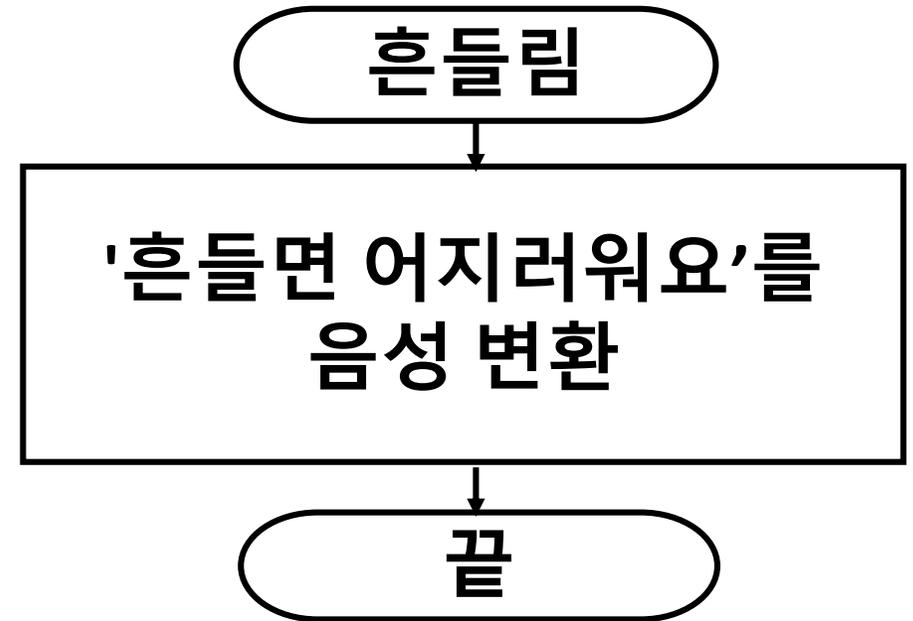
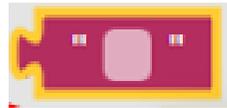
텍스트박스1 ▾ .배경색 ▾

# 앱 제작의 실제1 - TALK

## QUIZ2

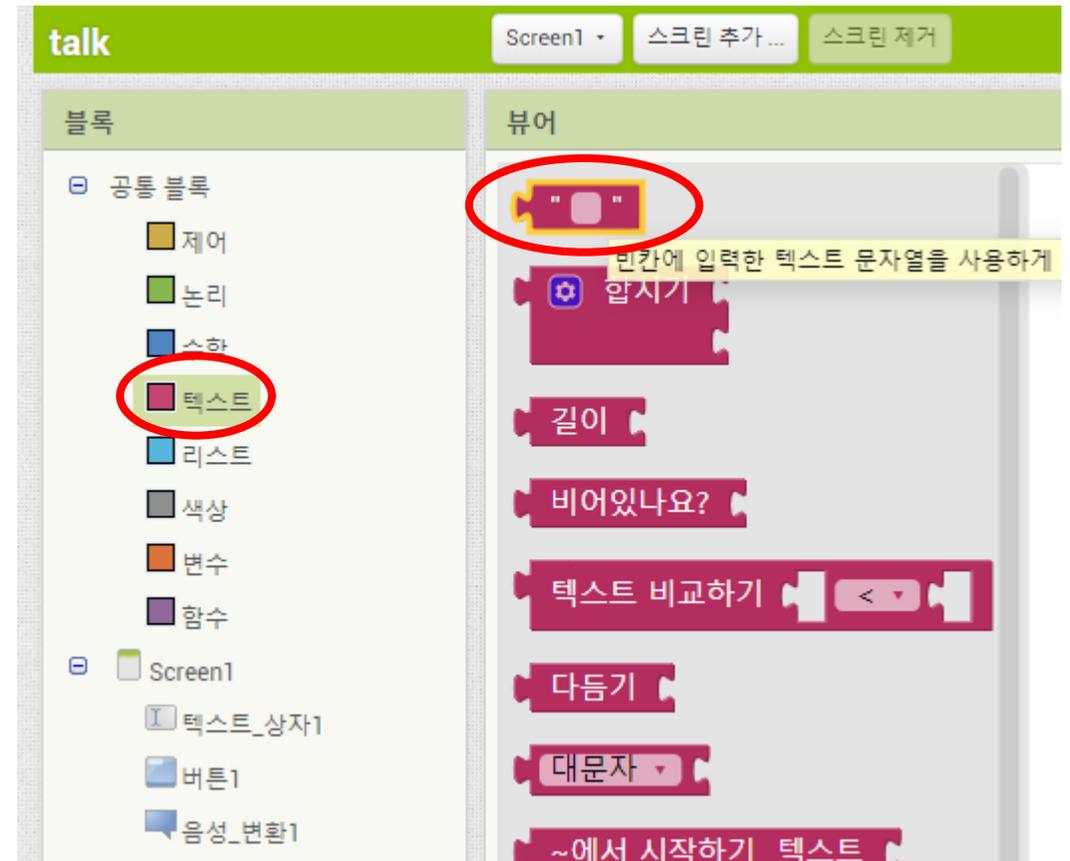
내용을 그때그때 입력하는 게 아니라  
스마트폰을 흔들면 항상 똑같은 말을 하도록  
해보자

(텍스트- 빈칸 블록 활용)

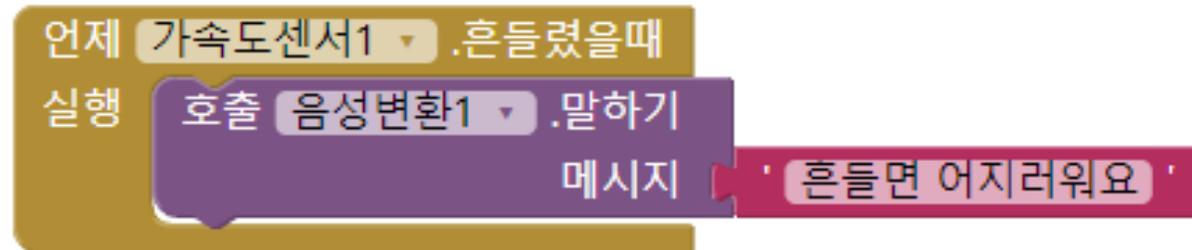
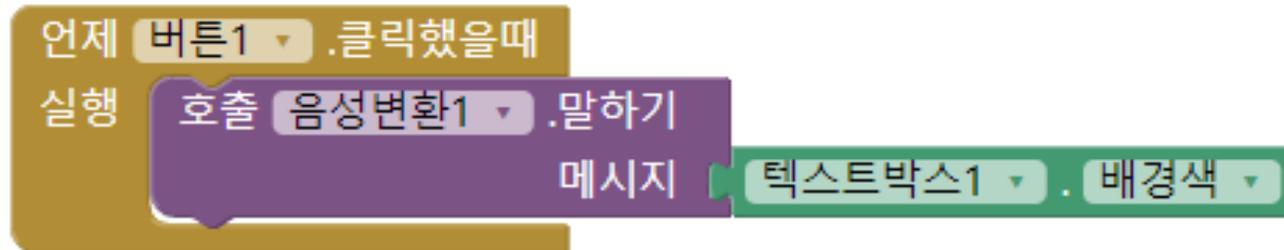


# 앱 제작의 실제1 – TALK

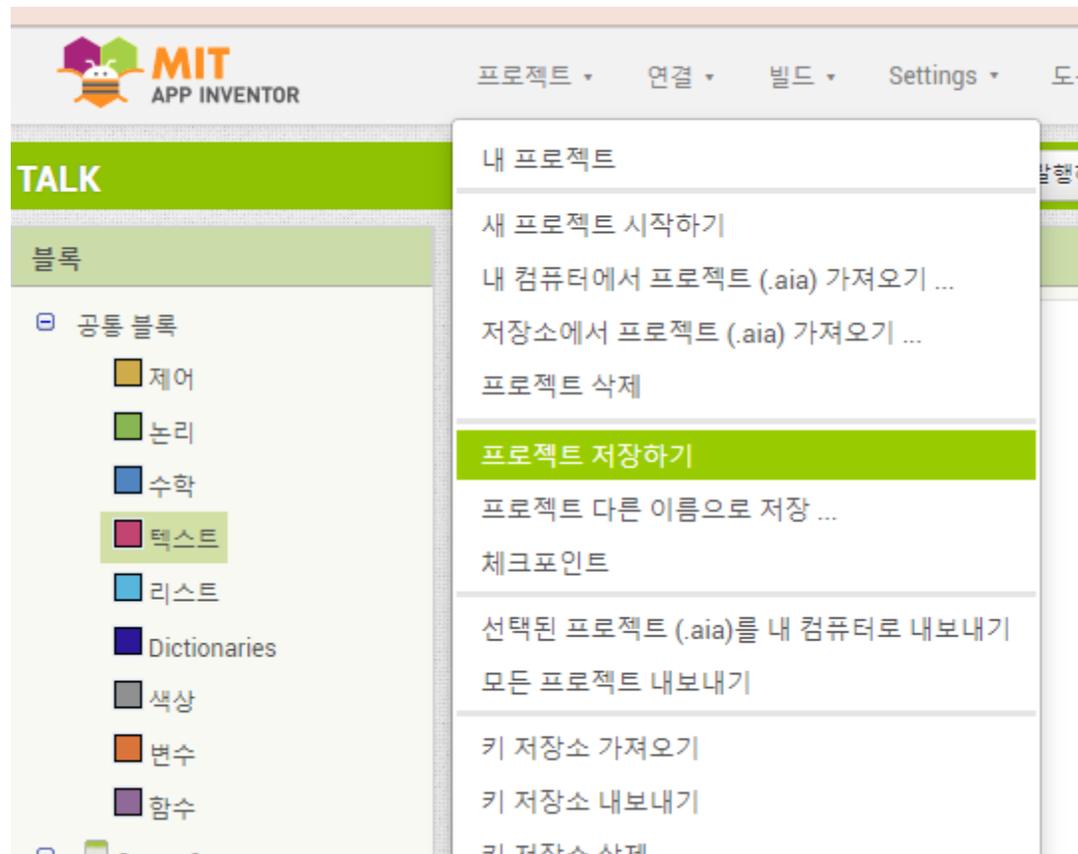
오른쪽 블록을 활용해서 만들어 보자.



# 앱 제작의 실제1 – TALK



# 앱 제작의 실제1 – TALK



## 2. 기본적인 앱 만들기



## 앱 제작의 실제2 – VOICETALK

- 청각장애인 택시 기사에게 목적지를 알려주고 싶을 때
- 어떤 앱을 만들면 좋을까?



## 앱 제작의 실제2 – VOICETALK

① 말을 시작한다는 버튼을 누른다.

② 말을 한다.

③ 버튼을 누르면 글자로 바꿔준다.

# 앱 제작의 실제2 – VOICETALK

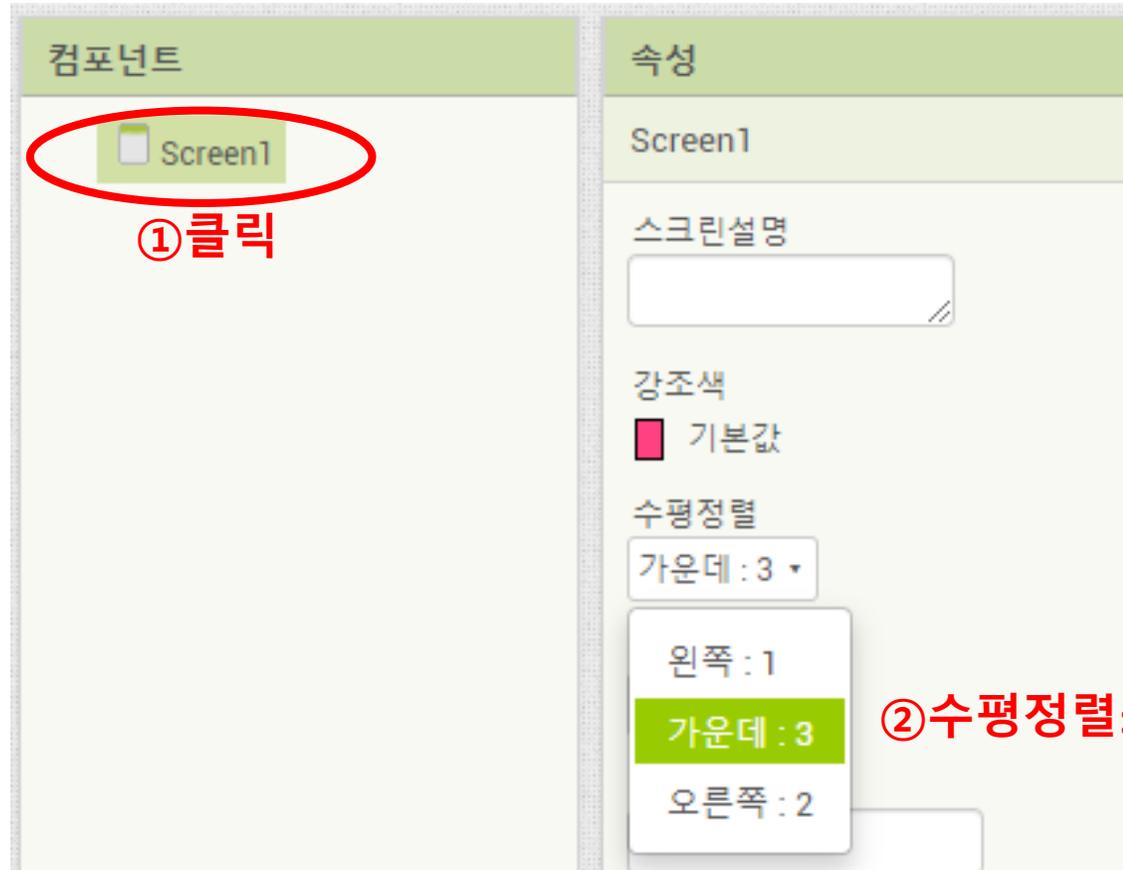


## 앱 제작의 실제2 – VOICETALK

새로운 앱인벤토리 프로젝트 만들기

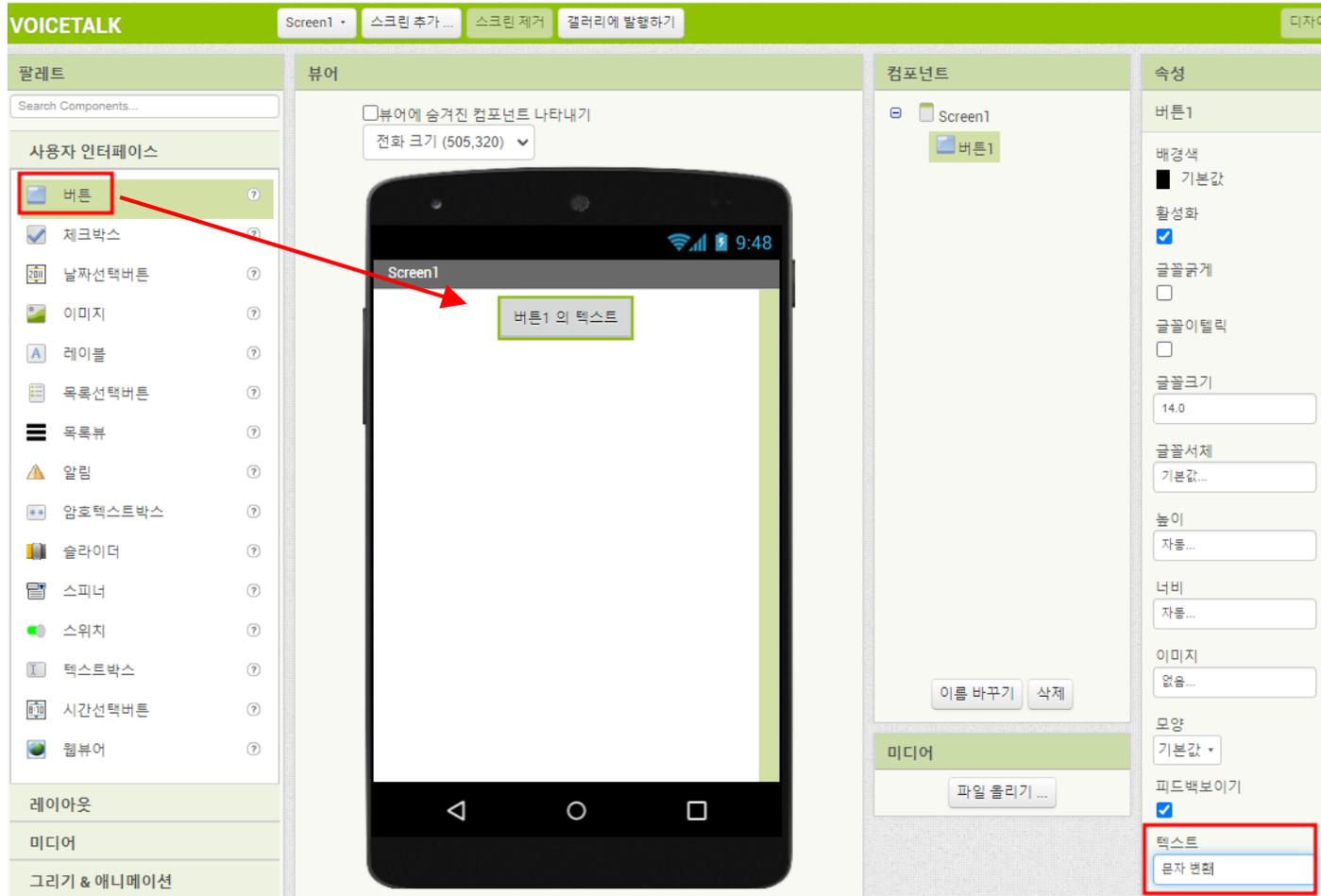
프로젝트 이름:

# 앱 제작의 실제2 – VOICETALK



# 앱 제작의 실제2 – VOICETALK

① 드래그



② 텍스트: 문자 변환

# 앱 제작의 실제2 – VOICETALK



② 텍스트: 삭제

# 앱 제작의 실제2 – VOICETALK



# 앱 제작의 실제2 – VOICETALK

① '블록' 클릭



# 앱 제작의 실제2 – VOICETALK

The screenshot displays the VOICETALK development environment. At the top, there is a green header with the text "VOICETALK" and several buttons: "Screen1", "스크린 추가...", "스크린 제거", and "갤러리에 발행하기".

The interface is divided into two main sections:

- 블록 (Blocks):** A vertical list of components on the left. Under "공통 블록" (Common Blocks), there are categories like "제어" (Control), "논리" (Logic), "수학" (Math), "텍스트" (Text), "리스트" (List), "Dictionaries", "색상" (Color), "변수" (Variable), and "함수" (Function). Below these, under "Screen1", a blue "버튼1" (Button1) block is highlighted with a red box and a red circle containing the number "1", with the Korean text "클릭" (Click) next to it.
- 뷰어 (Viewer):** A central workspace showing a storyboard of five yellow speech bubble-like blocks. Each block contains a trigger event and an action:
  - 언제 버튼1 .클릭했을때 (When Button1 is clicked)
  - 언제 버튼1 .포커스를받았을때 (When Button1 receives focus)
  - 언제 버튼1 .길게눌렸을때 (When Button1 is long-pressed)
  - 언제 버튼1 .포커스를잃었을때 (When Button1 loses focus)
  - 언제 버튼1 .터치다운했을때 (When Button1 is touched down)Each block has a "실행" (Execute) field below the trigger. A red arrow labeled "2 드래그" (2 Drag) points from the right side of the first block to a second, identical block that is being dragged from the right edge of the viewer.

# 앱 제작의 실제2 – VOICETALK

The screenshot displays the VOICETALK development environment. At the top, there are navigation buttons: "Screen1", "스크린 추가...", "스크린 제거", and "갤러리에 발행하기".

**블록 (Blocks):** A sidebar on the left lists various block categories:

- 공통 블록 (Common Blocks): 제어 (Control), 논리 (Logic), 수학 (Math), 텍스트 (Text), 리스트 (List), Dictionaries, 색상 (Color), 변수 (Variable), 함수 (Function).
- Screen1: 버튼1 (Button1), 레이블1 (Label1), 음성인식1 (Voice Recognition1).
- 모든 컴포넌트 (All Components):

**뷰어 (Viewer):** The main workspace shows a sequence of logic blocks:

- 언제 음성인식1 .텍스트가져온후에 (When Voice Recognition1 .Text is loaded after)
- 결과 partial (Result partial)
- 실행 (Execute)
- 언제 음성인식1 .텍스트가져오기전에 (When Voice Recognition1 .Text is loaded before)
- 실행 (Execute)
- 호출 음성인식1 .텍스트가져오기 (Call Voice Recognition1 .Text is loaded)
- 호출 음성인식1 .정지 (Call Voice Recognition1 .Stop)
- 음성인식1 .언어 (Voice Recognition1 .Language)
- 지정하기 음성인식1 .언어 값 (Specify Voice Recognition1 .Language value)
- 음성인식1 .결과 (Voice Recognition1 .Result)

**Annotations:**

- A red box highlights "음성인식1" in the "Screen1" block list, with the label "① 클릭" (Click).
- A red arrow points from the "호출 음성인식1 .텍스트가져오기" block in the viewer to a yellow callout box containing "언제 버튼1 .클릭했을때" (When Button1 .Clicked) and "호출 음성인식1 .텍스트가져오기" (Call Voice Recognition1 .Text is loaded). This callout is labeled "② 드래그" (Drag).

# 앱 제작의 실제2 – VOICETALK

**블록**

- 공통 블록
  - 제어
  - 논리
  - 수학
  - 텍스트
  - 리스트
  - Dictionaries
  - 색상
  - 변수
  - 함수
- Screen1
  - 버튼1
  - 레이블1
  - 음성인식1
- 모든 컴포넌트

**뷰어**

언제 음성인식1 .텍스트가져온후에  
결과 partial  
실행

언제 음성인식1 .텍스트가져오기전에  
실행

호출 음성인식1 .텍스트가져오기

호출 음성인식1 .정지

음성인식1 . 언어

지정하기 음성인식1 . 언어 값

음성인식1 . 결과

언제 버튼1 .클릭했을때  
실행 호출 음성인식1 .텍스트가져오기

언제 음성인식1 .텍스트가져온후에  
결과 partial  
실행

# 앱 제작의 실제2 – VOICETALK

The screenshot displays the VOICETALK development environment. On the left, a sidebar lists various components under '공통 블록' (Common Blocks), 'Screen1', and '모든 컴포넌트' (All Components). The '레이블1' (Label1) component is highlighted in red. The main workspace shows a list of components with their properties, such as '배경색' (Background Color), '글꼴크기' (Font Size), 'HTMLContent', '여백여부' (Margin), '높이' (Height), and '텍스트' (Text). A red box highlights the '지정하기 레이블1 . 텍스트 값' (Assign Label1 . Text Value) property. On the right, a detailed view of the '지정하기 레이블1 . 텍스트 값' property is shown, including a warning icon (a red 'X' in a circle) and a '결과' (Result) field set to 'partial'. A red arrow points from the highlighted property in the main workspace to this detailed view. The interface also includes a 'Show warnings' button at the bottom and a trash icon on the right.

# 앱 제작의 실제2 – VOICETALK

The screenshot displays the VOICETALK development environment. The interface is divided into several sections:

- VOICETALK Header:** Includes a dropdown menu for 'Screen1', buttons for '스크린 추가 ...', '스크린 제거', and '갤러리에 발행하기', and a '디자인' button on the right.
- 블록 (Blocks) Panel:** A sidebar on the left containing a '공통 블록' (Common Blocks) section with categories like 제어, 논리, 수학, 텍스트, 리스트, Dictionaries, 색상, 변수, and 함수. Below this is a 'Screen1' section with '버튼1', '레이블1', and '음성인식1' (highlighted with a red box). At the bottom is a '모든 컴포넌트' (All Components) section.
- 뷰어 (Viewer) Panel:** The main workspace showing a sequence of logic blocks:
  - 언제 음성인식1 . 텍스트가져온후에 (When Voice Recognition 1 . Text is loaded)
  - 결과 partial (Result partial)
  - 실행 (Execute)
  - 언제 음성인식1 . 텍스트가져오기전에 (When Voice Recognition 1 . Text is loaded before)
  - 실행 (Execute)
  - 호출 음성인식1 . 텍스트가져오기 (Call Voice Recognition 1 . Text is loaded)
  - 호출 음성인식1 . 정지 (Call Voice Recognition 1 . Stop)
  - 음성인식1 . 언어 (Voice Recognition 1 . Language)
  - 지정하기 음성인식1 . 언어 값 (Assign Voice Recognition 1 . Language value)
  - 음성인식1 . 결과 (Voice Recognition 1 . Result) - highlighted with a red box
  - 음성인식1 . UseLegacy (Voice Recognition 1 . Use Legacy)
  - 지정하기 음성인식1 . UseLegacy 값 (Assign Voice Recognition 1 . Use Legacy value)
- Logic Flow:** A separate flow of logic blocks on the right:
  - 언제 버튼1 . 클릭했을때 (When Button 1 . Clicked)
  - 실행 호출 음성인식1 . 텍스트가져오기 (Execute Call Voice Recognition 1 . Text is loaded)
  - 언제 음성인식1 . 텍스트가져온후에 (When Voice Recognition 1 . Text is loaded)
  - 결과 partial (Result partial)
  - 실행 지정하기 레이블1 . 텍스트 값 음성인식1 . 결과 (Execute Assign Label 1 . Text value Voice Recognition 1 . Result)

A red arrow points from the '음성인식1 . 결과' block in the main viewer to the '음성인식1 . 결과' block in the logic flow on the right.

## 앱 제작의 실제2 – VOICETALK

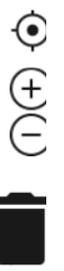
끝판왕!

말한 내용이 다시 음성으로 나오도록  
해보자

힌트

1. 음성 변환 센서가 필요하다

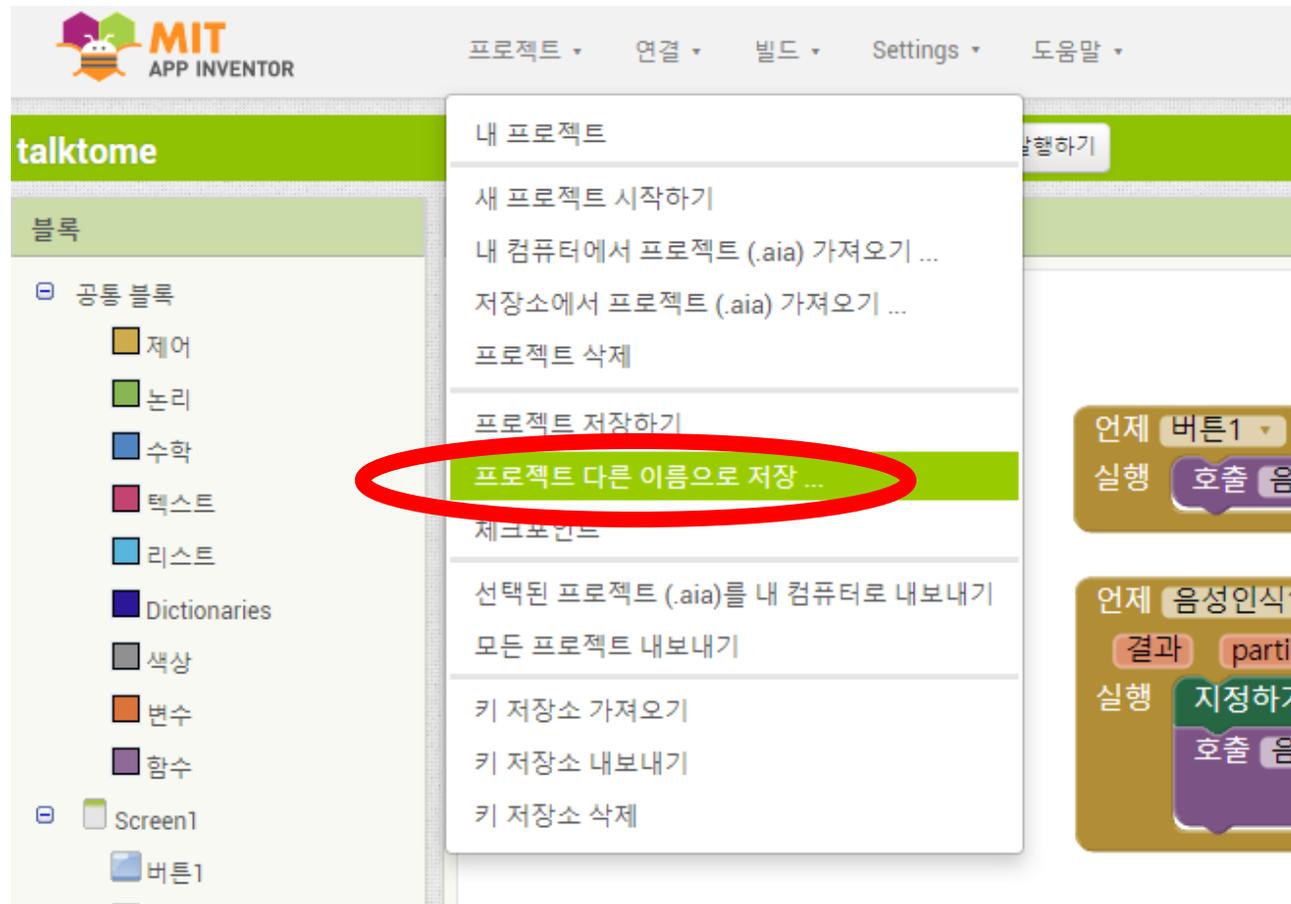
# 앱 제작의 실제2 – VOICETALK



## 앱 제작의 실제2 – VOICETALK

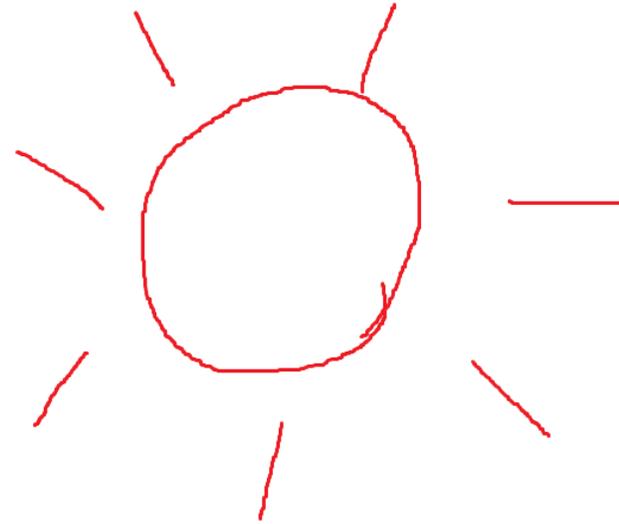
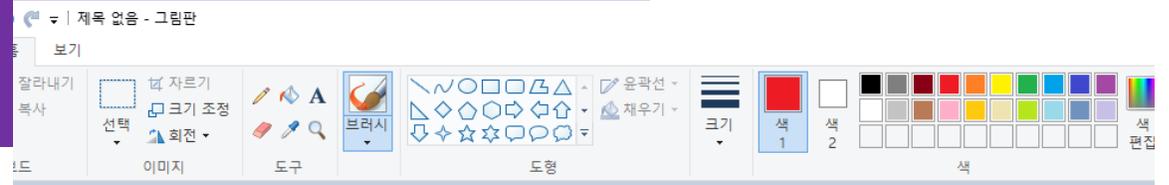


# 앱 제작의 실제2 – VOICETALK

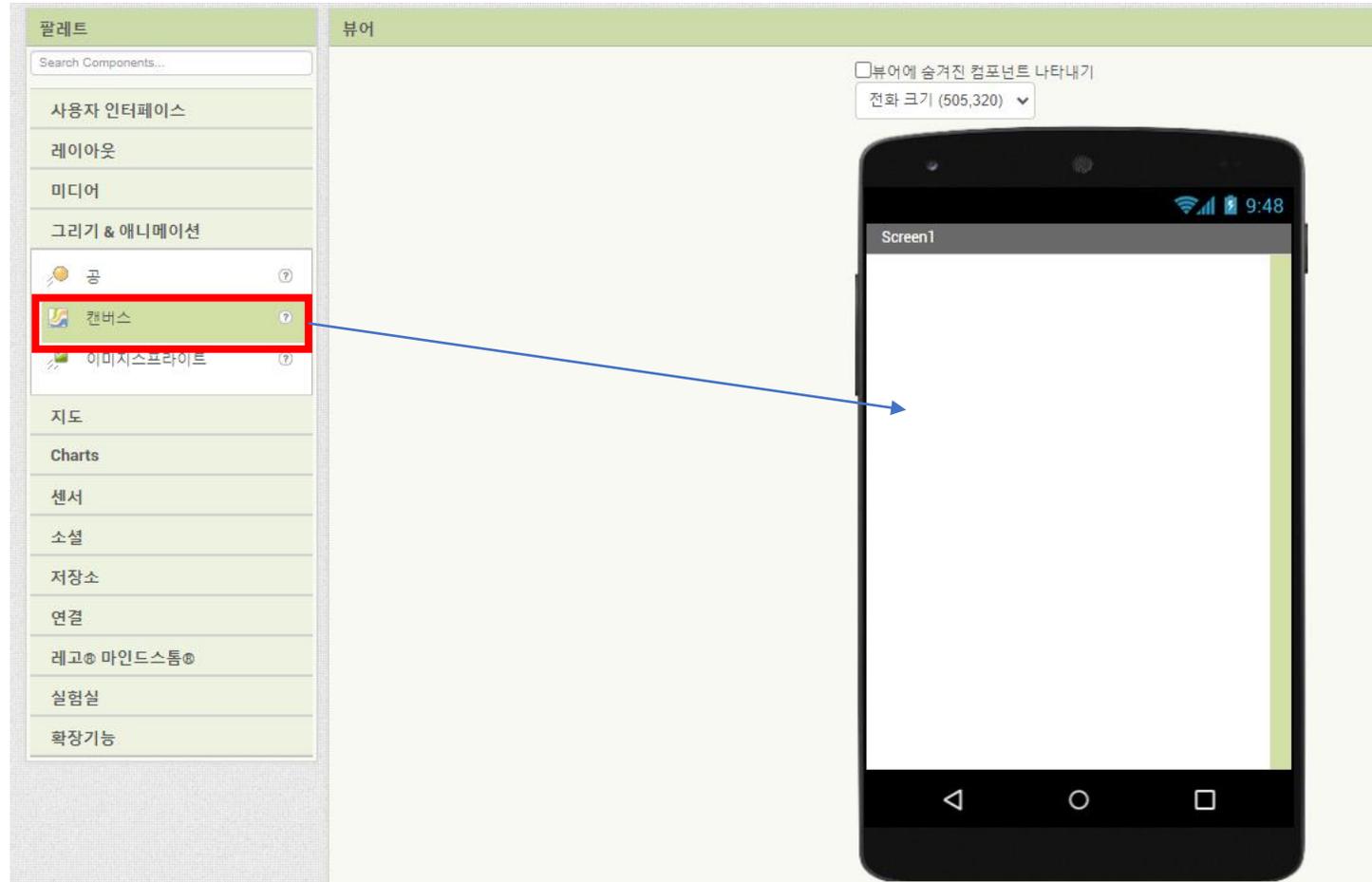


# 앱 제작의 실제3 - 그림판

- 스마트폰에서도 그림판으로 그림을 그리고 싶다.



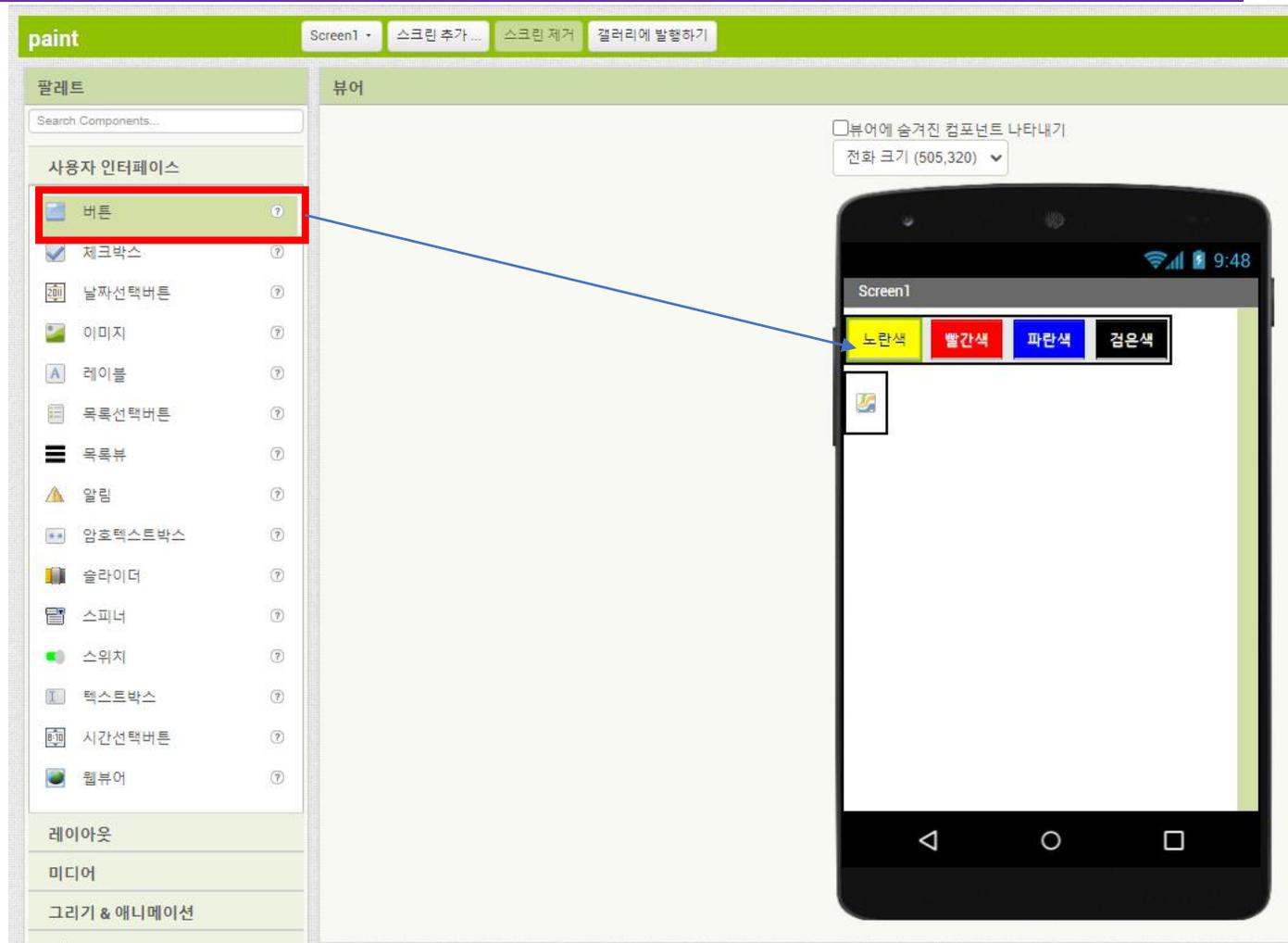
# 앱 제작의 실제3 – 그림판



# 앱 제작의 실제3 – 그림판



# 앱 제작의 실제3 – 그림판



# 앱 제작의 실제3 - 그림판

뷰어에 숨겨진 컴포넌트 나타내기  
전화 크기 (505,320) ▾



컴포넌트

- Screen1
  - 수령배지1
    - 버튼1
    - 버튼2
    - 버튼3
    - 버튼4
  - 캔버스1

이름 바꾸기 삭제

속성

버튼1

배경색  
노랑

활성화

글꼴굵게

글꼴이탤릭

글꼴크기  
14.0

글꼴서체  
기본값...

높이  
자동...

너비  
자동...

이미지  
없음...

모양  
기본값 ▾

피드백보이기

텍스트  
노란색

텍스트정렬  
가운데 : 1 ▾

미디어  
파일 올리기 ...

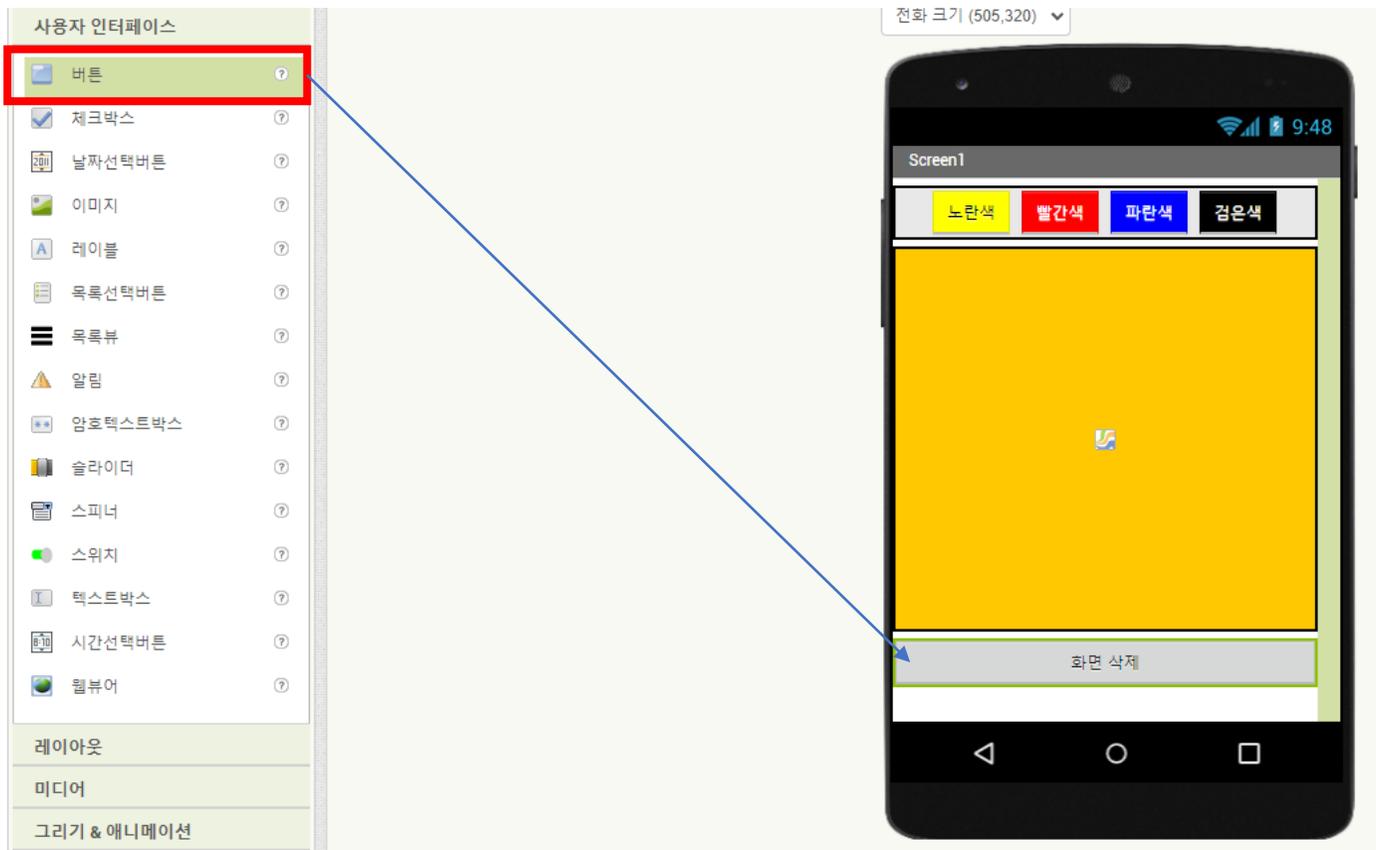
# 앱 제작의 실제3 – 그림판



# 앱 제작의 실제3 – 그림판



# 앱 제작의 실제3 – 그림판

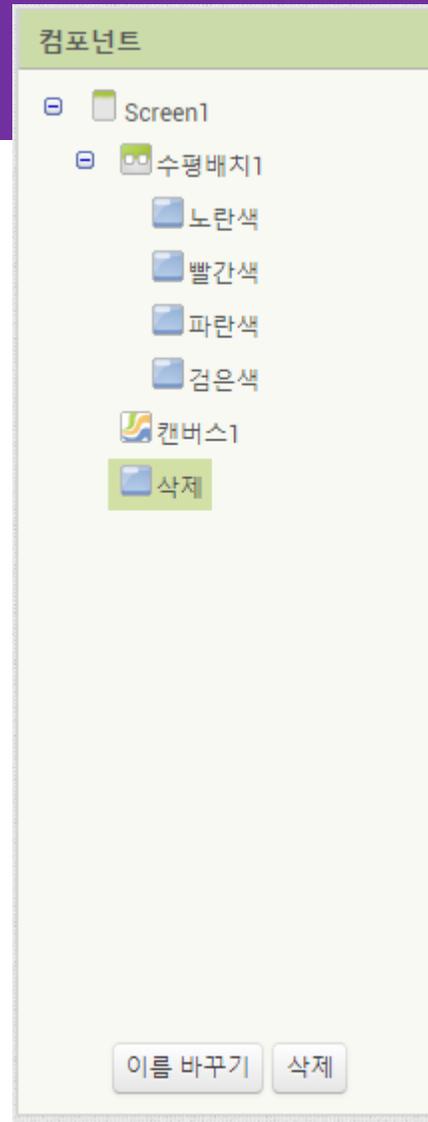


# 앱 제작의 실제3 – 그림판

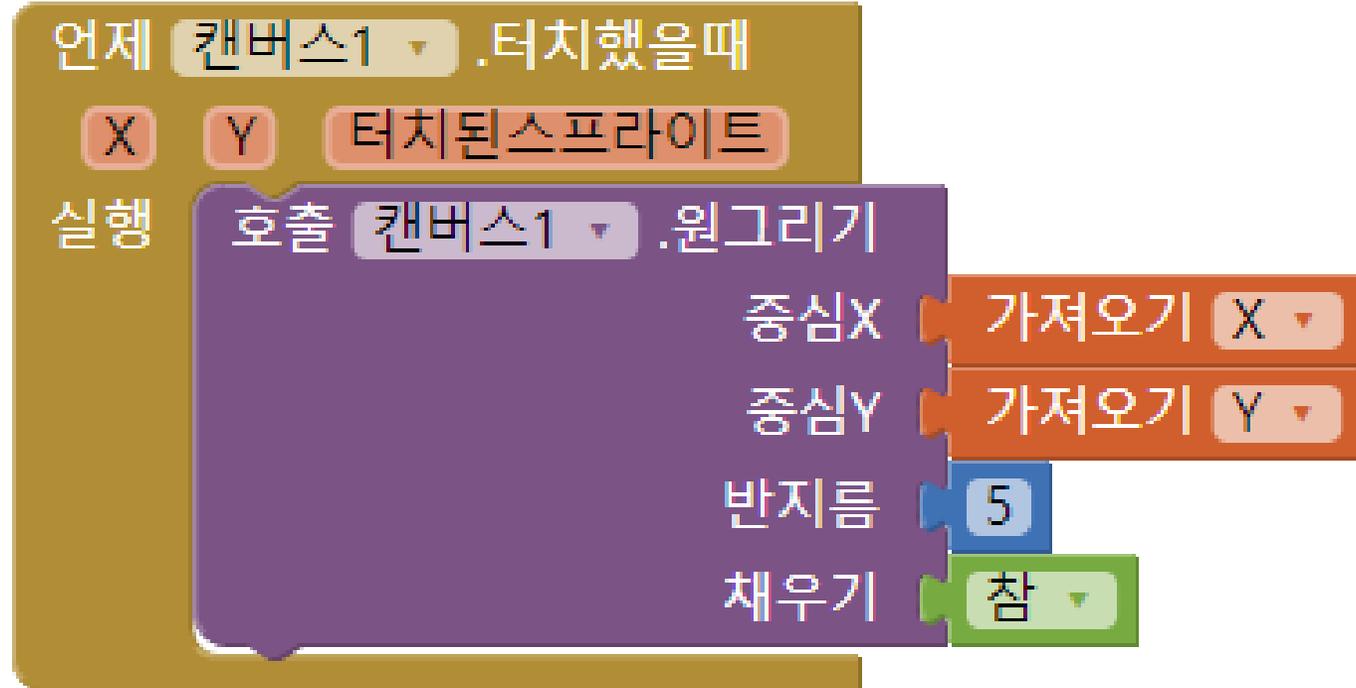


## 앱 제작의 실제3 – 그림판

- 나중에 헛갈리지 않도록 버튼들의 이름을 다음과 같이 미리 변경해두자



## 앱 제작의 실제3 – 그림판



## 앱 제작의 실제3 – 그림판

언제 캔버스1 ▾ .드래그

시작X

시작Y

이전X

이전Y

현재X

현재Y

드래그된모든스프라이트

실행

호출 캔버스1 ▾ .선그리기

x1 가져오기 이전X ▾

Y1 가져오기 이전Y ▾

x2 가져오기 현재X ▾

Y2 가져오기 현재Y ▾

## 앱 제작의 실제3 – 그림판

언제 노란색 ▾ .클릭했을때

실행 지정하기 캔버스1 ▾ . 페인트색상 ▾ 값 

언제 빨간색 ▾ .클릭했을때

실행 지정하기 캔버스1 ▾ . 페인트색상 ▾ 값 

언제 파란색 ▾ .클릭했을때

실행 지정하기 캔버스1 ▾ . 페인트색상 ▾ 값 

언제 검은색 ▾ .클릭했을때

실행 지정하기 캔버스1 ▾ . 페인트색상 ▾ 값 

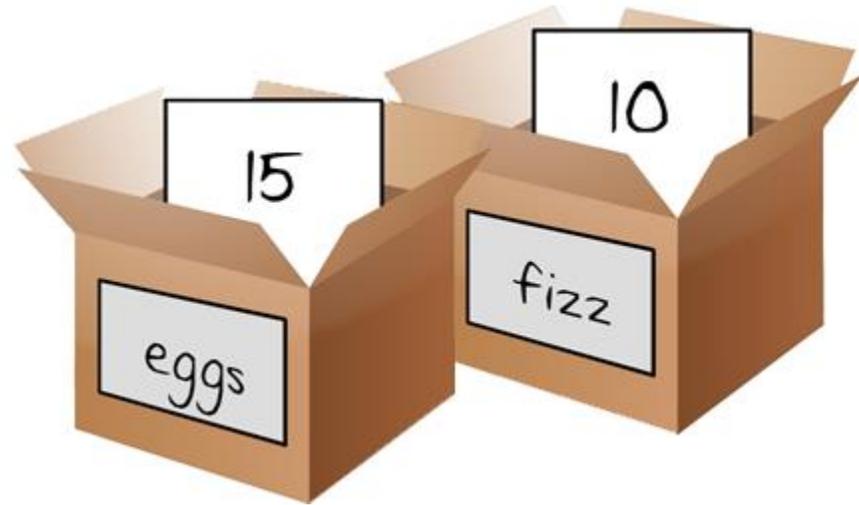
언제 삭제 ▾ .클릭했을때

실행 호출 캔버스1 ▾ .지우기

## 앱 제작의 실제3 – 그림판

### 변수

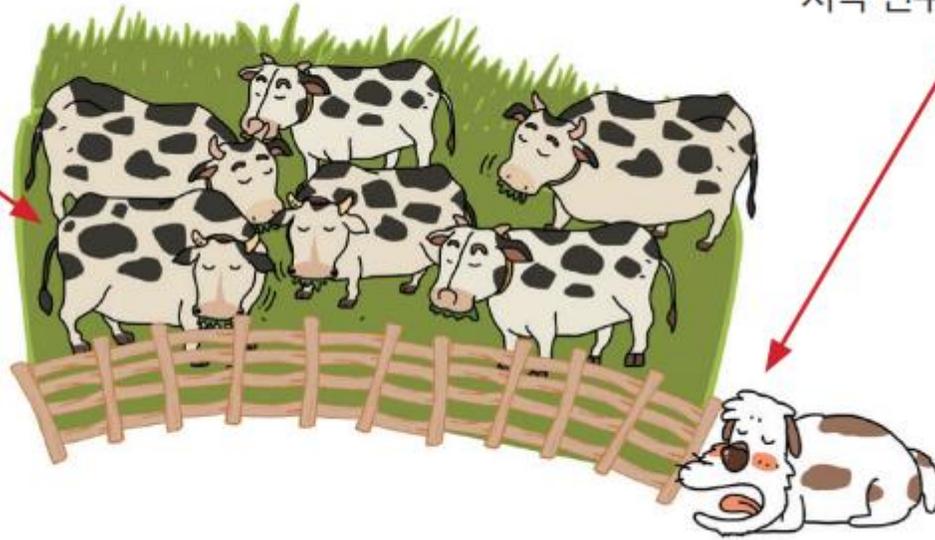
- 데이터를 저장하는 공간
- 상자와 같다.



## 앱 제작의 실제3 – 그림판

전역 변수(global variable)

지역 변수(local variable)



# 앱 제작의 실제3 - 그림판

The image shows a software development environment with a block-based programming interface. It is divided into three main sections: '블록' (Blocks), '뷰어' (Viewer), and '미디어' (Media).

- 블록 (Blocks):** A list of reusable code blocks. Under '공통 블록' (Common Blocks), there are categories like '제어' (Control), '논리' (Logic), '수학' (Math), '텍스트' (Text), '리스트' (List), 'Dictionaries', '색상' (Color), '변수' (Variable), and '함수' (Function). The '변수' (Variable) category is highlighted. Under 'Screen1', there are blocks for '수평배치1' (Horizontal Layout 1) with sub-blocks for colors (노란색, 빨간색, 파란색, 검은색), '캔버스1' (Canvas 1), and '삭제' (Delete). Buttons for '이름 바꾸기' (Rename) and '삭제' (Delete) are at the bottom.
- 뷰어 (Viewer):** A workspace where blocks are assembled. It contains several blocks:
  - '지역변수 만들기 이름 초기값' (Create local variable name initial value) blocks.
  - '가져오기' (Load) block.
  - '지정하기 값' (Specify value) block.
  - '지역변수 만들기 이름 초기값 실행' (Create local variable name initial value execute) blocks.
- 미디어 (Media):** A section at the bottom with a '파일 올리기 ...' (Upload file ...) button.

## 앱 제작의 실제3 – 그림판

전역변수 만들기 굵기 초기값 5

언제 캔버스1 .터치했을때

X Y 터치된스프라이트

실행

호출 캔버스1 .원그리기

중심X

가져오기 X

중심Y

가져오기 Y

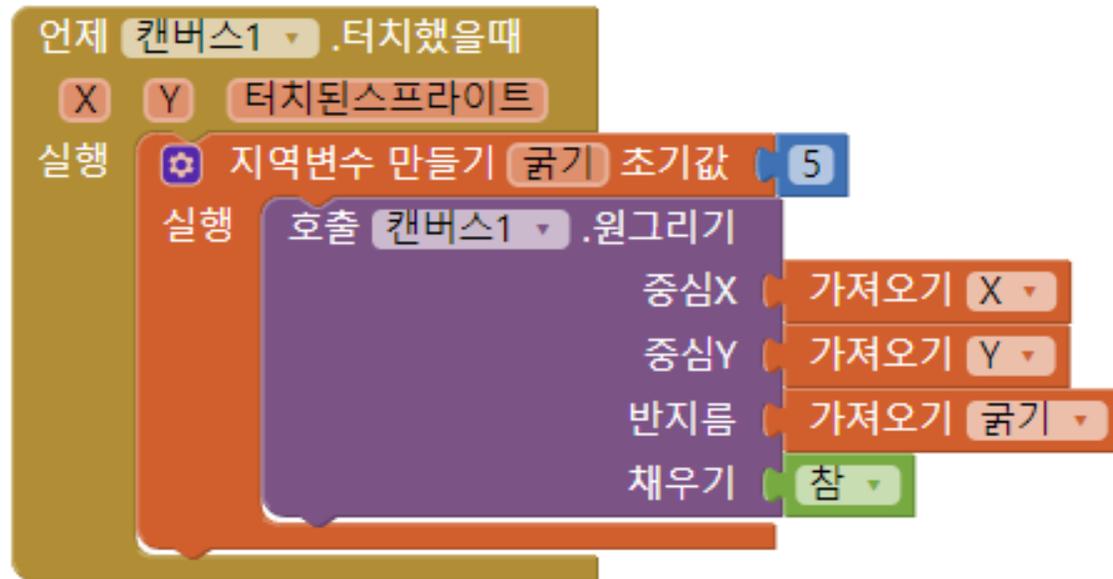
반지름

가져오기 전역변수 굵기

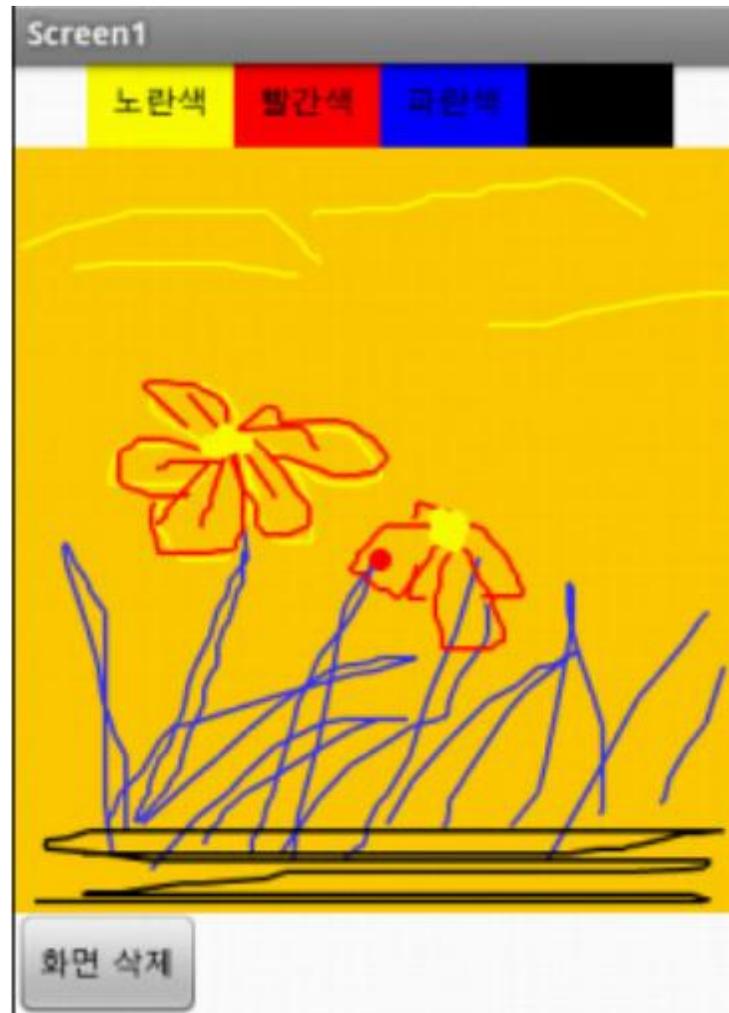
채우기

참

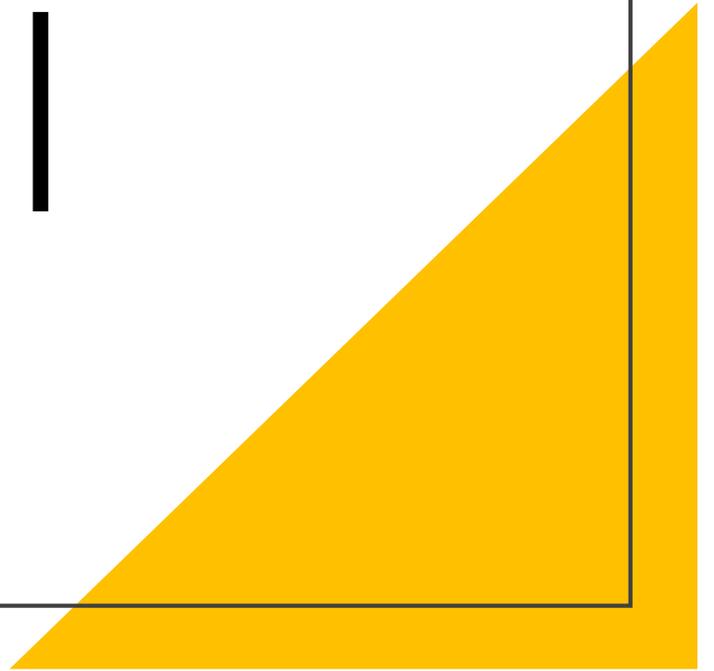
## 앱 제작의 실제3 – 그림판



## 앱 제작의 실제3 – 그림판

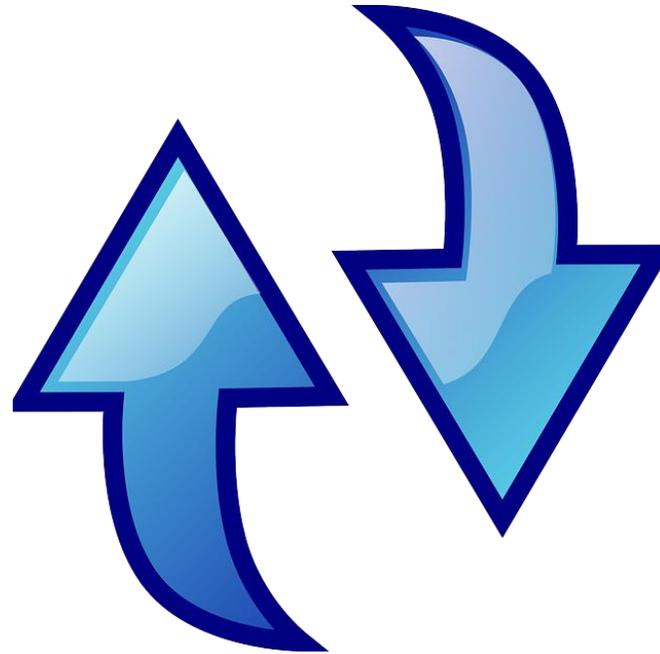


# 3. 게임 앱 만들기



## 앱 제작의 실제4 – 숫자 예측 게임

- 숫자 예측 게임을 만들고 싶다.

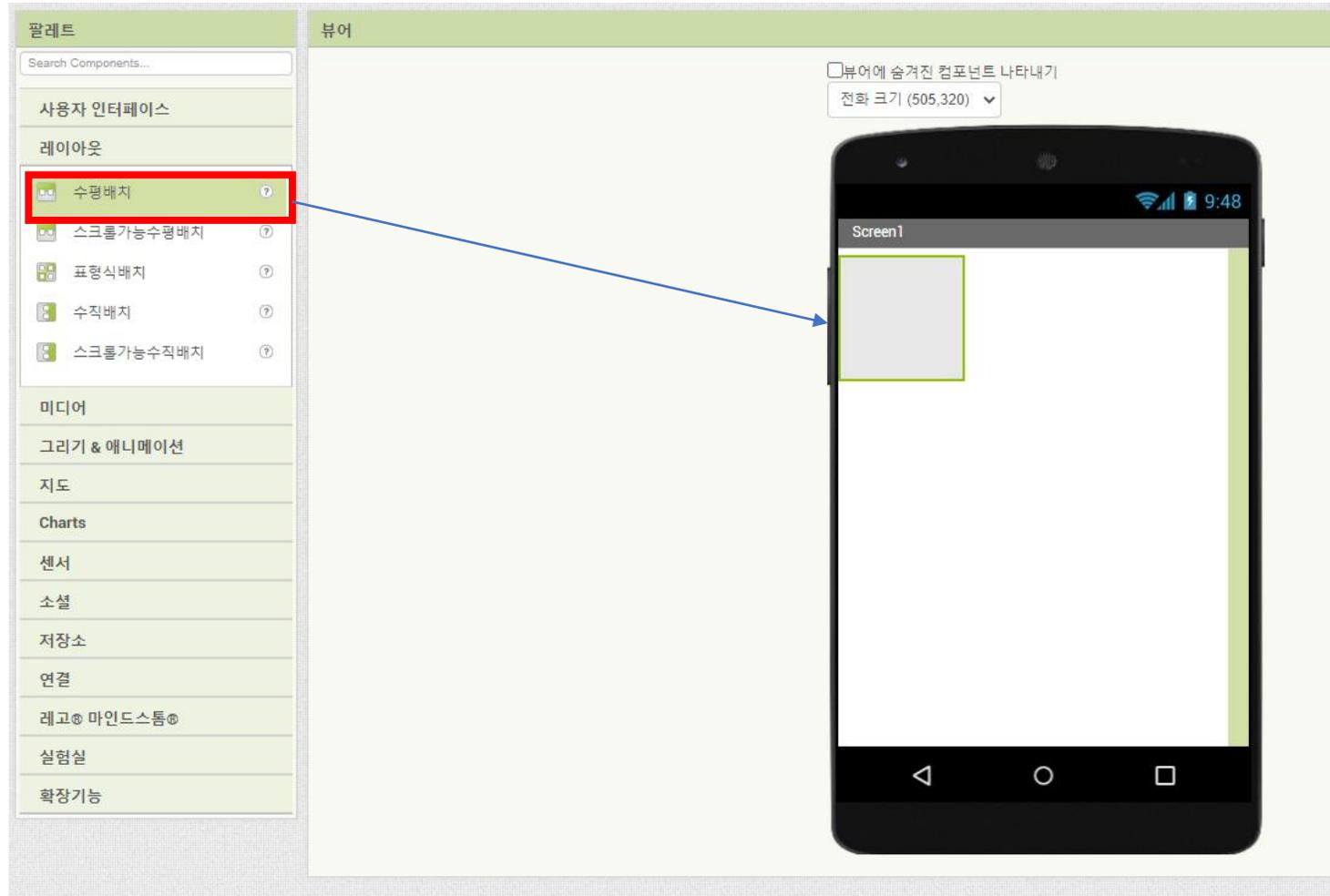


## 앱 제작의 실제4 – 숫자 예측 게임

새로운 앱인벤토리 프로젝트 만들기

프로젝트 이름:

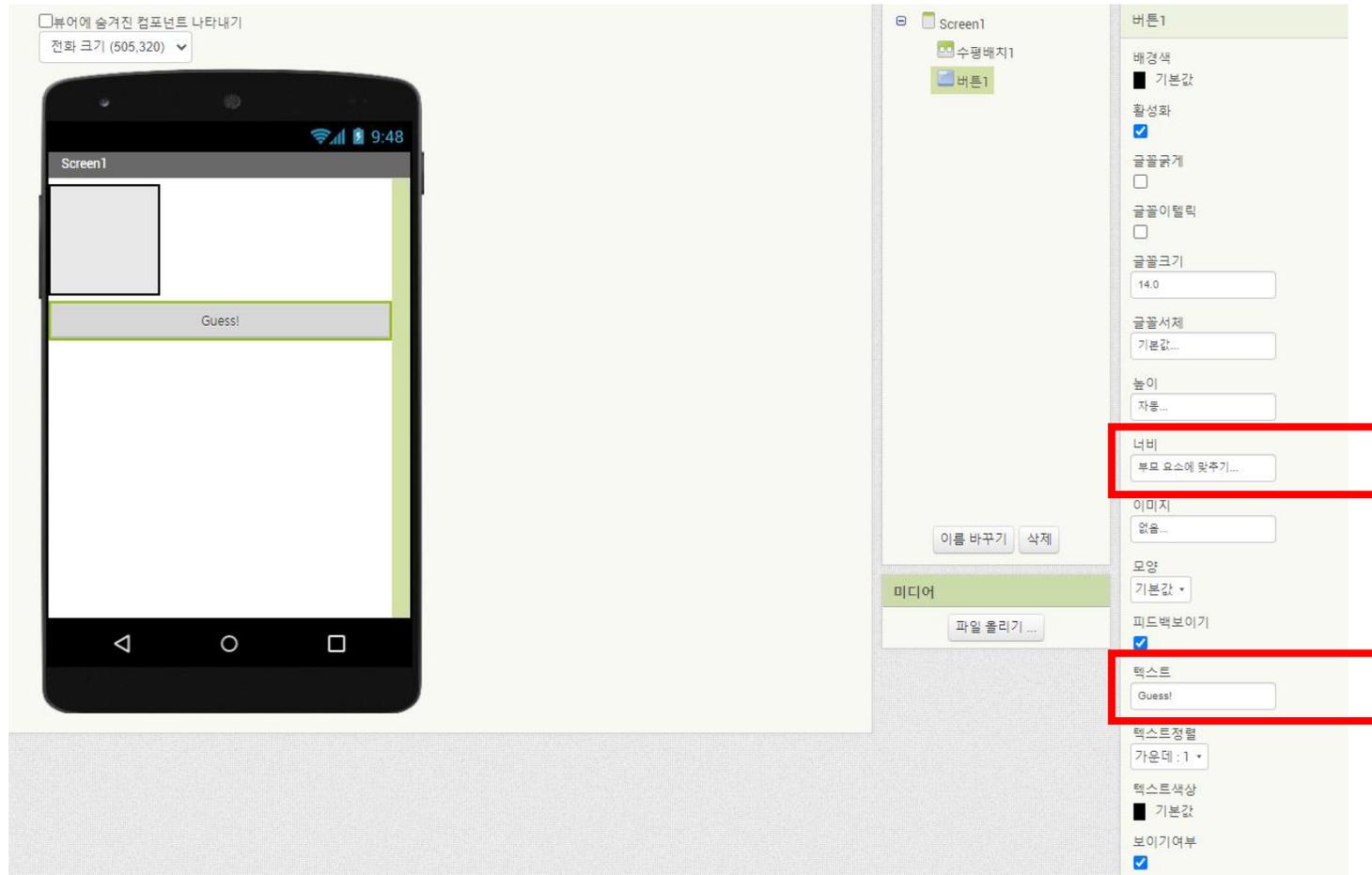
# 앱 제작의 실제4 – 숫자 예측 게임



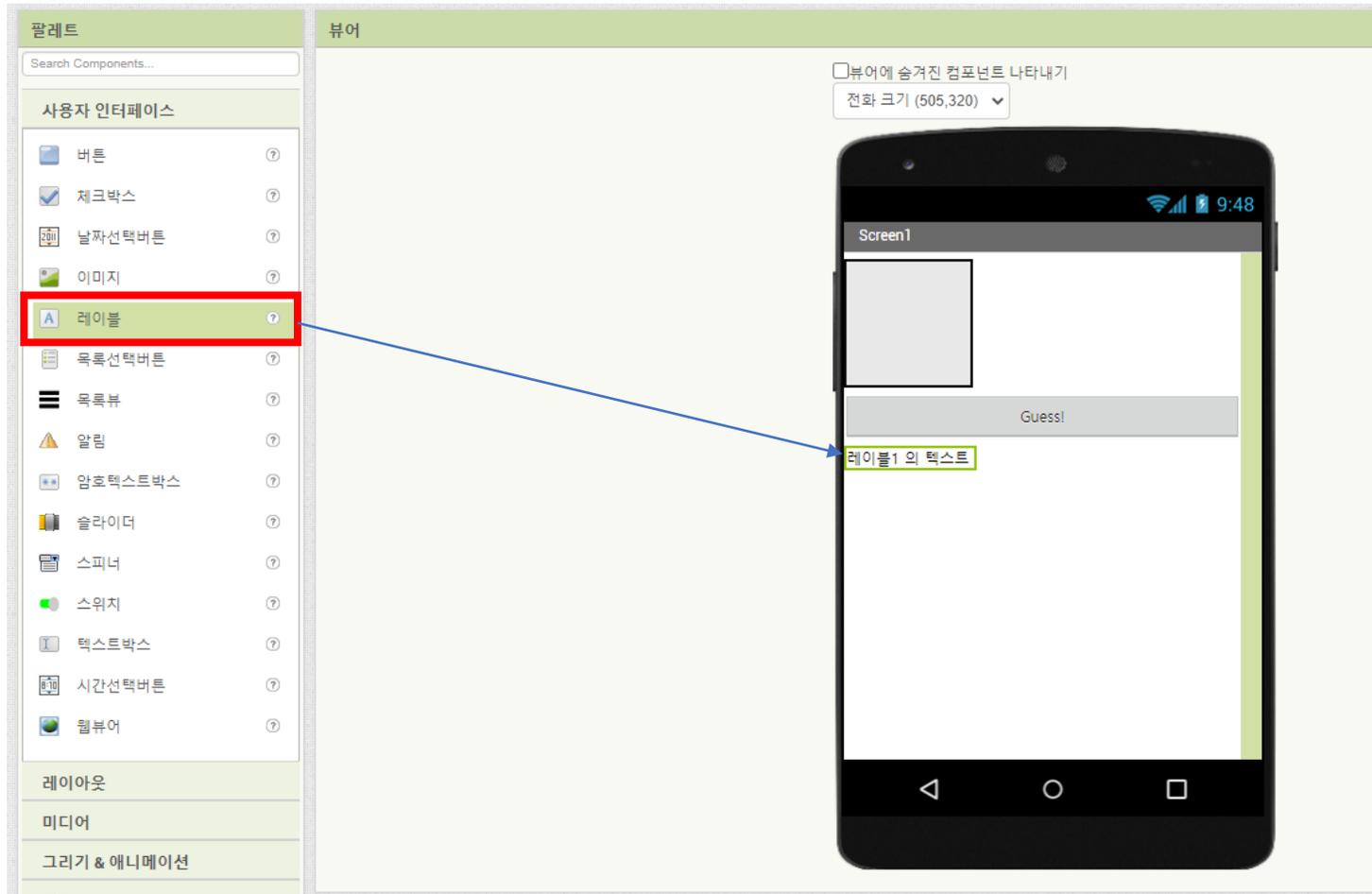
# 앱 제작의 실제4 – 숫자 예측 게임



# 앱 제작의 실제4 – 숫자 예측 게임



# 앱 제작의 실제4 – 숫자 예측 게임



# 앱 제작의 실제4 – 숫자 예측 게임



# 앱 제작의 실제4 – 숫자 예측 게임



# 앱 제작의 실제4 – 숫자 예측 게임

이 화면은 안드로이드 스튜디오의 인터페이스를 보여줍니다. 왼쪽에는 앱의 미리보기 화면이 표시되어 있으며, 화면 상단에는 'Screen1'이라는 제목과 '9:48'의 시간이 보입니다. 화면에는 '숫자를 입력하세요 :'라는 텍스트 입력 필드, 'Guess!' 버튼, 그리고 '여기에 힌트가 표시됩니다.'라는 텍스트가 포함되어 있습니다.

오른쪽에는 '컴포넌트' (Component Tree)와 '속성' (Properties) 패널이 있습니다. '컴포넌트' 패널에는 'Screen1' 하위에 '수평배치1', '레이블2', '텍스트박스1', '버튼1', '레이블1'이 나열되어 있습니다. '속성' 패널에서는 '레이블2'의 속성이 설정되어 있으며, '텍스트' 속성이 '숫자를 입력하세요 :'로 설정되어 있습니다. 이 속성 값은 빨간색 사각형으로 강조되어 있습니다.

# 앱 제작의 실제4 – 숫자 예측 게임

The image shows the Android Studio development environment for a number-guessing game. On the left, a virtual Android phone displays the app's interface. The screen contains the following text:

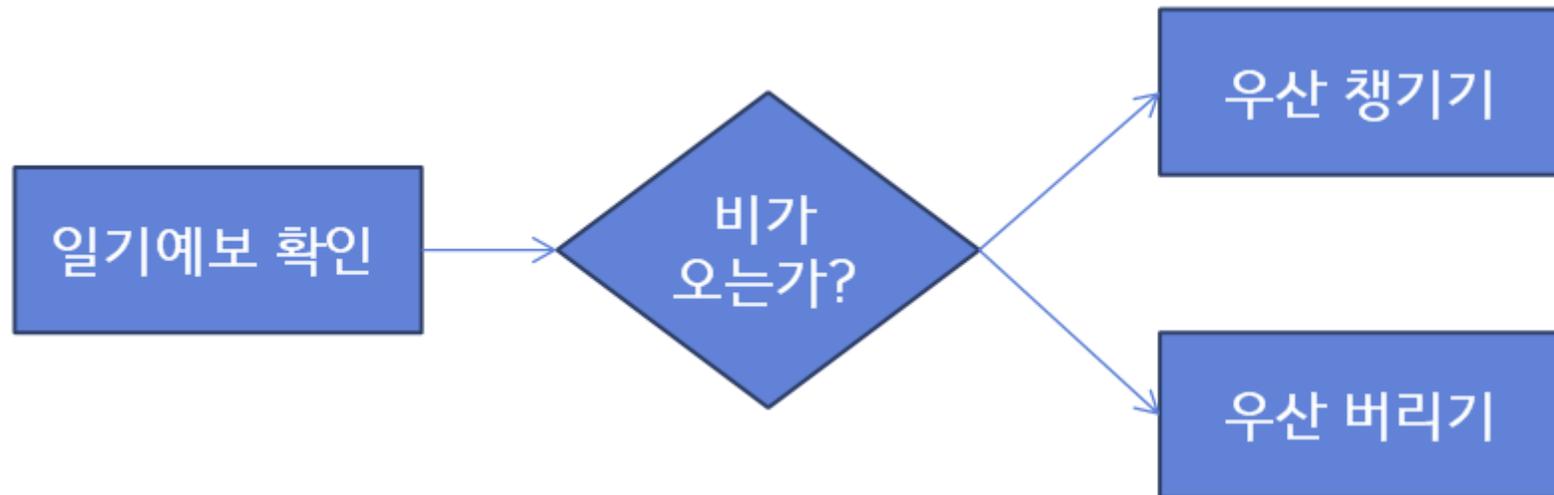
- Screen1
- 숫자를 입력하세요 : [input field]
- Guess!
- 여기에 힌트가 표시됩니다.

On the right, the 'Text View' properties panel for '텍스트박스1' is visible. The '힌트' (Hint) field is highlighted with a red box. Other visible settings include:

- 배경색: 기본값
- 활성화:
- 글꼴굵게:
- 글꼴이탤릭:
- 글꼴크기: 14.0
- 글꼴서체: 기본값...
- 높이: 자동...
- 너비: 자동...
- 힌트: [input field]
- 여러줄:
- 숫자만:
- ReadOnly:
- 텍스트: [input field]
- 텍스트정렬: 왼쪽: 0

# 선택구조

- 조건에 따라 다른 결과를 선택하게 되는 구조
- 만약 A라면 B이고, 그렇지 않으면 C이다.



# 앱 제작의 실제4 – 숫자 예측 게임

전역변수 만들기 **정답** 초기값 임의의 정수 시작 **1** 끝 **100**

언제 **버튼1** .클릭했을때

실행

- 만약 **텍스트박스1** . 텍스트 = 가져오기 **전역변수 정답**  
이러면 실행 지정하기 **레이블2** . 텍스트 값 축하합니다! 정답입니다.
- 만약 **텍스트박스1** . 텍스트 > 가져오기 **전역변수 정답**  
이러면 실행 지정하기 **레이블2** . 텍스트 값 정답보다 큼니다.
- 만약 **텍스트박스1** . 텍스트 < 가져오기 **전역변수 정답**  
이러면 실행 지정하기 **레이블2** . 텍스트 값 정답보다 작습니다.

## 앱 제작의 실제4 – 숫자 예측 게임

- 컴퓨터가 무작위로 추출한 숫자를 맞춰보세요!



## 앱 제작의 실제4 – 숫자 예측 게임

### QUIZ3

정해진 횟수(10번)만큼의 기회를 소진하면,  
게임오버가 되게 해보자.

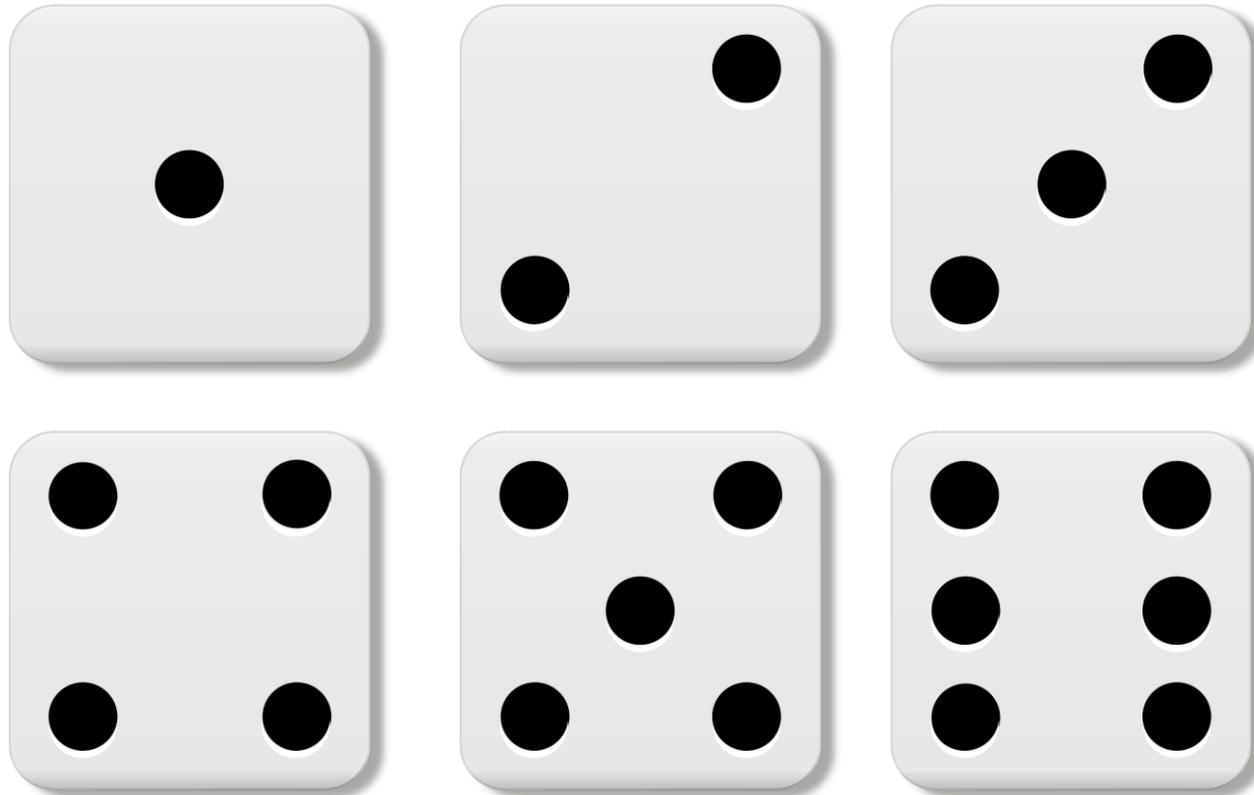
GAME OVER

## 앱 제작의 실제5 – 주사위 게임

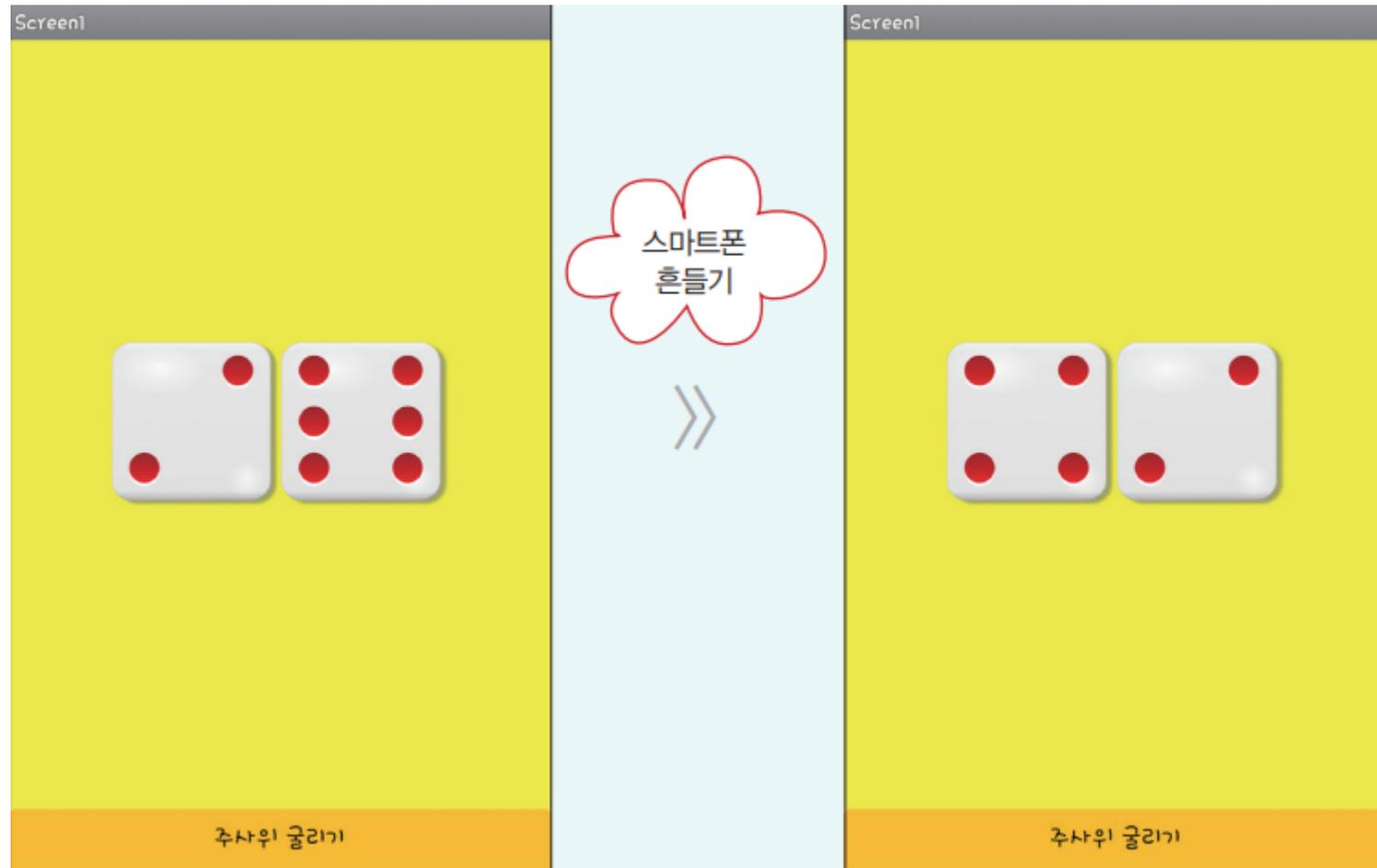
- 친구들과 여행을 와서 보드게임을 하려고 하는데, 주사위를 깜빡하고 놓고 왔다.
- 스마트폰으로 주사위를 만들 수 있을까?



# 앱 제작의 실제5 - 주사위 게임



# 앱 제작의 실제5 – 주사위 게임



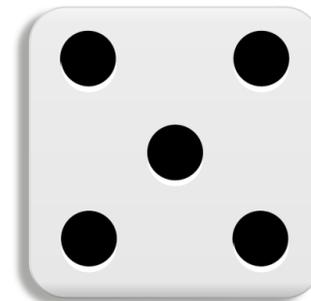
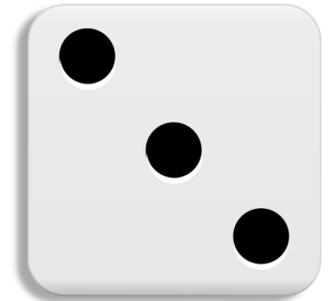
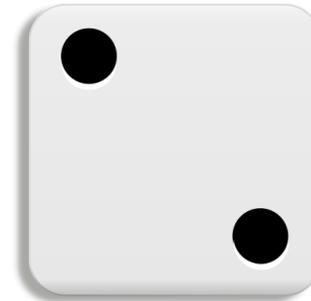
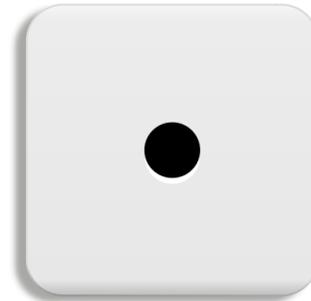
# 앱 제작의 실제5 – 주사위 게임

새로운 앱인벤퍼 프로젝트 만들기

프로젝트 이름:

## 앱 제작의 실제5 - 주사위 게임

- 주사위 이미지 다운받기
- 이미지 추가하기



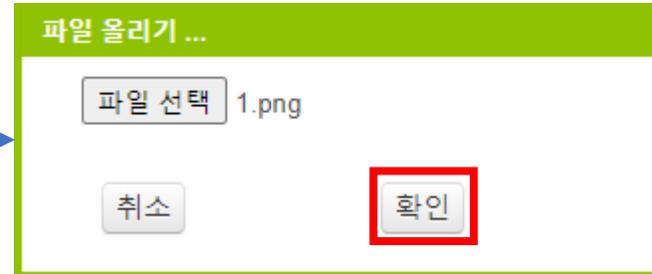
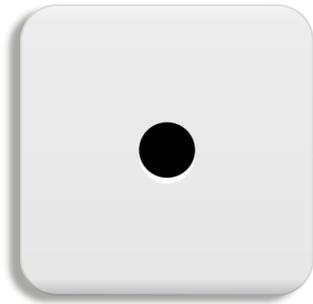
# 앱 제작의 실제5 – 주사위 게임



## 앱 제작의 실제5 – 주사위 게임



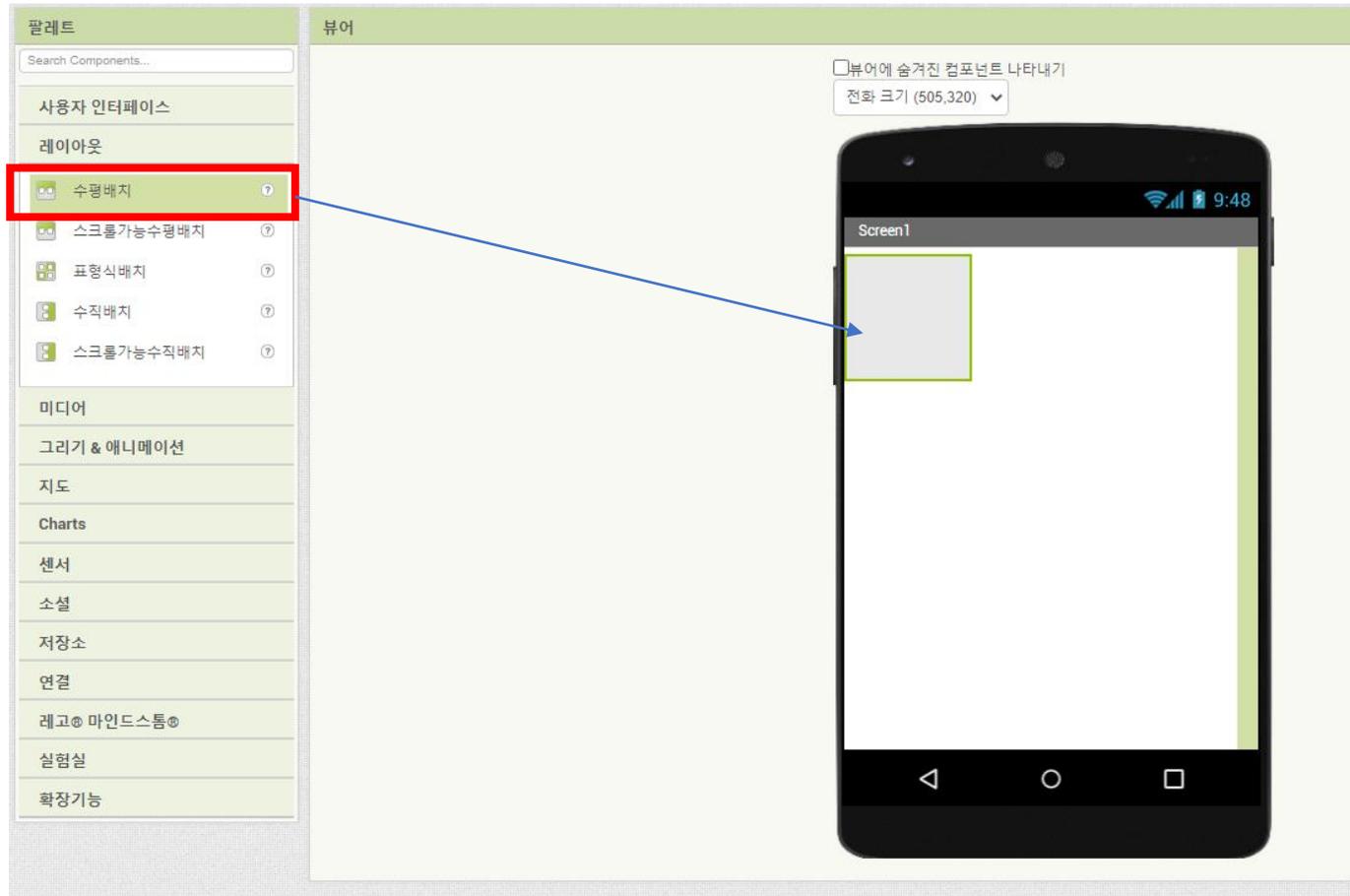
# 앱 제작의 실제5 - 주사위 게임



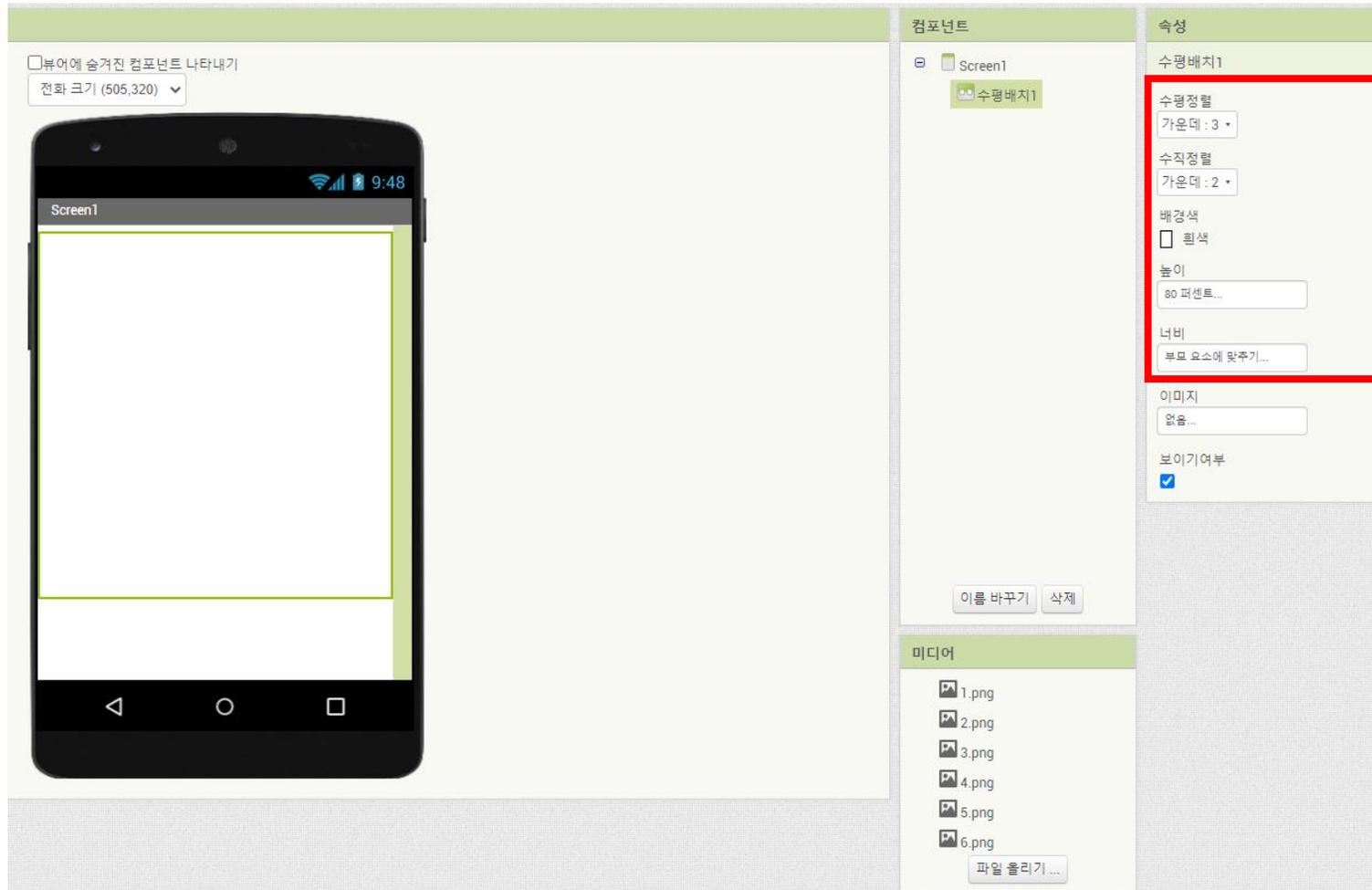
# 앱 제작의 실제5 – 주사위 게임



# 앱 제작의 실제5 – 주사위 게임



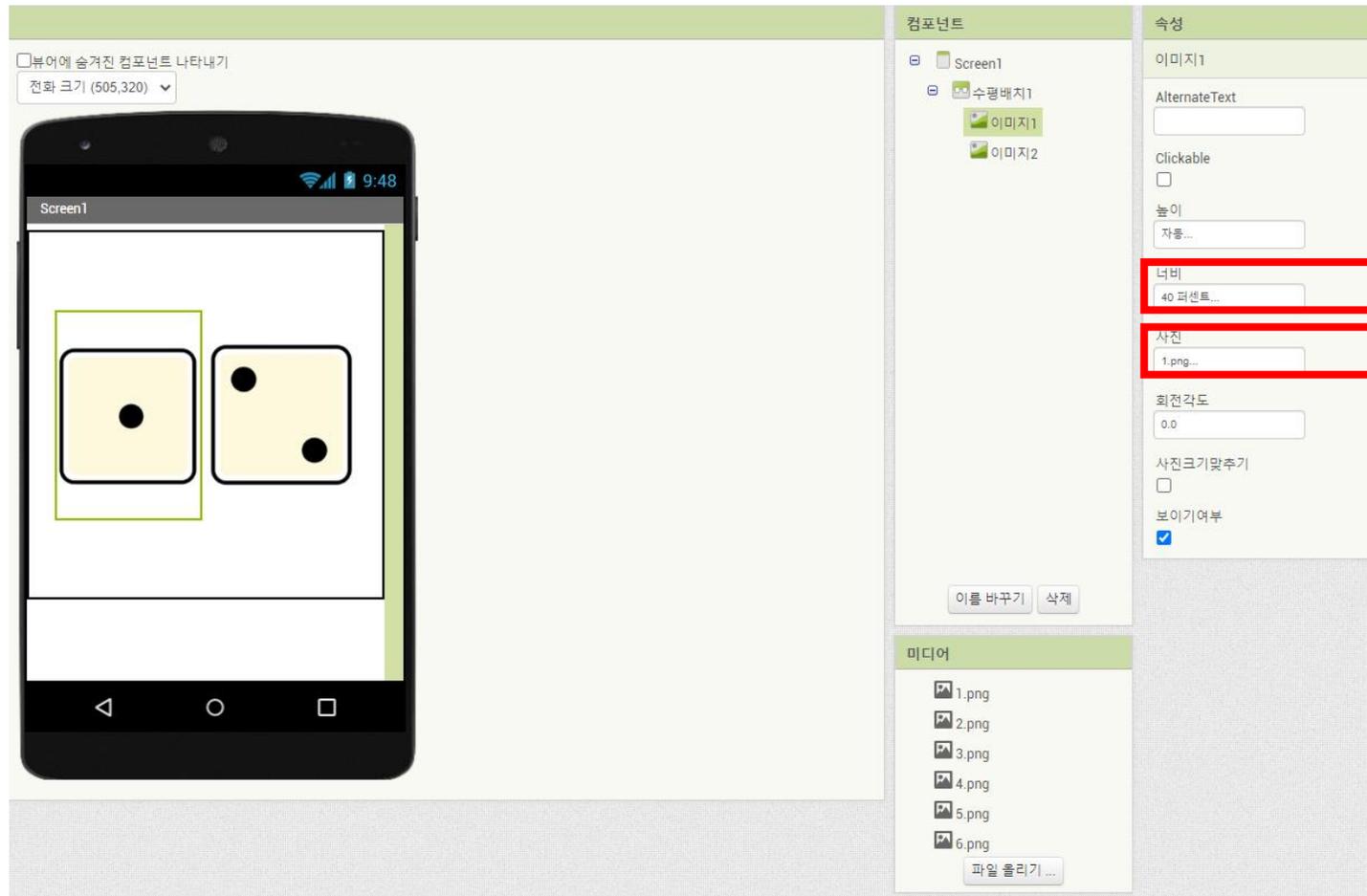
# 앱 제작의 실제5 – 주사위 게임



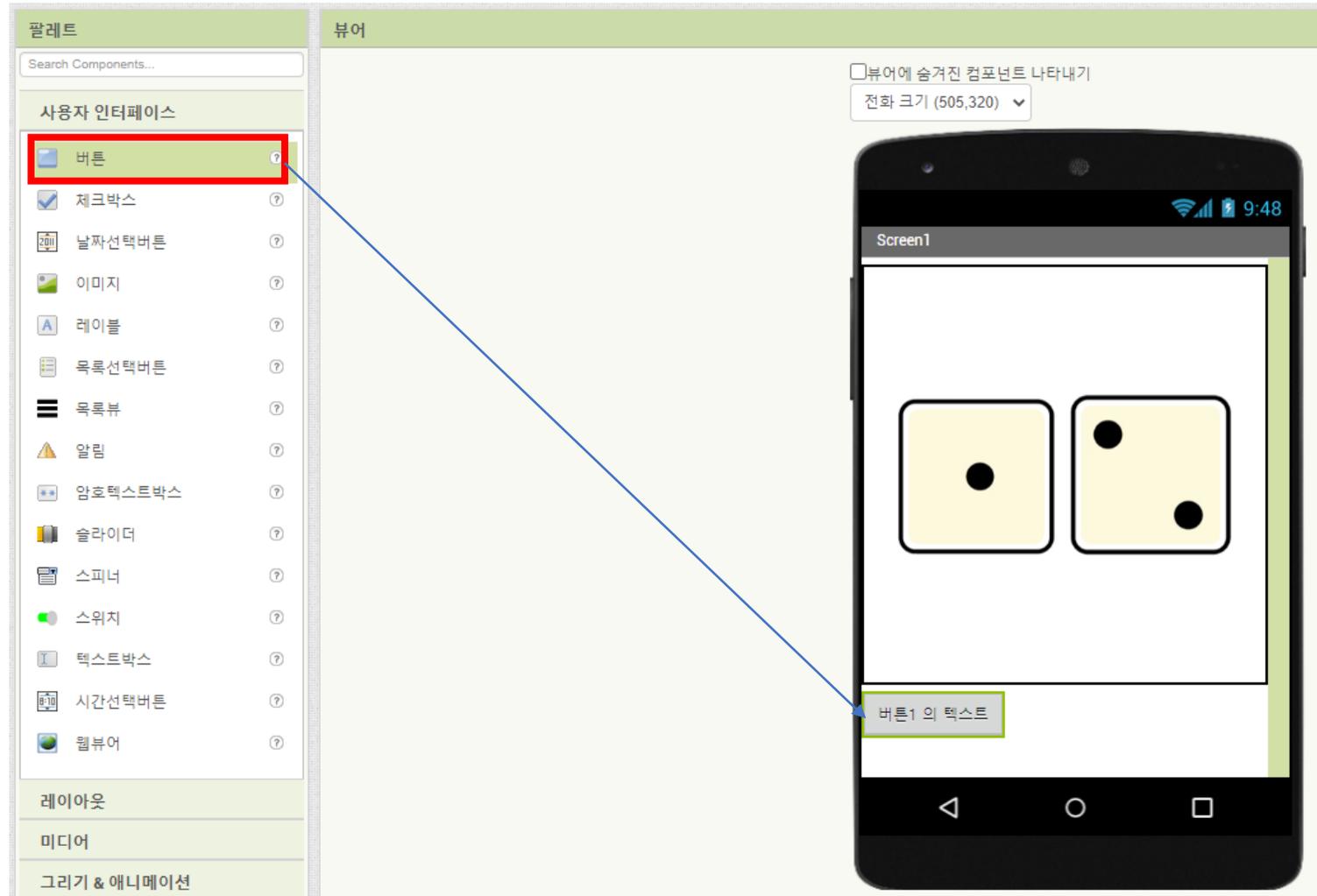
# 앱 제작의 실제5 – 주사위 게임



# 앱 제작의 실제5 – 주사위 게임



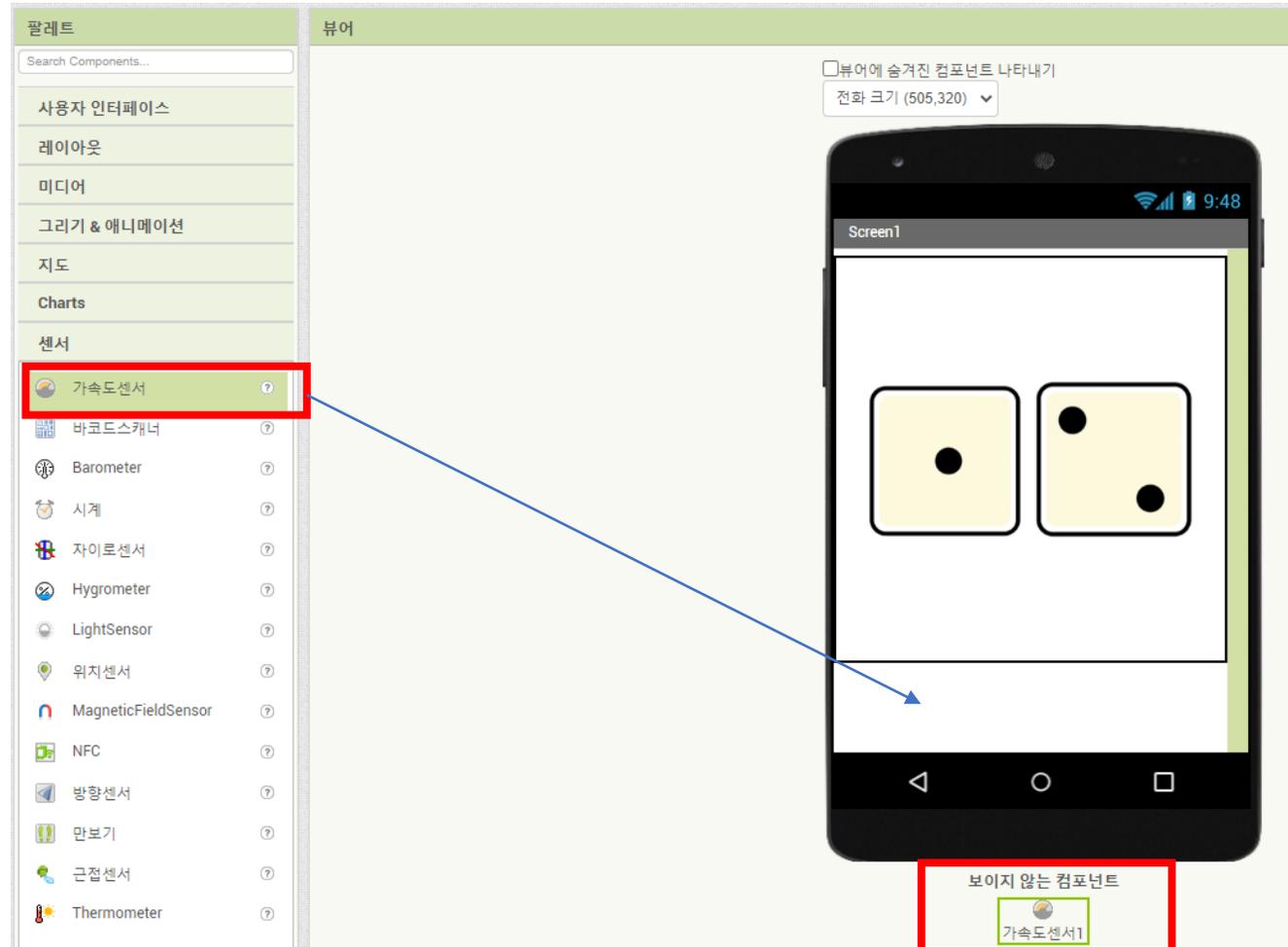
# 앱 제작의 실제5 – 주사위 게임



# 앱 제작의 실제5 – 주사위 게임



# 앱 제작의 실제5 – 주사위 게임



## 앱 제작의 실제5 – 주사위 게임

전역변수 만들기 주사위1 초기값 1

전역변수 만들기 주사위2 초기값 1



## 앱 제작의 실제5 – 주사위 게임

QUIZ4

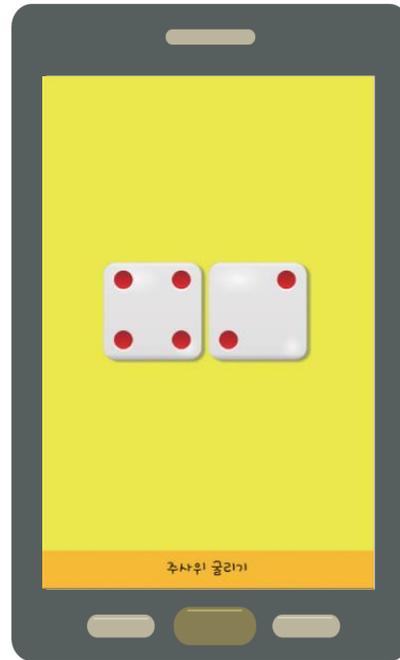
스마트폰을 흔들면 주사위의 눈이 바뀌도록  
만들어보자.

# 앱 제작의 실제5 – 주사위 게임

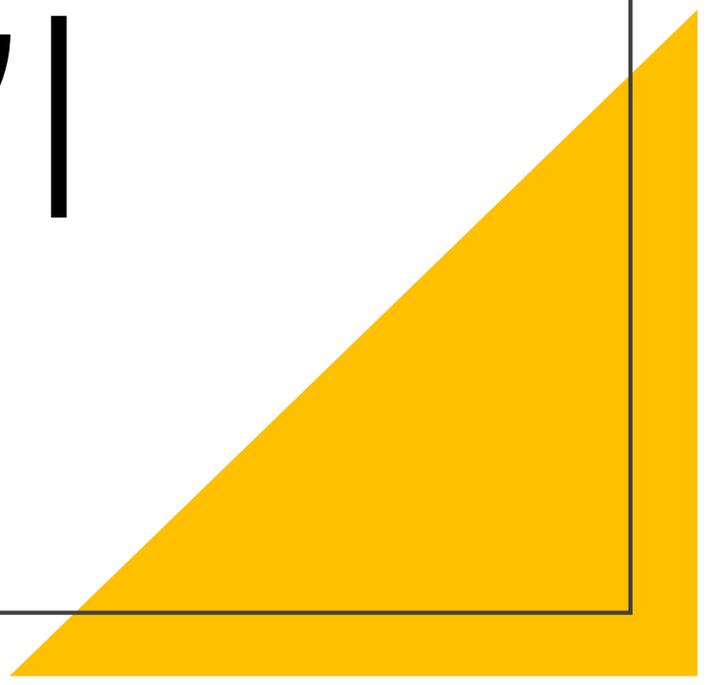


# 앱 제작의 실제5 - 주사위 게임

친구와 함께 주사위  
게임을 해봅시다 ^^

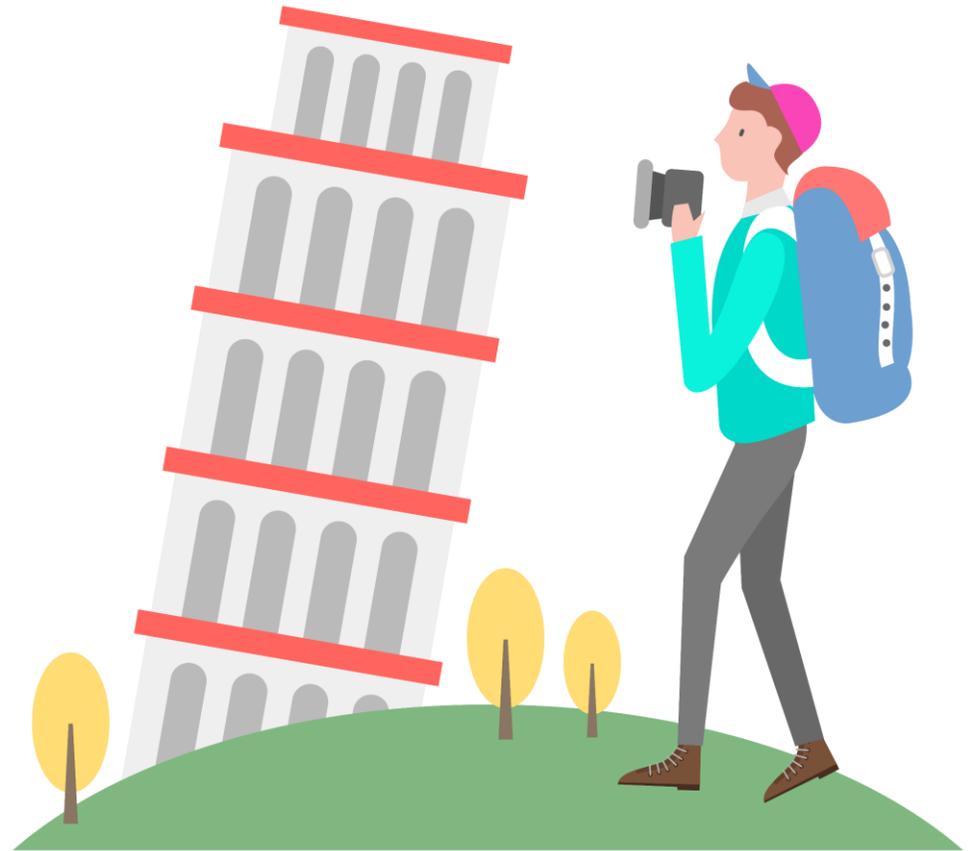


# 4. 번역 앱 만들기

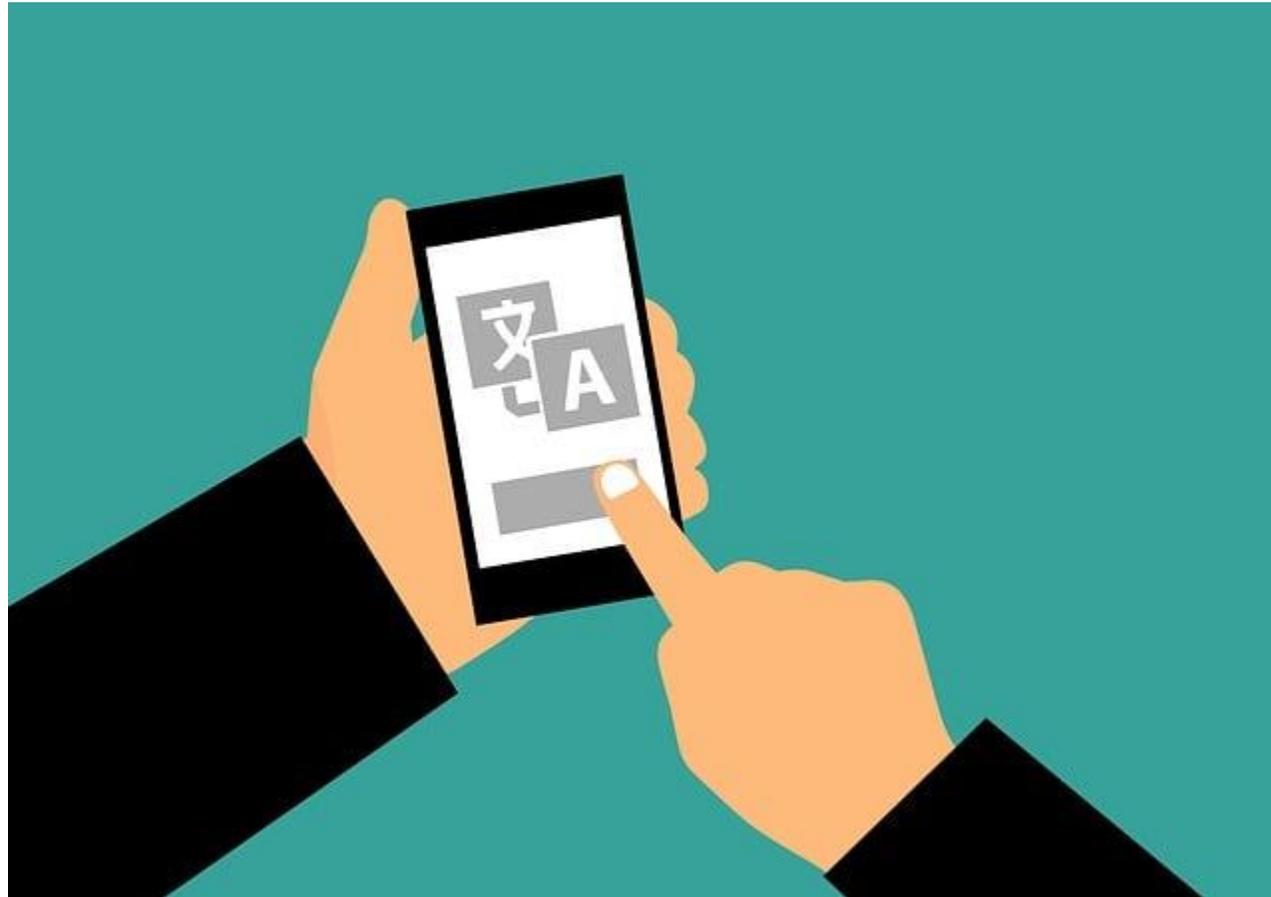


## 앱 제작의 실제6 – 번역기

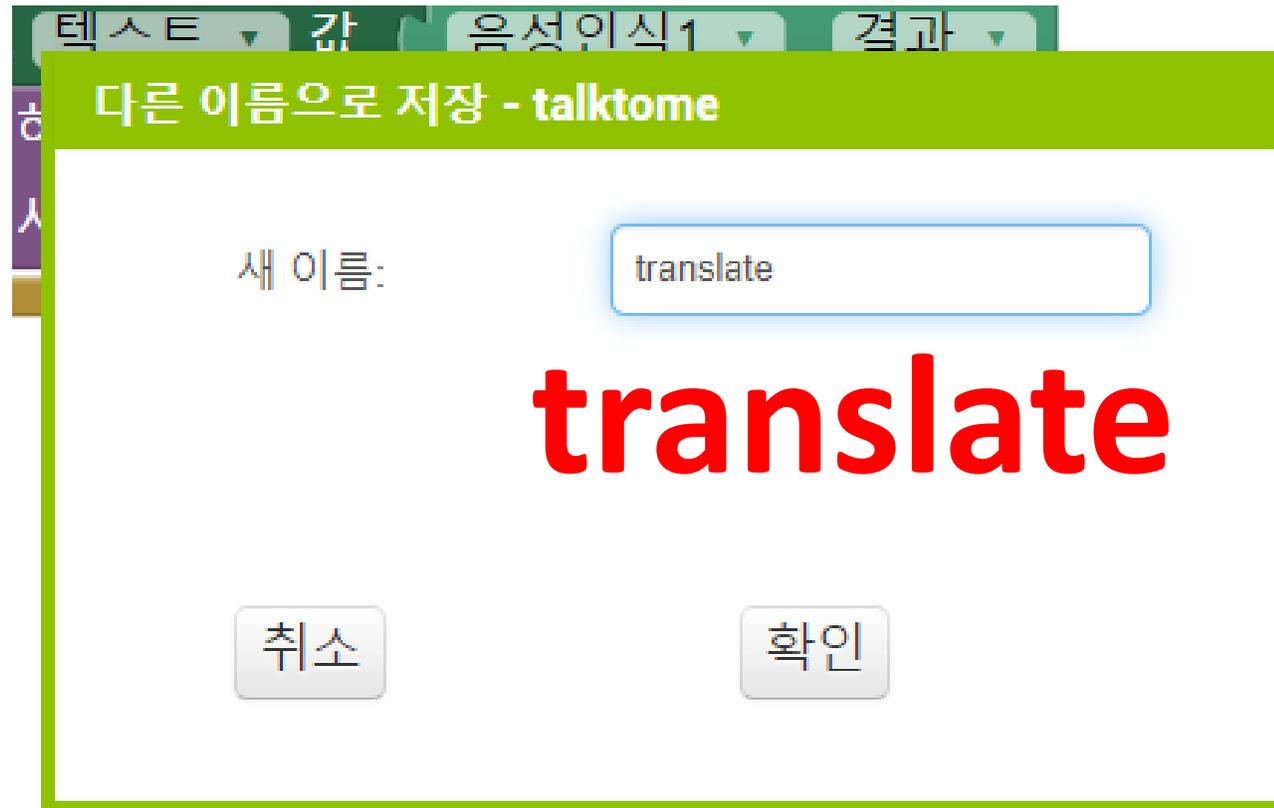
- 외국에 놀러갔는데 그 나라 말을 모른다.
- 믿었던 파X고가 작동하지 않는다.
- 외국 친구들과 말하고 싶다!



## 앱 제작의 실제6 – 번역기



## 앱 제작의 실제6 – 번역기



## 앱 제작의 실제6 – 번역기

속성

Screen1

스크린설명

강조색  
 기본값

수평정렬  
가운데 : 3 ▾

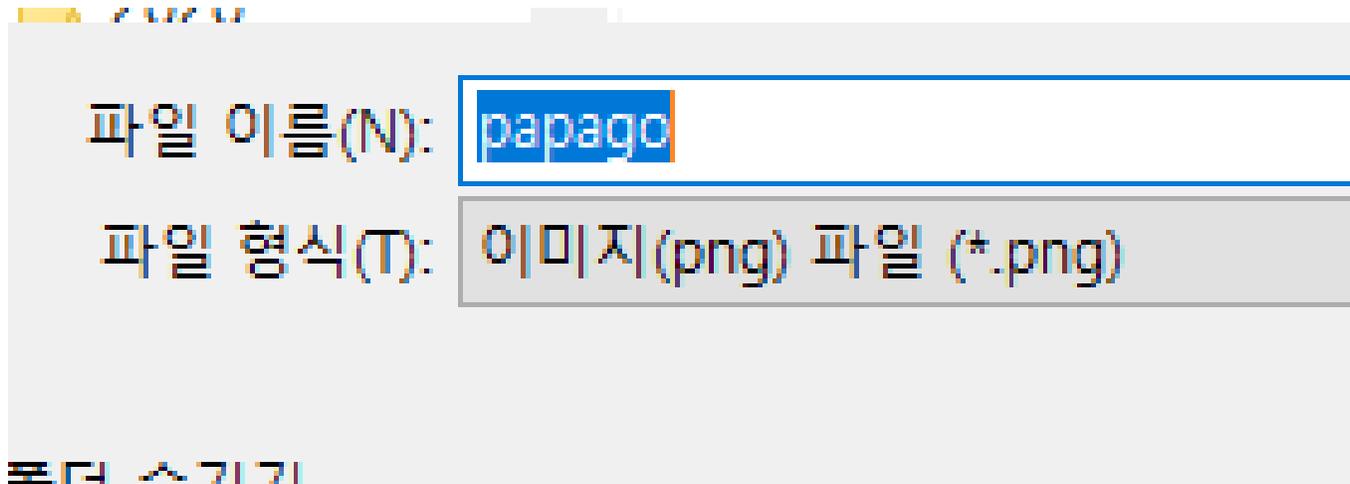
수직정렬  
위 : 1 ▾

앱이름

배경색

translate

# 영어로 저장



## 앱 제작의 실제6 – 번역기



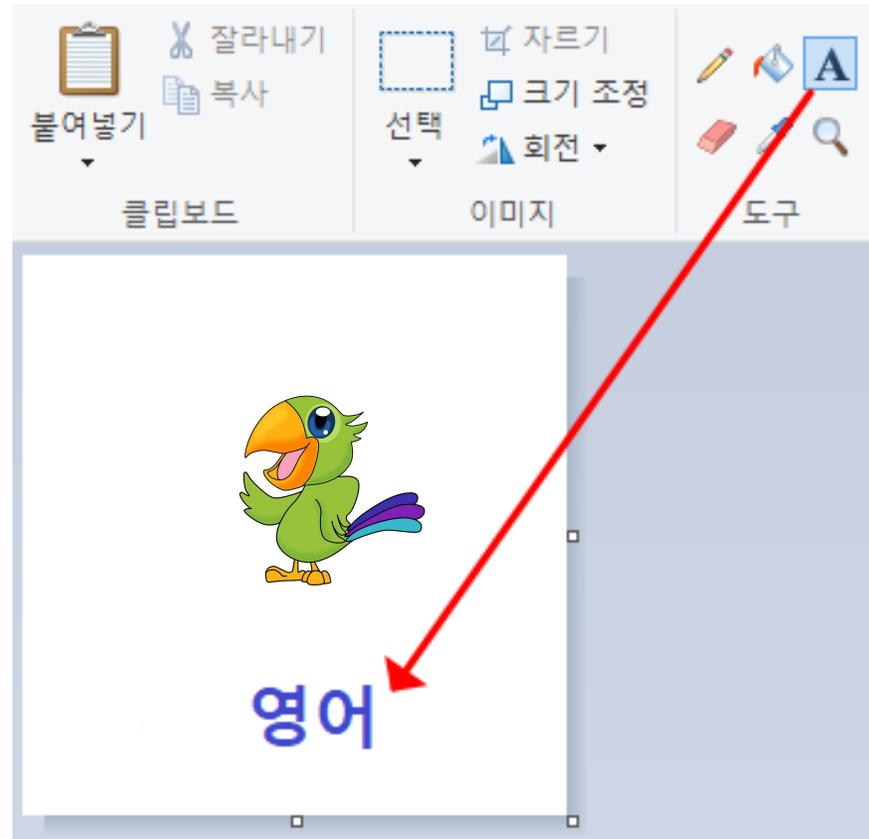
# 아이콘 교체

## 앱 제작의 실제6 – 번역기

# 그림판에서 글자 삭제

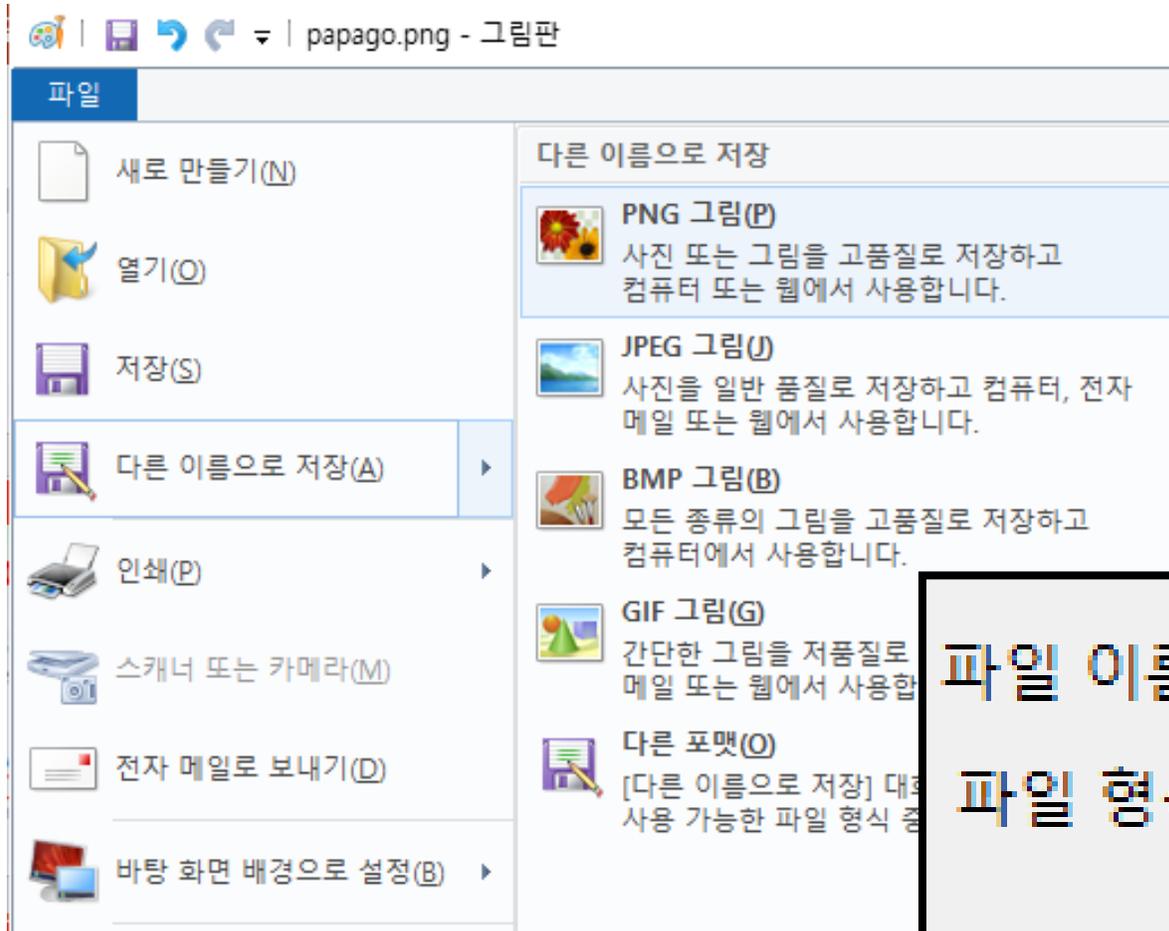


## 앱 제작의 실제6 – 번역기



‘영어’

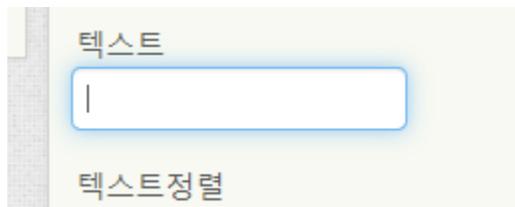
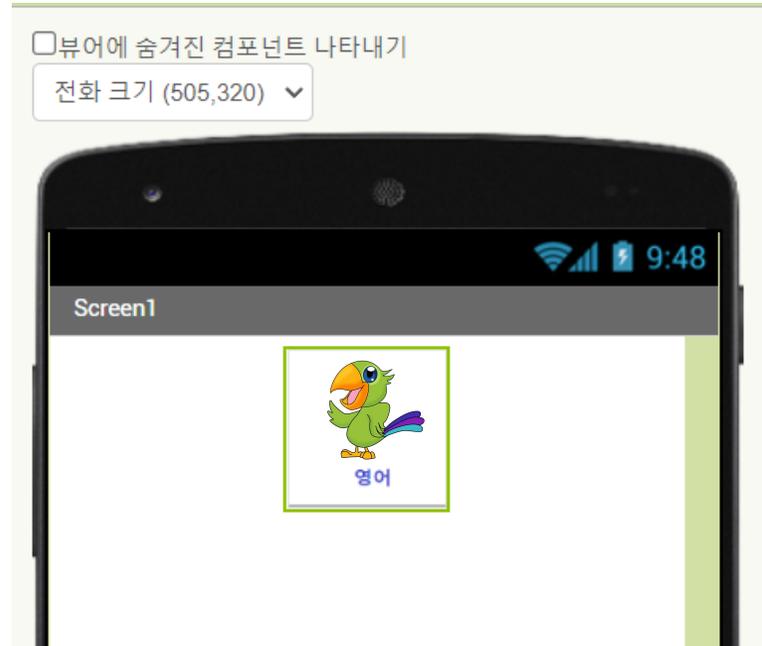
# 앱 제작의 실제6 – 번역기



파일 이름(N): en

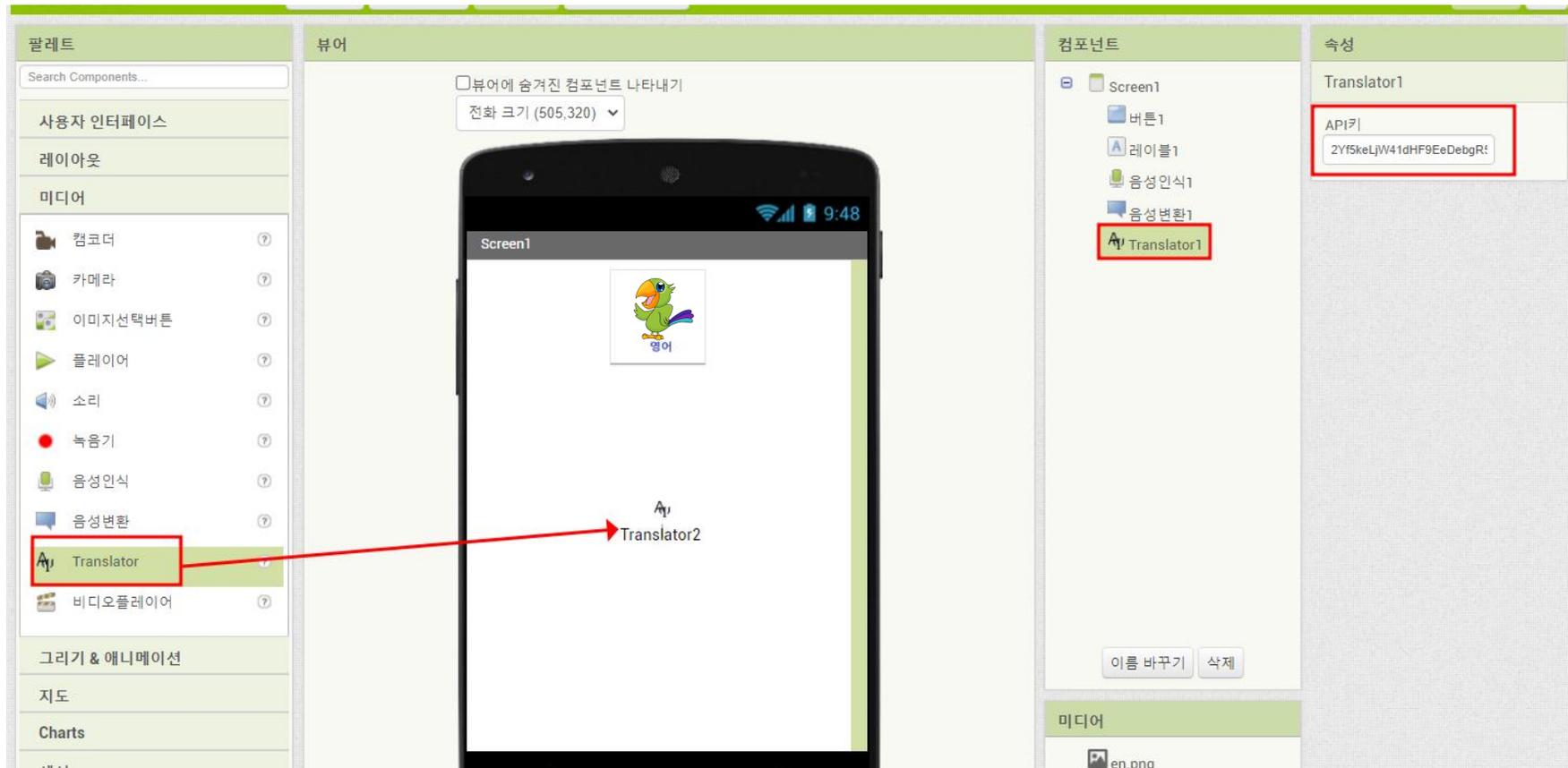
파일 형식(T): PNG (\*.png)

## 앱 제작의 실제6 – 번역기

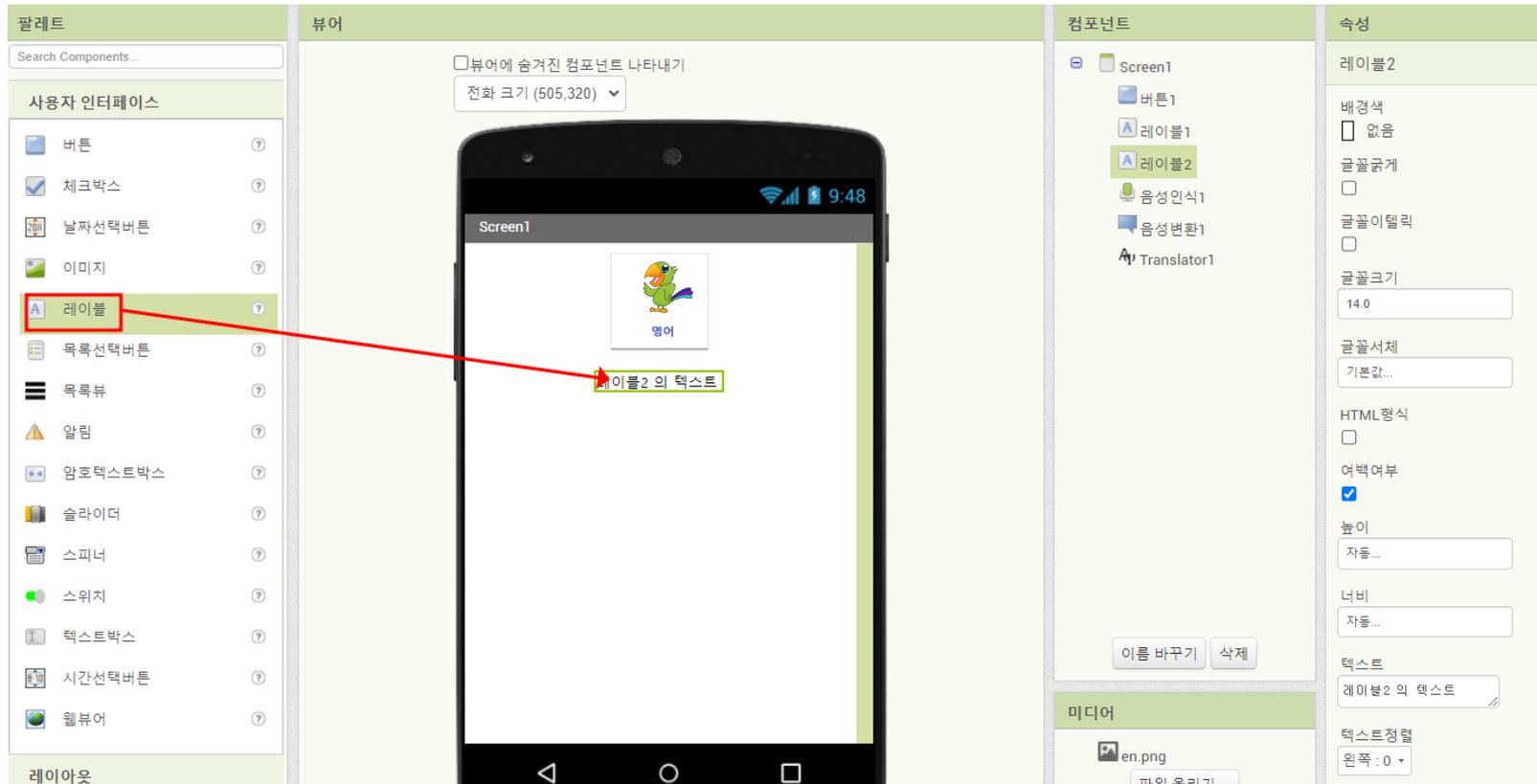


**버튼 이미지 바꾸기**  
**80픽셀**  
**80픽셀**  
**텍스트 삭제**

# 앱 제작의 실제6 – 번역기



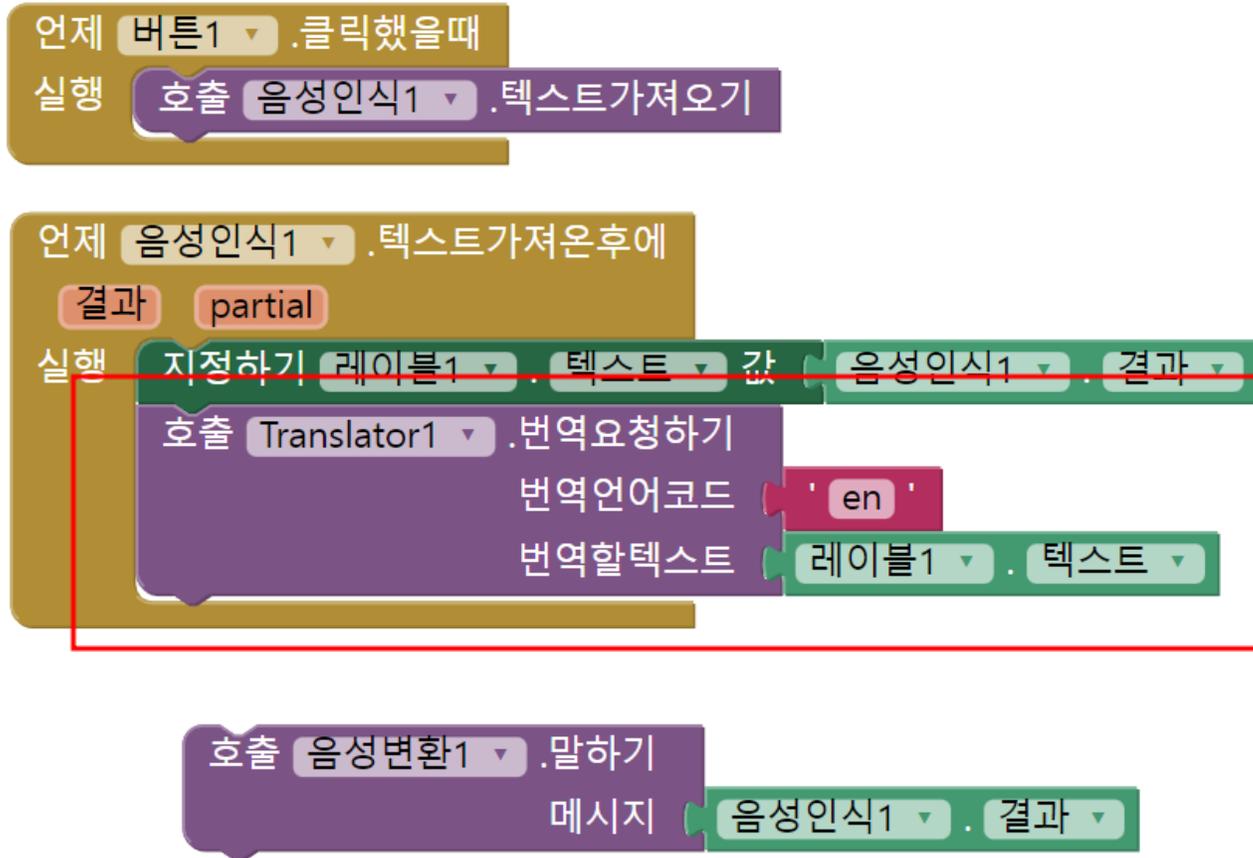
# 앱 제작의 실제6 – 번역기



# 앱 제작의 실제6 – 번역기



# 앱 제작의 실제6 – 번역기



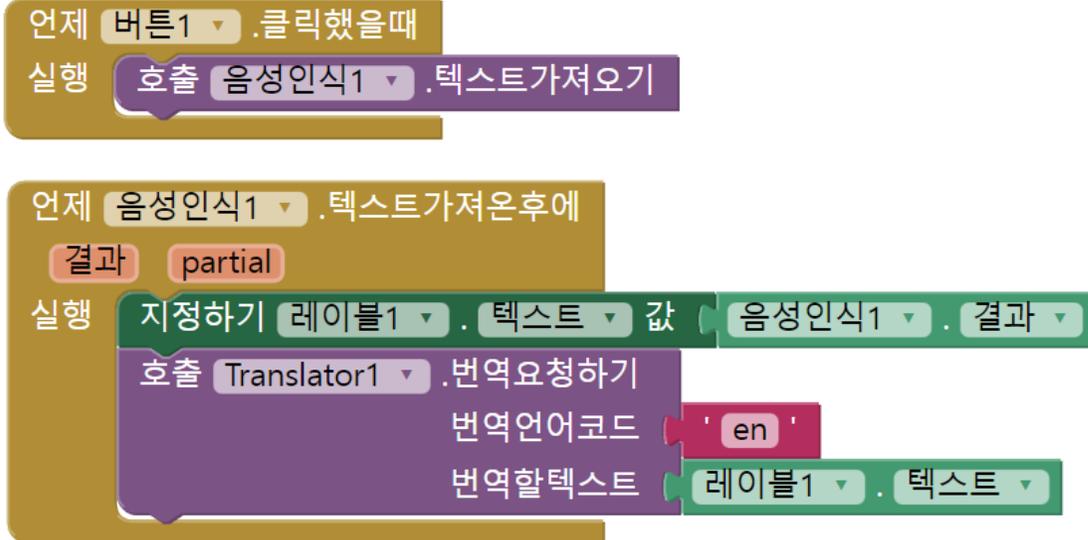
Basque (바스크어)	eu	Xhosa (코사어)	xh	Swahili (스와힐리어)	sw
Bashkir (바쉬키르어)	ba	Khmer (크메르어)	km	Sundanese (순다어)	su
Belarusian (벨로루시어)	be	Laotian (라오스어)	lo	Tajik (타지크어)	tg
Bengali (벥골어)	bn	Latin (라틴어)	la	Thai (타이어)	th
Burmese (버마어)	my	Latvian (라트비아어)	lv	Tagalog (타갈로그어)	tl
Bulgarian (불가리아어)	bg	Lithuanian (리투아니아어)	lt	Tamil (타밀어)	ta
Bosnian (보스니아어)	bs	Luxembourgish (룩셈부르크어)	lb	Tatar (타타르어)	tt
Welsh (웨일스어)	cy	Malagasy (마다가스카르어)	mg	Telugu (텔루구어)	te
Hungarian (헝가리어)	hu	Malay (말레이어)	ms	Turkish (터키어)	tr
Vietnamese (베트남어)	vi	Malayalam (말라얄람어)	ml	Udmurt (우드무르트어)	udm
Haitian/Creole (아이티어/크레올)	ht	Maltese (몰티즈어)	mt	Uzbek (우즈베크어)	uz
Galician (갈리시아어)	gl	Macedonian (마케도니아어)	mk	Ukrainian (우크라이나어)	uk
Dutch (네덜란드어)	nl	Maori (마오리어)	mi	Urdu (우르두어)	ur
Hill Mari (힐마리어)	mrj	Marathi (마라티어)	mr	Finnish (핀란드어)	fi
Greek (그리스어)	el	Mari (마리어)	mhr	French (프랑스어)	fr
Georgian (조르지아어)	ka	Mongolian (몽고어)	mn	Hindi (힌디어)	hi
Gujarati (구자라트어)	gu	German (독일어)	de	Croatian (크로아티어)	hr

언어	코드	언어	코드	언어	코드
Azerbaijan (아제르바이잔어)	az	Spanish (스페인어)	es	Romanian (루마니아어)	ro
Albanian (알바니아어)	sq	Kazakh (카자흐어)	kk	Russian (러시아어)	ru
Amharic (암하라어)	am	Kannada (칸나다어)	kn	Cebuano (세부아노어)	ceb
English (영어)	en	Catalan (카탈로니아어)	ca	Serbian (세르비아어)	sr
Arabic (아랍어)	ar	Kyrgyz (키르기스어)	ky	Sinhala (싱할라어)	si
Armenian (아르메니아어)	hy	Chinese (중국어)	zh	Slovakian (슬로바키아어)	sk
Afrikaans (아프리카어)	af	Korean (한국어)	ko	Slovenian (슬로베니아어)	sl
Danish (덴마크어)	da	Nepali (네팔어)	ne	Czech (체코어)	cs
Hebrew (히브리어)	he	Norwegian (노르웨이어)	no	Swedish (스웨덴어)	sv
Yiddish (이디시어)	yi	Punjabi (펀자브어)	pa	Scottish (스코틀랜드어)	gd
Indonesian (인도네시아어)	id	Papiamento (파피아멘토어)	pap	Estonian (에스토니아어)	et
Irish (아일랜드어)	ga	Persian (페르시아어)	fa	Esperanto (에스페란토어)	eo
Italian (이탈리아어)	it	Polish (폴란드어)	pl	Javanese (자바어)	jv
Icelandic (아이슬란드어)	is	Portuguese (포르투갈어)	pt	Japanese (일본어)	ja

## 앱 제작의 실제6 – 번역기

언어	코드
영어	en
중국어	zh
스페인어	es
이탈리아어	it
아랍어	ar

# 앱 제작의 실제6 – 번역기



발음 조정

블록 오류가 뜨면 드롭다운  
눌러 다시 선택

# 앱 제작의 실제6 – 번역기



언제 버튼1 ▾ .클릭했을때  
실행 호출 음성인식1 ▾ .텍스트가져오기

언제 음성인식1 ▾ .텍스트가져온후에  
결과 partial  
실행 지정하기 레이블1 ▾ . 텍스트 ▾ 값 음성인식1 ▾ . 결과 ▾  
호출 Translator1 ▾ .번역요청하기  
번역언어코드 'en'  
번역할텍스트 레이블1 ▾ . 텍스트 ▾

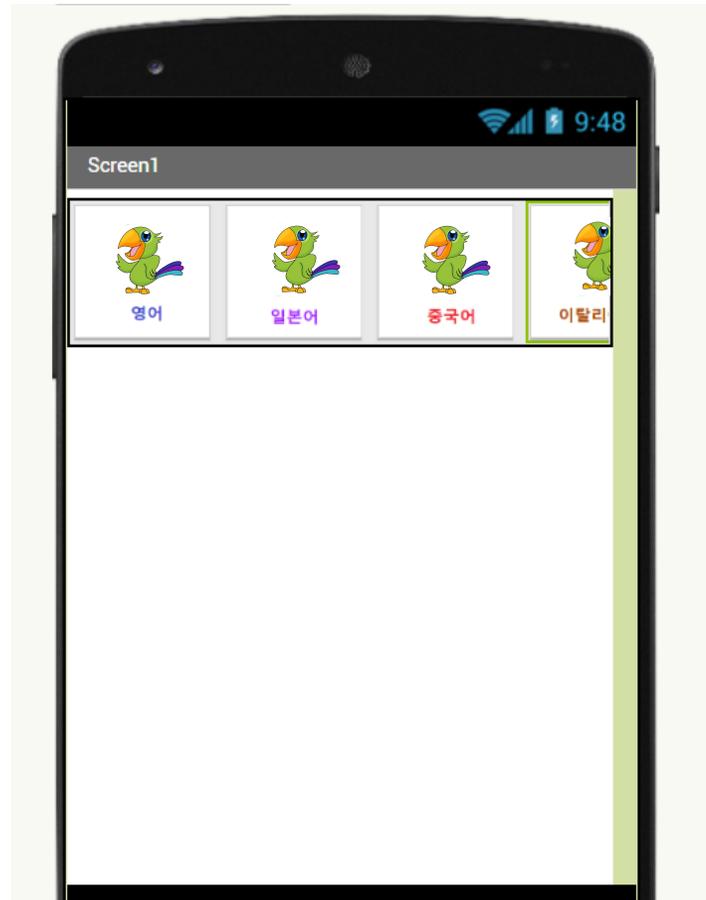
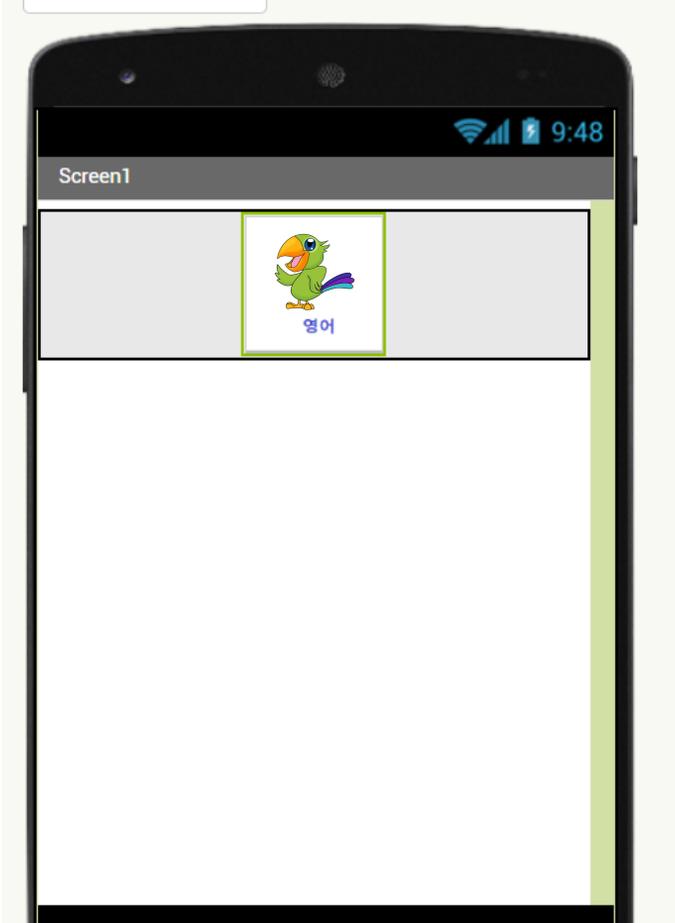
언제 Translator1 ▾ .번역을받았을때  
응답코드 번역  
실행 지정하기 음성변환1 ▾ . 언어 ▾ 값 'en'  
호출 음성변환1 ▾ .말하기  
메시지 가져오기 번역 ▾  
지정하기 레이블2 ▾ . 텍스트색상 ▾ 값   
시정하기 레이블2 ▾ . 텍스트 ▾ 값 가져오기 번역 ▾

# 앱 제작의 실제6 – 번역기

The screenshot displays the Axure RP software interface. On the left, the '팔레트' (Palette) shows the '수평배치' (Horizontal Layout) component selected. The central '뷰어' (Viewer) shows a mobile phone screen with a 'Screen1' header and a '영어' (English) button. The right side features the '컴포넌트' (Components) list and the '속성' (Properties) panel. The '속성' panel for '수평배치1' shows the '너비' (Width) property set to '부모 요소에 맞추기...' (Match parent element...).

This close-up view of the '속성' (Properties) panel for '수평배치1' shows the '너비' (Width) property. The '너비' dropdown is set to '가운데 : 3' (Center : 3), which is highlighted with a red box. Other visible properties include '수평배치1', '수평정렬' (Horizontal Alignment) set to '왼쪽 : 1' (Left : 1), and '오른쪽 : 2' (Right : 2).

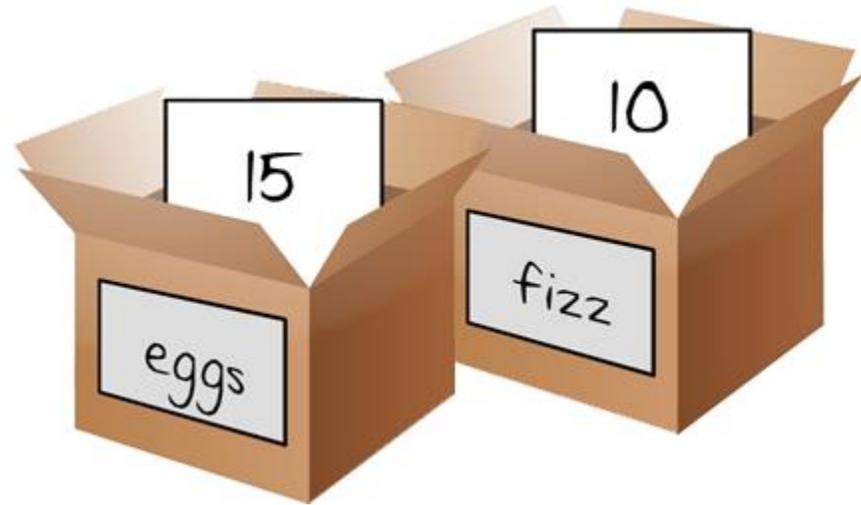
# 앱 제작의 실제6 – 번역기



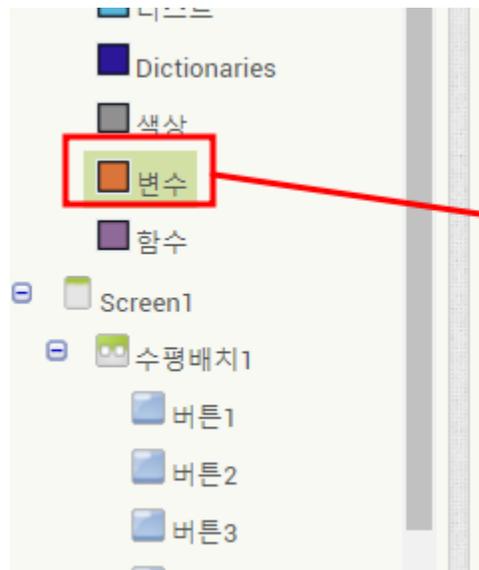
## 앱 제작의 실제6 – 번역기

### 변수

- 데이터를 저장하는 공간
- 상자와 같다.



## 앱 제작의 실제6 – 번역기

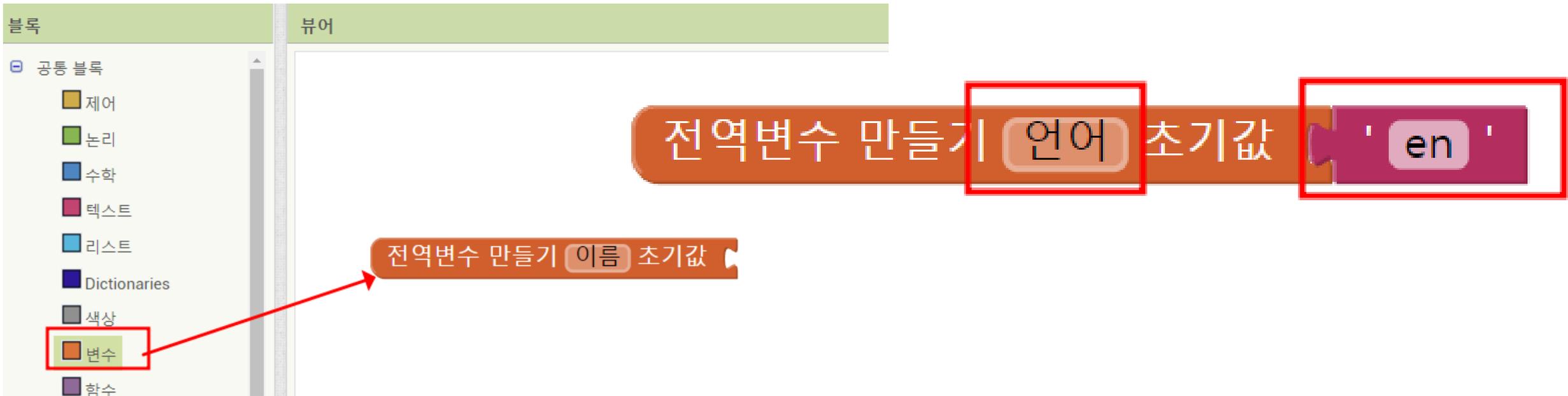


전역변수 만들기 이름 초기값

언제 버튼1 .클릭했을때

실행 호출 음성인식1 .텍스트가져오기

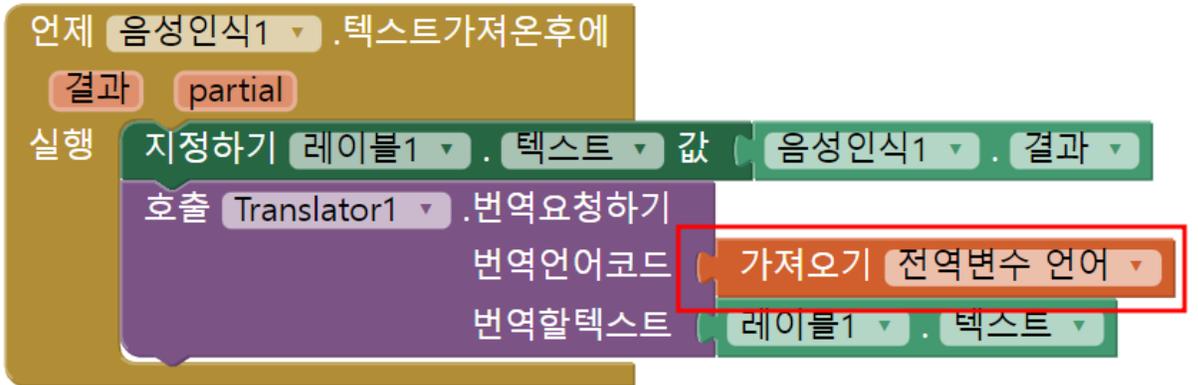
# 앱 제작의 실제6 – 번역기



# 앱 제작의 실제6 – 번역기

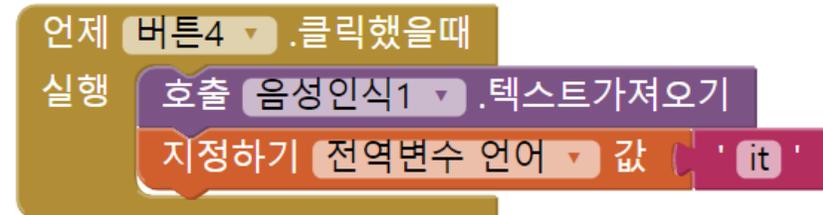
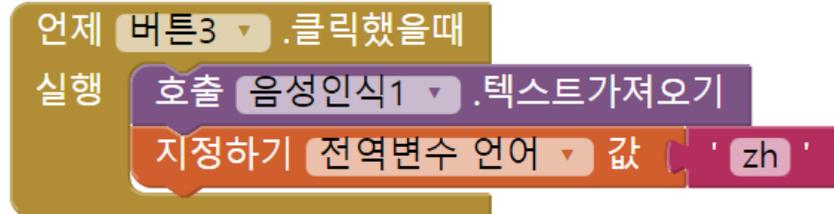
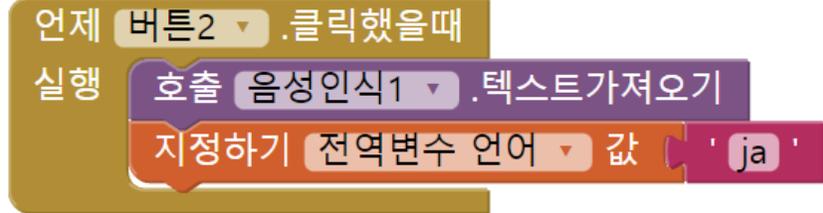
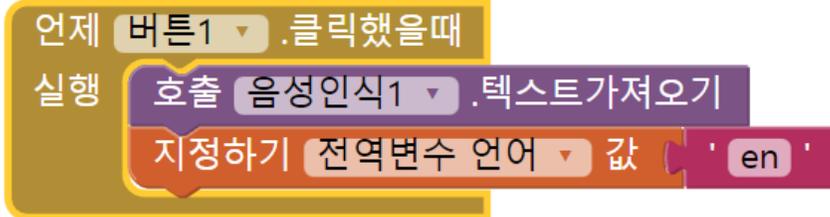


언제 **버튼1** .클릭했을때  
실행 **호출 음성인식1** .텍스트가져오기  
**지정하기 전역변수 언어** 값 **'en'**  
언제 **Translator1** .번역을받았을때  
**응답코드 번역**  
실행 **지정하기 음성변환1** . 언어 값 **가져오기 전역변수 언어**  
**호출 음성변환1** .말하기  
**메시지 가져오기 번역**  
**지정하기 레이블2** . 텍스트색상 값 **■**  
**지정하기 레이블2** . 텍스트 값 **가져오기 번역**



언제 **음성인식1** .텍스트가져온후에  
**결과 partial**  
실행 **지정하기 레이블1** . 텍스트 값 **음성인식1** . 결과  
**호출 Translator1** .번역요청하기  
**번역언어코드 가져오기 전역변수 언어**  
**번역할텍스트 레이블1** . 텍스트

# 앱 제작의 실제6 – 번역기



# 앱 제작의 실제6 – 번역기

엔트리 시작하기

6

## 변수와 리스트 비교

변수

변수 이름

리스트

리스트 이름

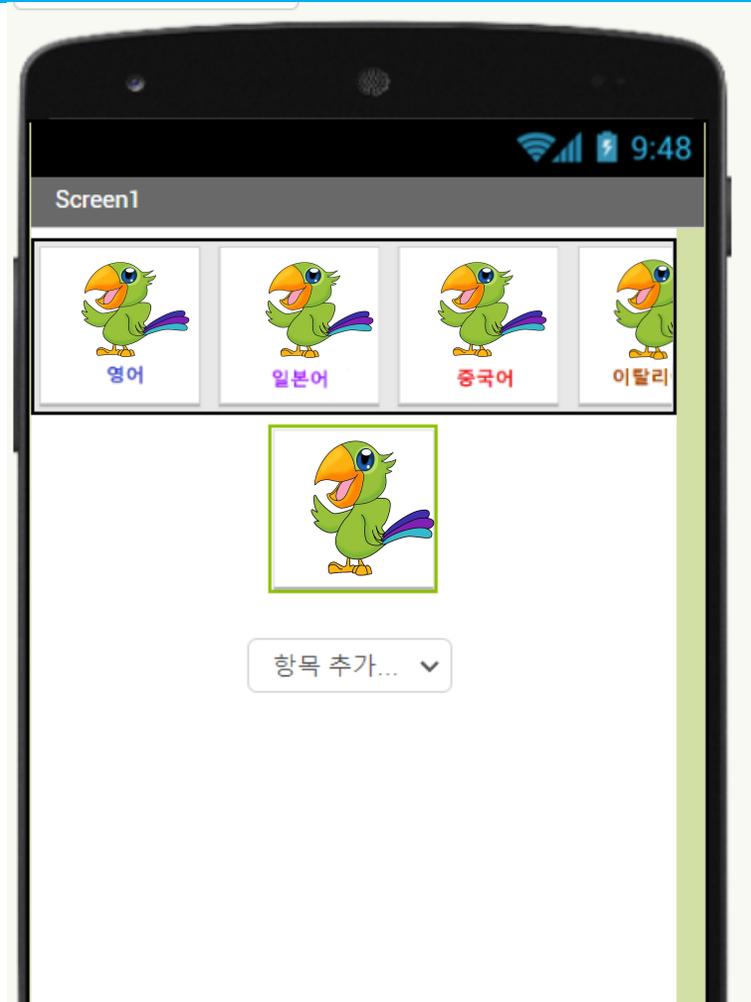
1	2	3	4	5	6	7	8	...
<input type="text"/>	...							

리스트는 여러 개의 정보를 저장할 수 있는 공간

# 앱 제작의 실제6 – 번역기



# 앱 제작의 실제6 – 번역기

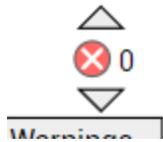
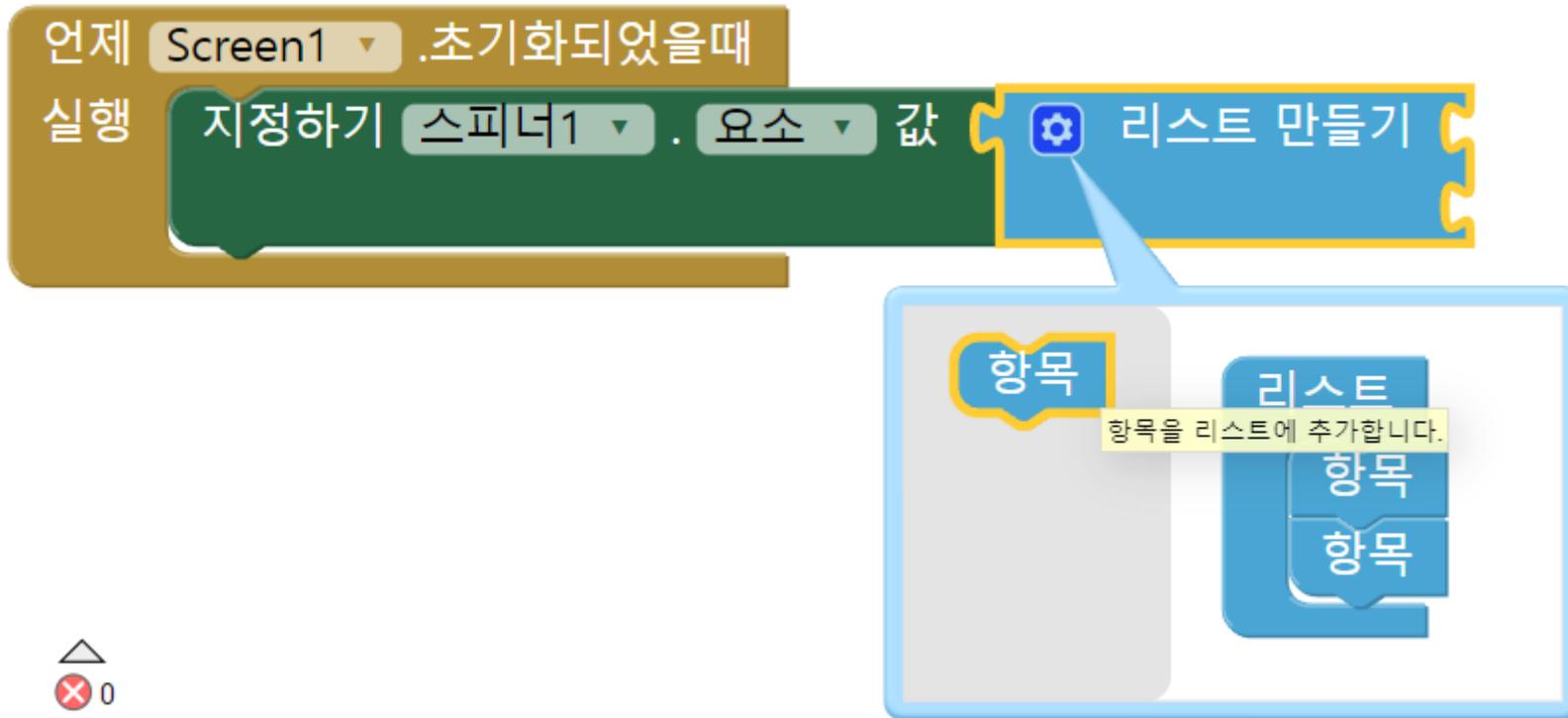


## 앱 제작의 실제6 – 번역기

언제 버튼5 ▾ .클릭했을때

실행 호출 음성인식1 ▾ .텍스트가져오기

# 앱 제작의 실제6 – 번역기



## 앱 제작의 실제6 – 번역기



## 앱 제작의 실제6 – 번역기

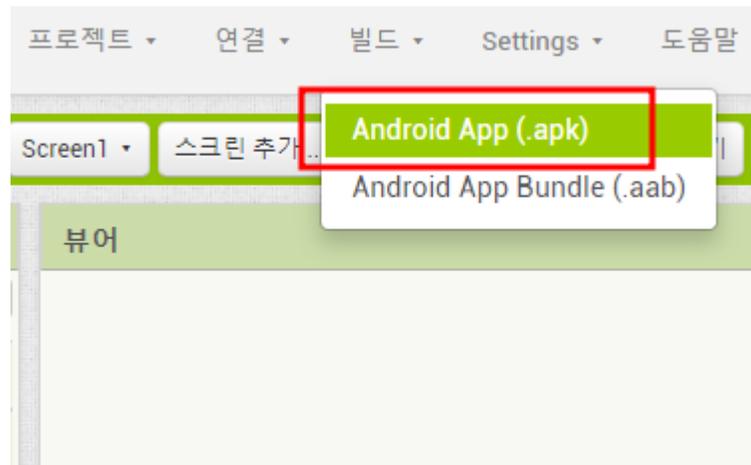
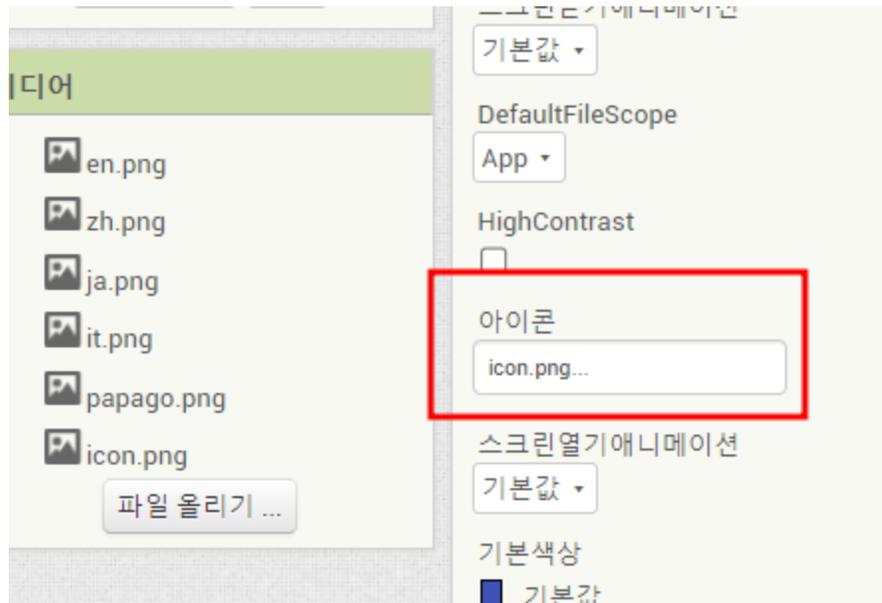
언제 스피너1 선택후에

선택된항목

실행

- 만약 가져오기 선택된항목 = 영어 이라면 실행 지정하기 전역변수 언어 값 'en'
- 만약 가져오기 선택된항목 = 일본어 이라면 실행 지정하기 전역변수 언어 값 'ja'
- 만약 가져오기 선택된항목 = 중국어 이라면 실행 지정하기 전역변수 언어 값 'zh'
- 만약 가져오기 선택된항목 = 이탈리아어 이라면 실행 지정하기 전역변수 언어 값 'it'

# 앱 제작의 실제6 – 번역기



# 앱 제작의 실제6 – 번역기

## ⚠️ 유해한 파일일 수도 있음

그래도 translate\_new2.apk 파일을 다운로드하시겠습니까?

취소

무시하고 다운로드

10:44 76%

[inventor.mit.edu](http://inventor.mit.edu) 58



다운로드 1개 완료  
(4.67MB) ai2.appinventor.mit.edu

열기



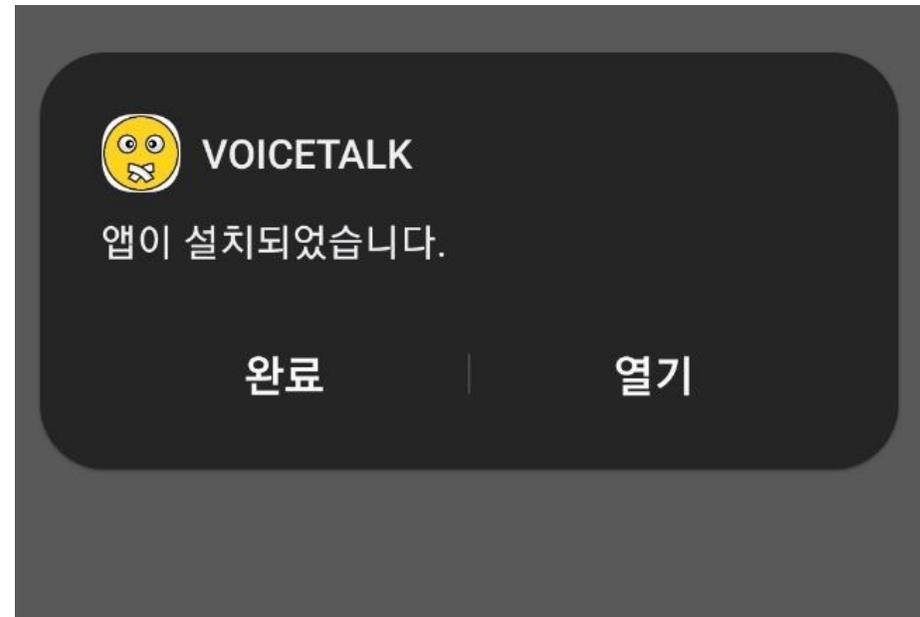
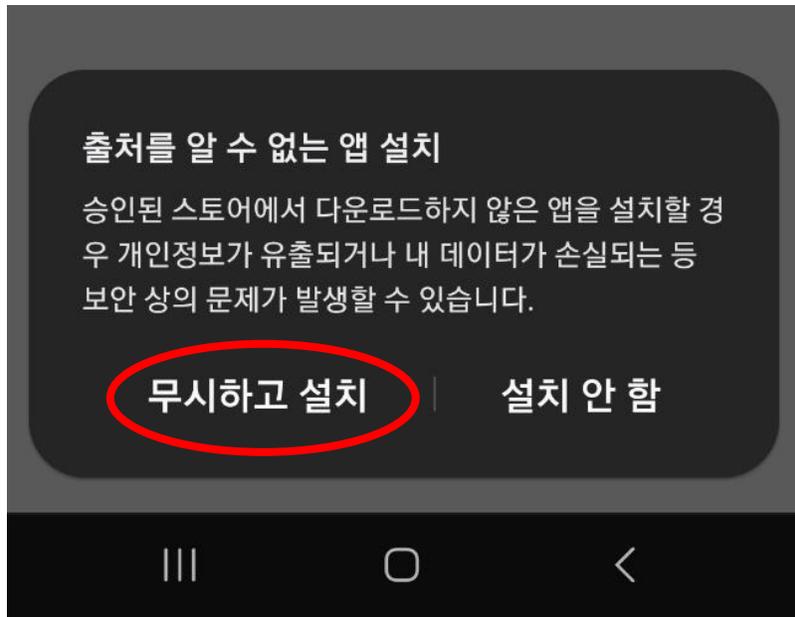
VOICETALK

이 앱을 설치하시겠습니까?

취소

설치

## 앱 제작의 실제6 – 번역기

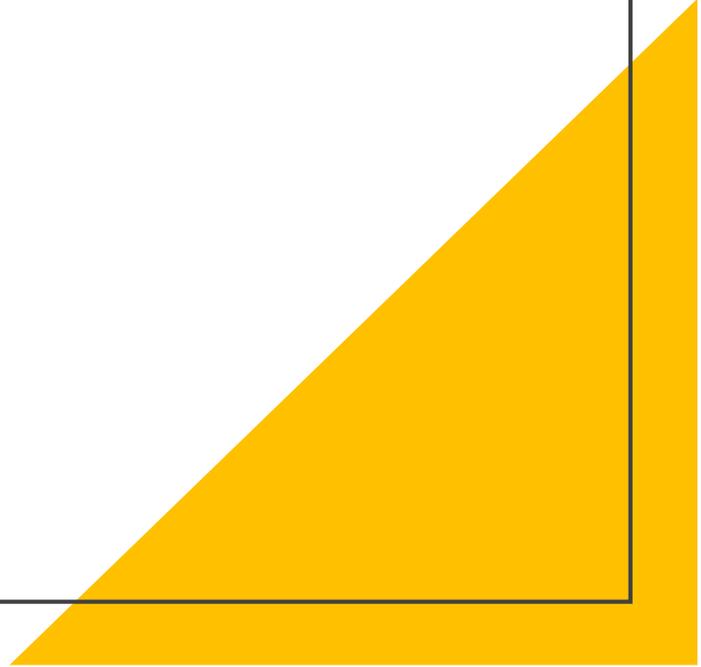


## 앱 제작의 실제6 – 번역기

앱이 잘 작동하는지 실행해보세요!

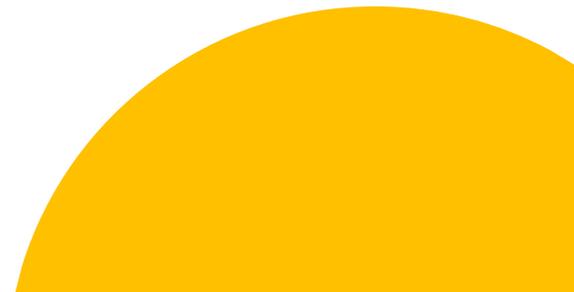


# 5. 영단어 학습 앱 만들기

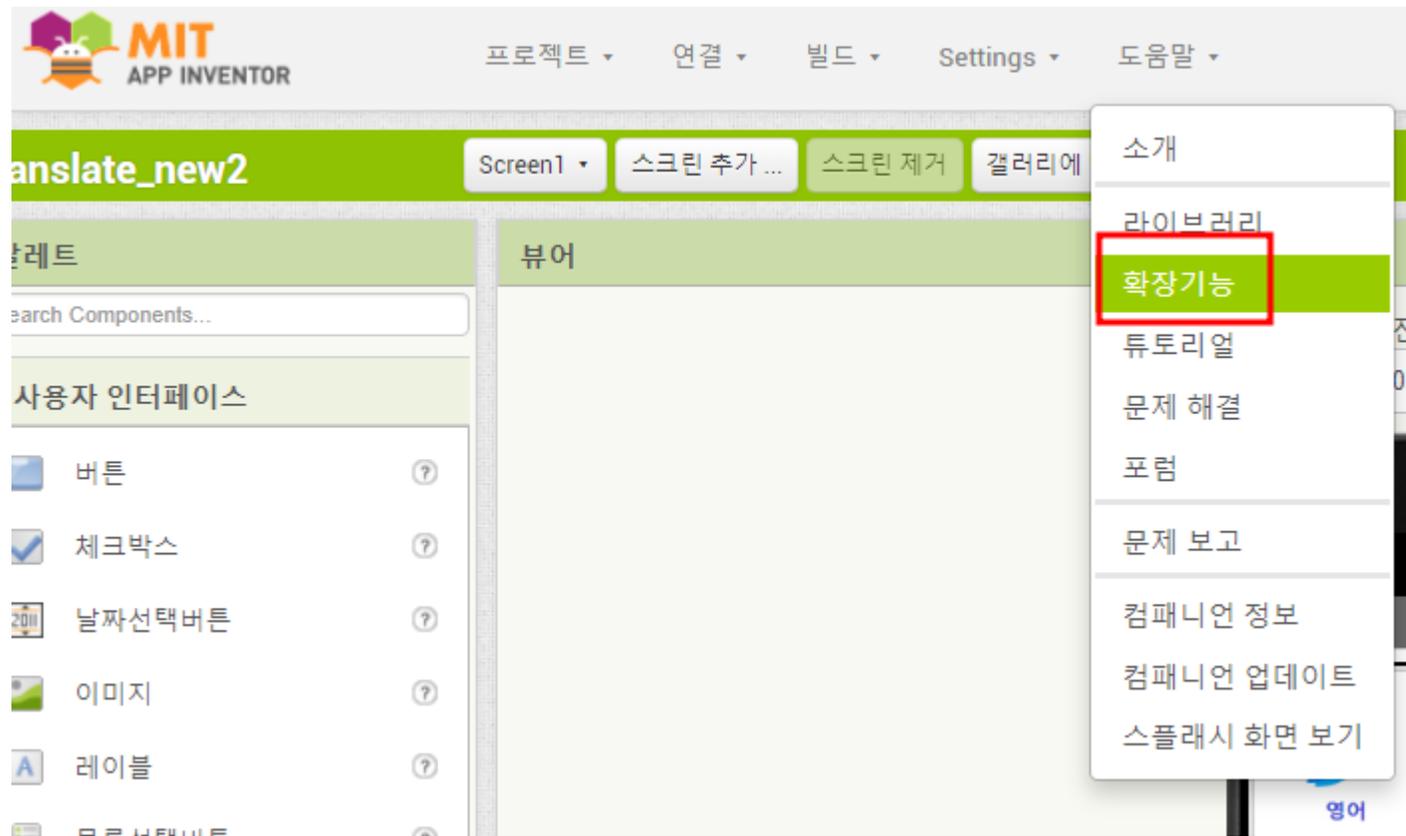


## 앱 제작의 실제7 – 영단어 학습

- 이 사물은 영어로 뭐라고 하지?



# 앱 제작의 실제7 – 영단어 학습



## 앱 제작의 실제7 – 영단어 학습

- 앱인벤터에서 지원하는 인공지능 모듈을 알아보시다.

Name	Description	Author	Version	Download .aix File	Source Code
BluetoothLE	Adds as Bluetooth Low Energy functionality to your applications. See <a href="#">BluetoothLE Documentation and Resources</a> for more information.	MIT App Inventor	20200828	<a href="#">BluetoothLE.aix</a>	<a href="#">Via GitHub</a>
LookExtension 사물 인식	Adds object recognition using a neural network compiled into the extension.	MIT App Inventor	20181124	<a href="#">LookExtension.aix</a>	<a href="#">Via GitHub</a>
PersonalAudioClassifier 오디오 분류기	Use your own neural network classifier to recognize sounds with this extension.	MIT App Inventor	20200904	<a href="#">PersonalAudioClassifier.aix</a>	<a href="#">Via GitHub</a>
PersonalImageClassifier 이미지 분류기	Use your own neural network classifier to recognize images with this extension.	MIT App Inventor	20210315	<a href="#">PersonalImageClassifier.aix</a>	<a href="#">Via GitHub</a>
PosenetExtension 자세 학습	Estimate pose with this extension.	MIT App Inventor	20200226	<a href="#">Posenet.aix</a>	<a href="#">Via GitHub</a>
FaceMeshExtension 얼굴 형태 추출	Estimate face landmarks with this extension.	MIT App Inventor	20210414	<a href="#">Facemesh.aix</a>	<a href="#">Via GitHub</a>

### 사물 인식 확장 기능



#### **Look Extension:**

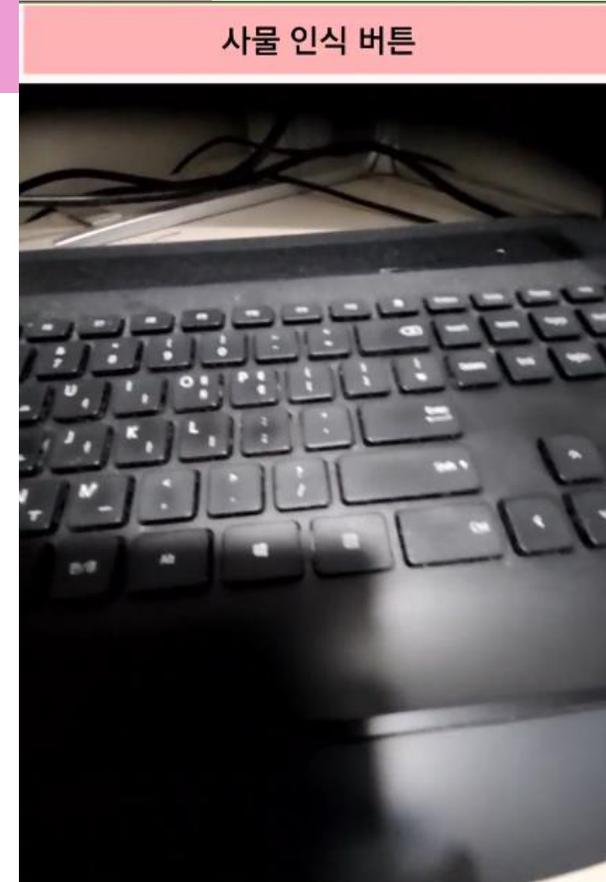
머신러닝 기반의 이미지 분류 모델

수백만 개의 이미지를 1000개의 클래스로 분류하여 학습한 결과

## 앱 제작의 실제7 – 영단어 학습

### 영어 교육 앱은 어떻게 작동할까?

- 1 사물을 카메라로 찍는다.
- 2 사물의 이름이 한글과 영어로 나타난다.
- 3 영어를 읽어준다.



keyboard 키보드

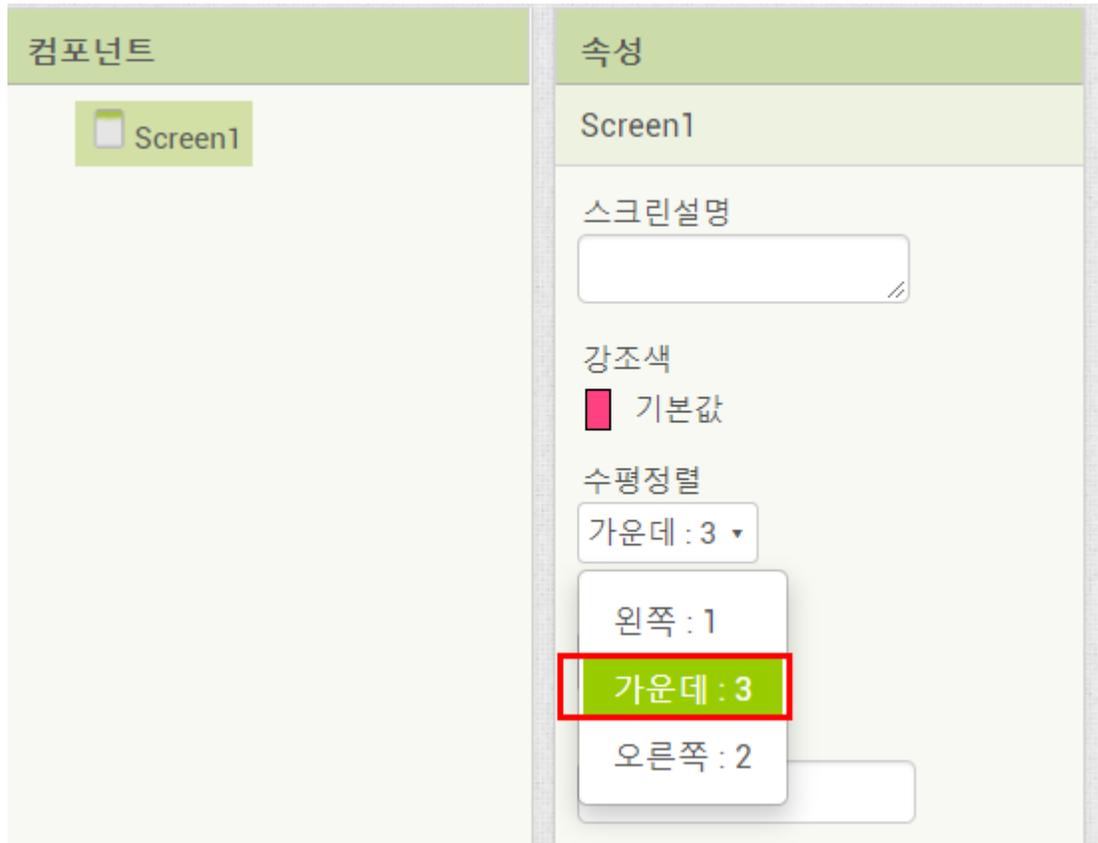
발음 듣기

# 앱 제작의 실제7 – 영단어 학습



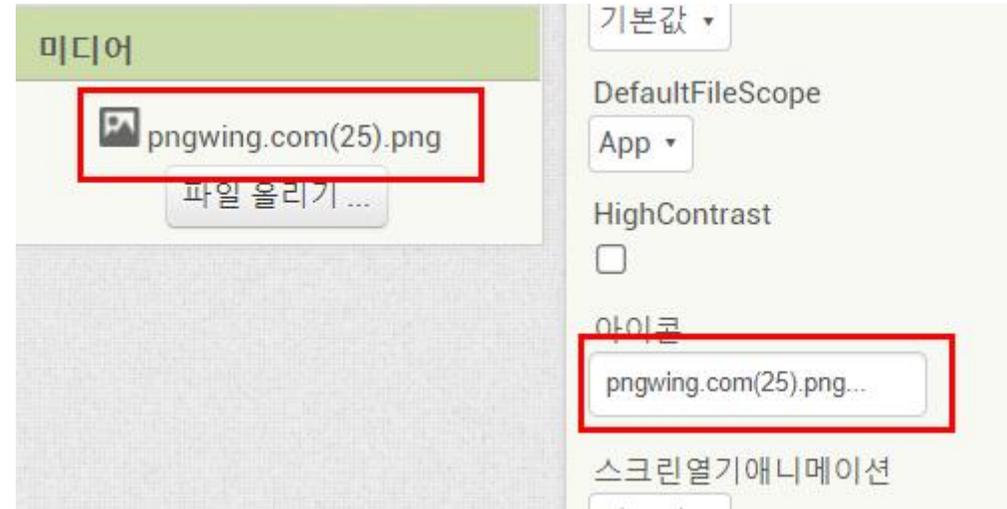
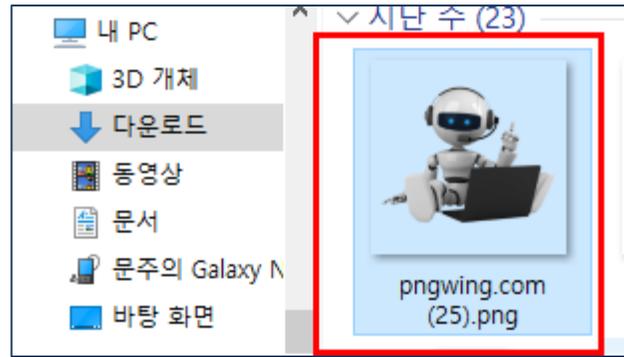
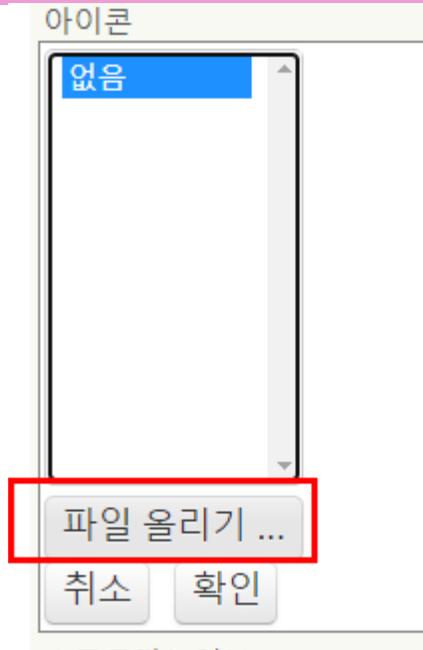
words

## 앱 제작의 실제7 – 영단어 학습



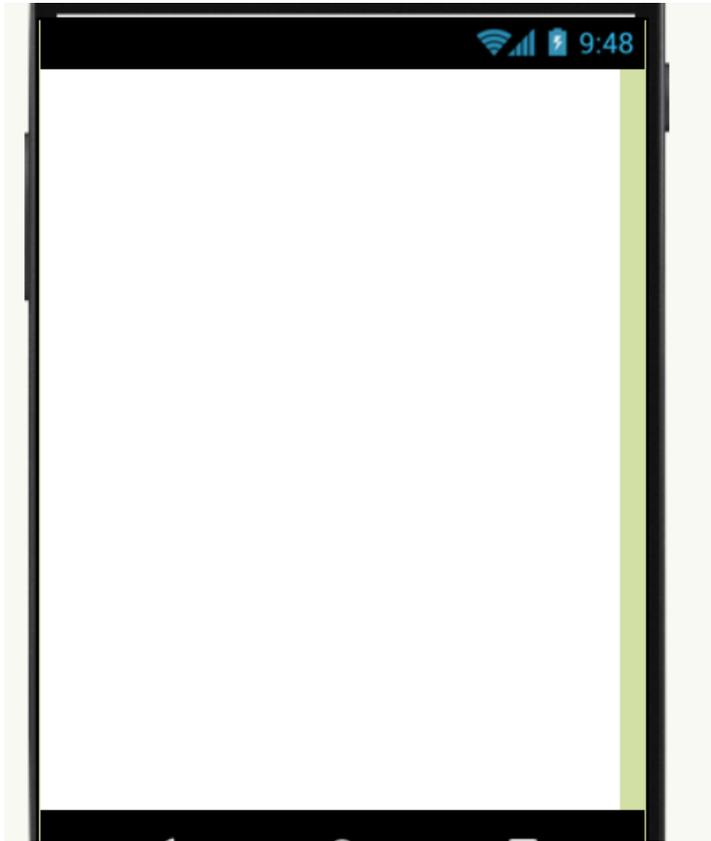
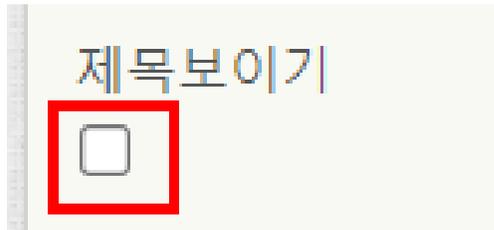
Screen1 수평 정렬 '가운데'

# 앱 제작의 실제7 – 영단어 학습



아이콘에 그림 올림  
앱을 설치하면 이 그림으로  
아이콘이 생김

## 앱 제작의 실제7 – 영단어 학습



제목 보이기 체크 해제  
뷰어에서 제목이 사라짐

# 앱 제작의 실제7 – 영단어 학습



배경색  
 분홍

활성화

글꼴굵게

글꼴이텔릭

글꼴크기  
20

글꼴서체  
기본값 ▾

높이  
자동...

너비  
부모 요소에 맞추기...

이미지  
없음...

모양  
기본값 ▾

피드백보이기

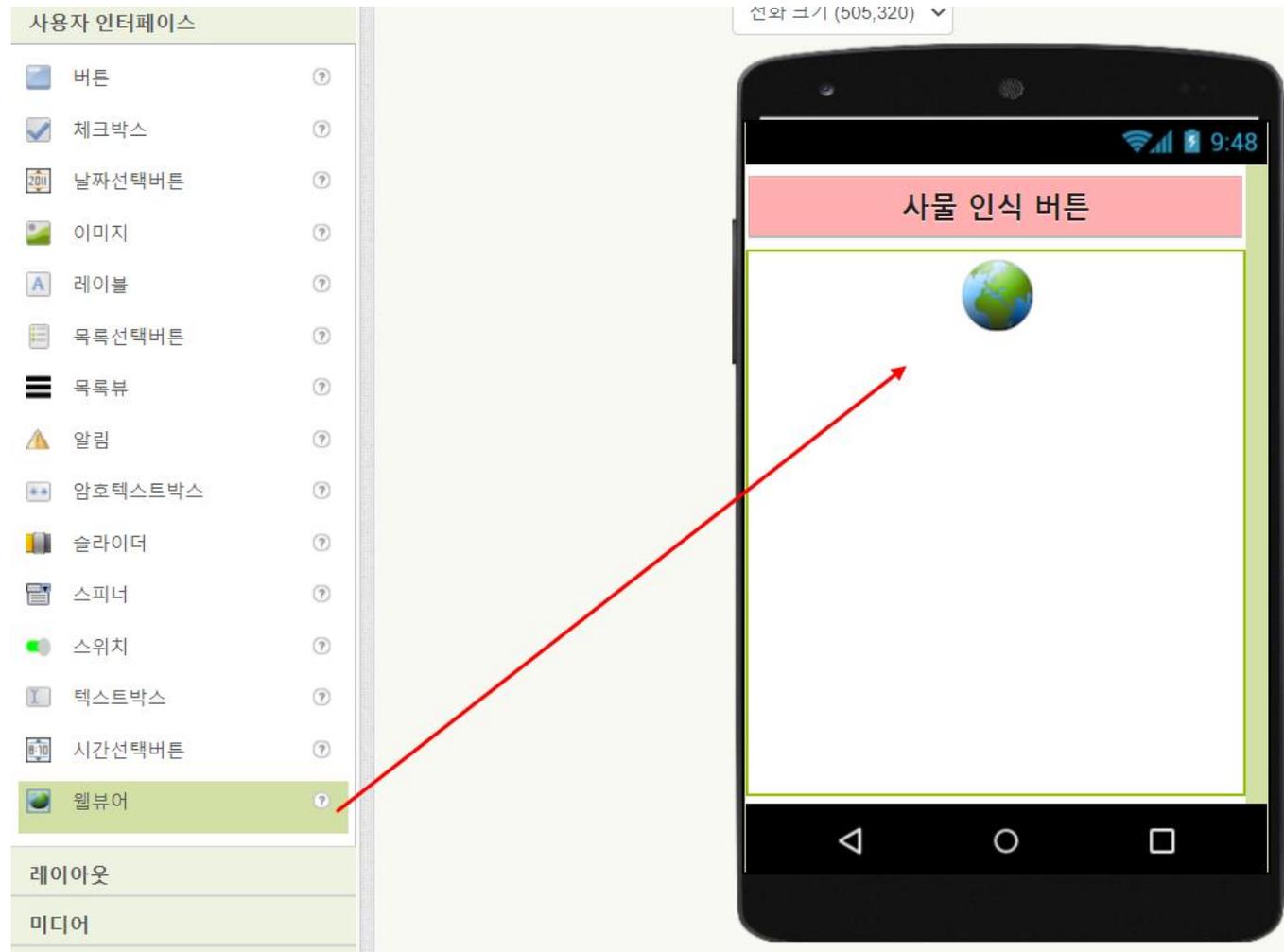
텍스트  
사물 인식 버튼

텍스트정렬  
가운데 : 1 ▾

텍스트색상  
 기본값

반영기여도

# 앱 제작의 실제7 – 영단어 학습



# 앱 제작의 실제7 – 영단어 학습

레이아웃

- 수평배치
- 스크롤가능수평배치
- 표형식배치
- 수직배치
- 스크롤가능수직배치

미디어

그리기 & 애니메이션

지도

센서

소셜

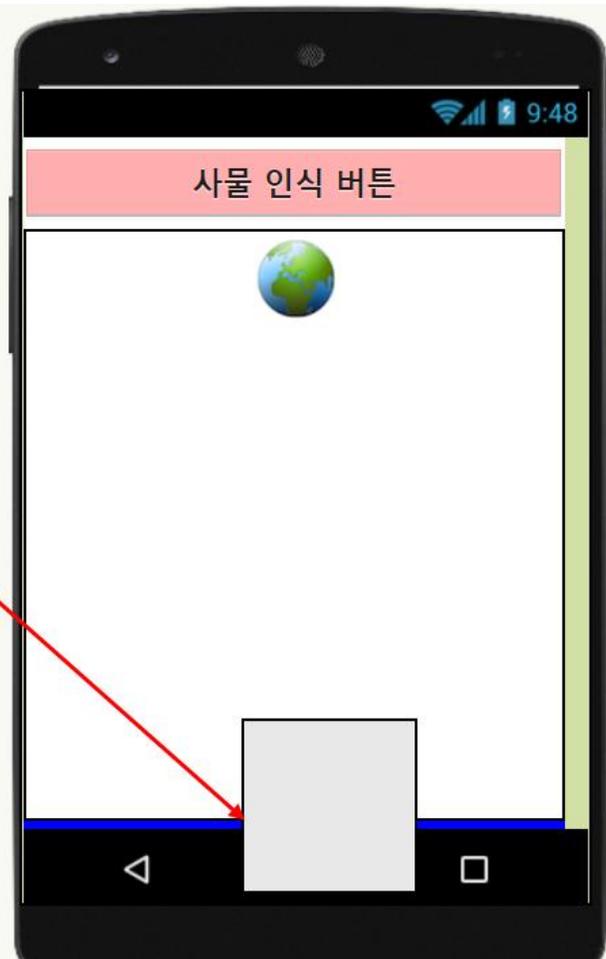
저장소

연결

레고® 마인드스톰®

실험실

확장기능



속성

수평배치1

수평정렬  
가운데 : 3

수직정렬  
위 : 1

배경색  
■ 기본값

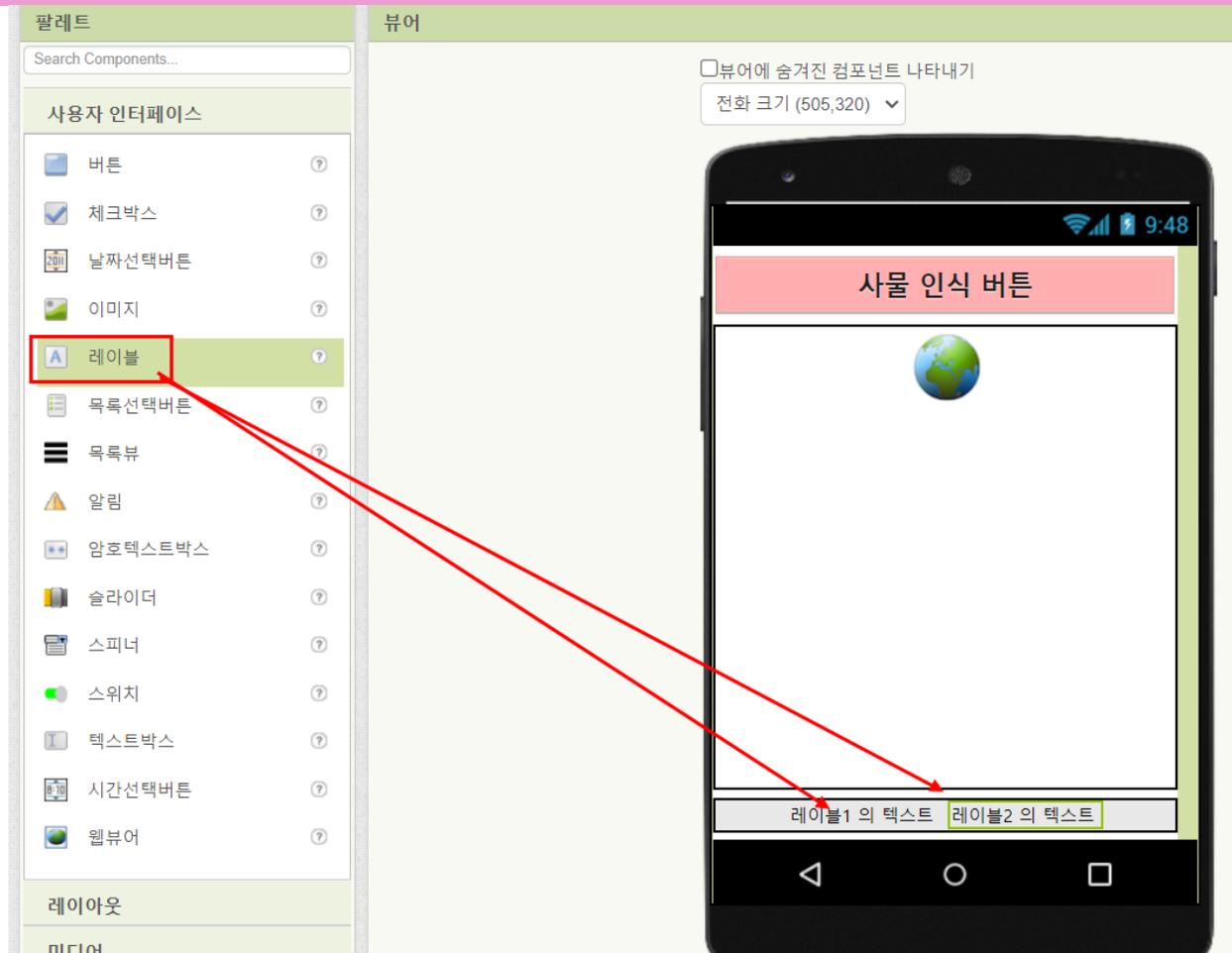
높이  
자동...

너비

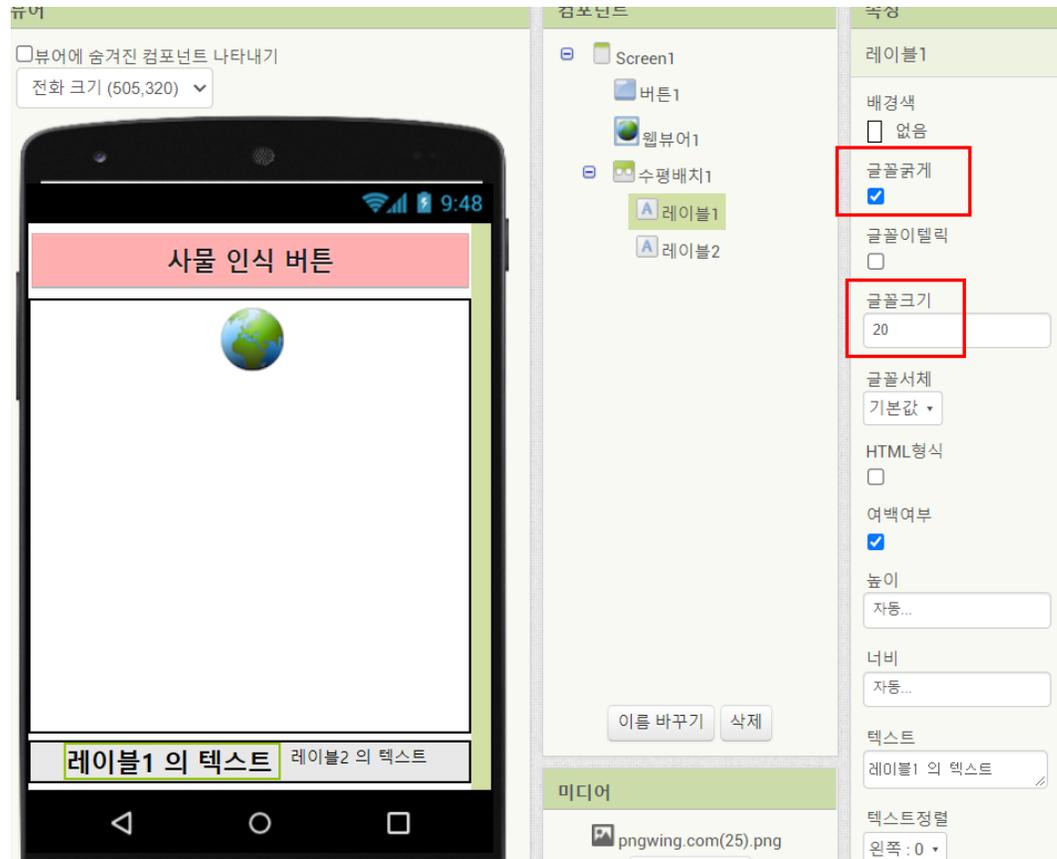
- 자동
- 부모 요소에 맞추기
- 픽셀
- 퍼센트

취소    확인

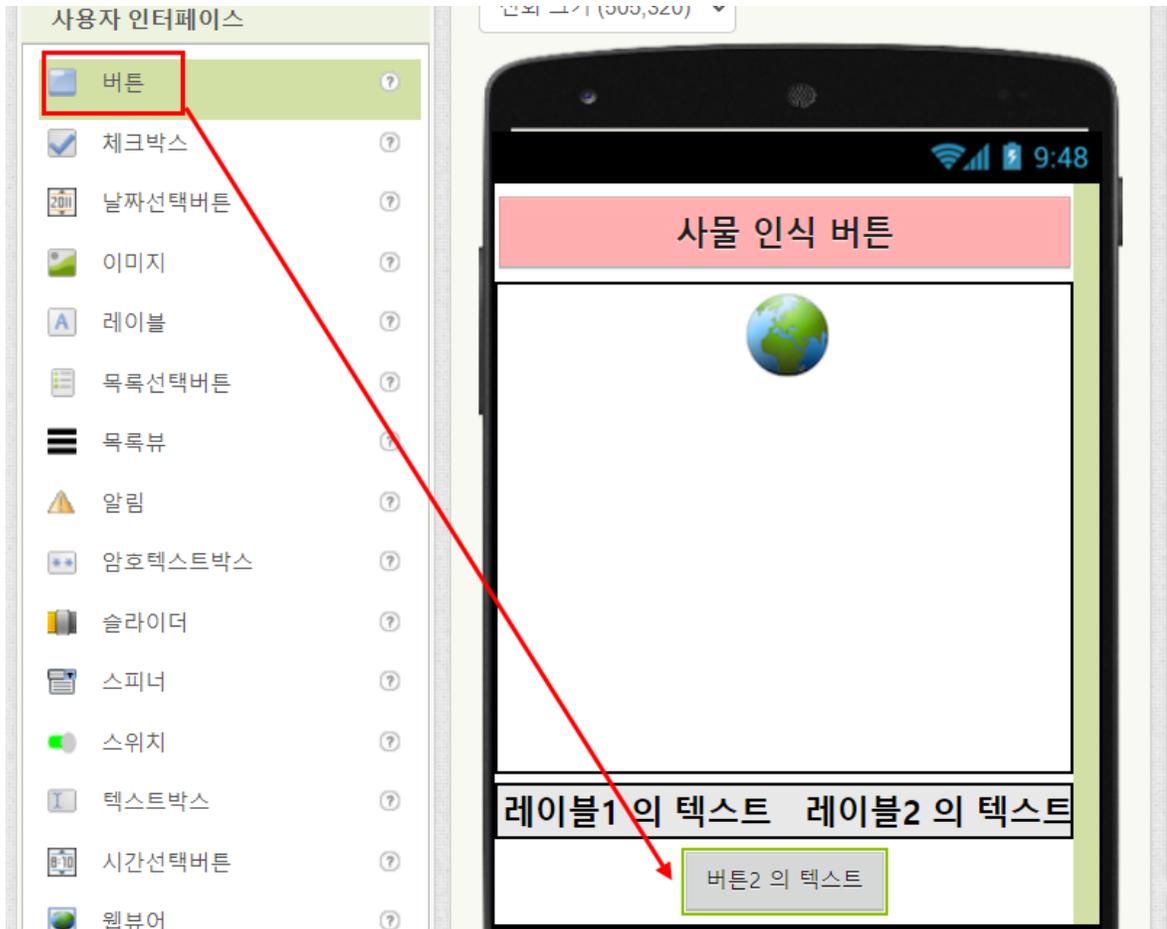
# 앱 제작의 실제7 – 영단어 학습



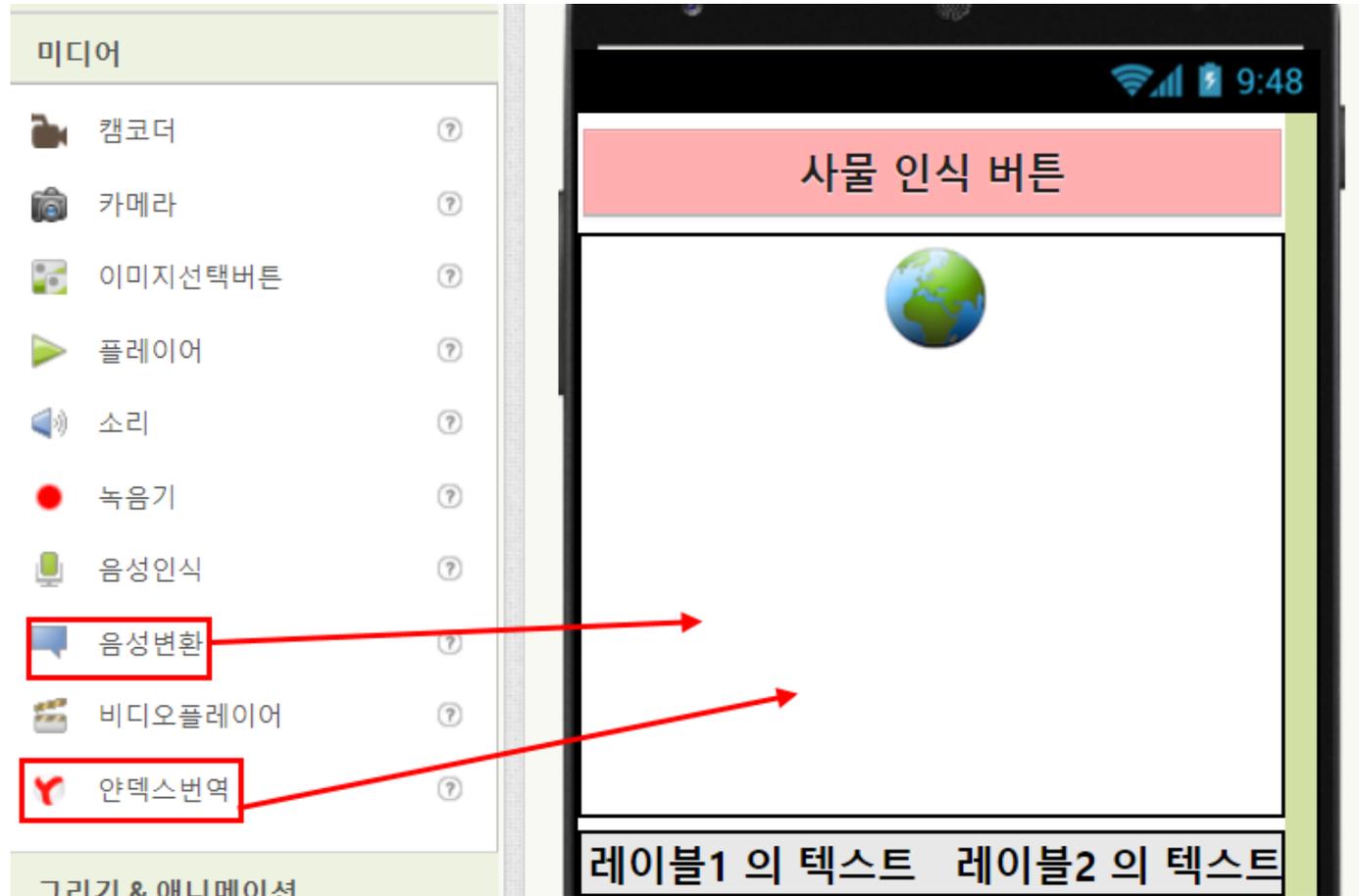
# 앱 제작의 실제7 – 영단어 학습



# 앱 제작의 실제7 – 영단어 학습



# 앱 제작의 실제7 – 영단어 학습



## 앱 제작의 실제7 – 영단어 학습

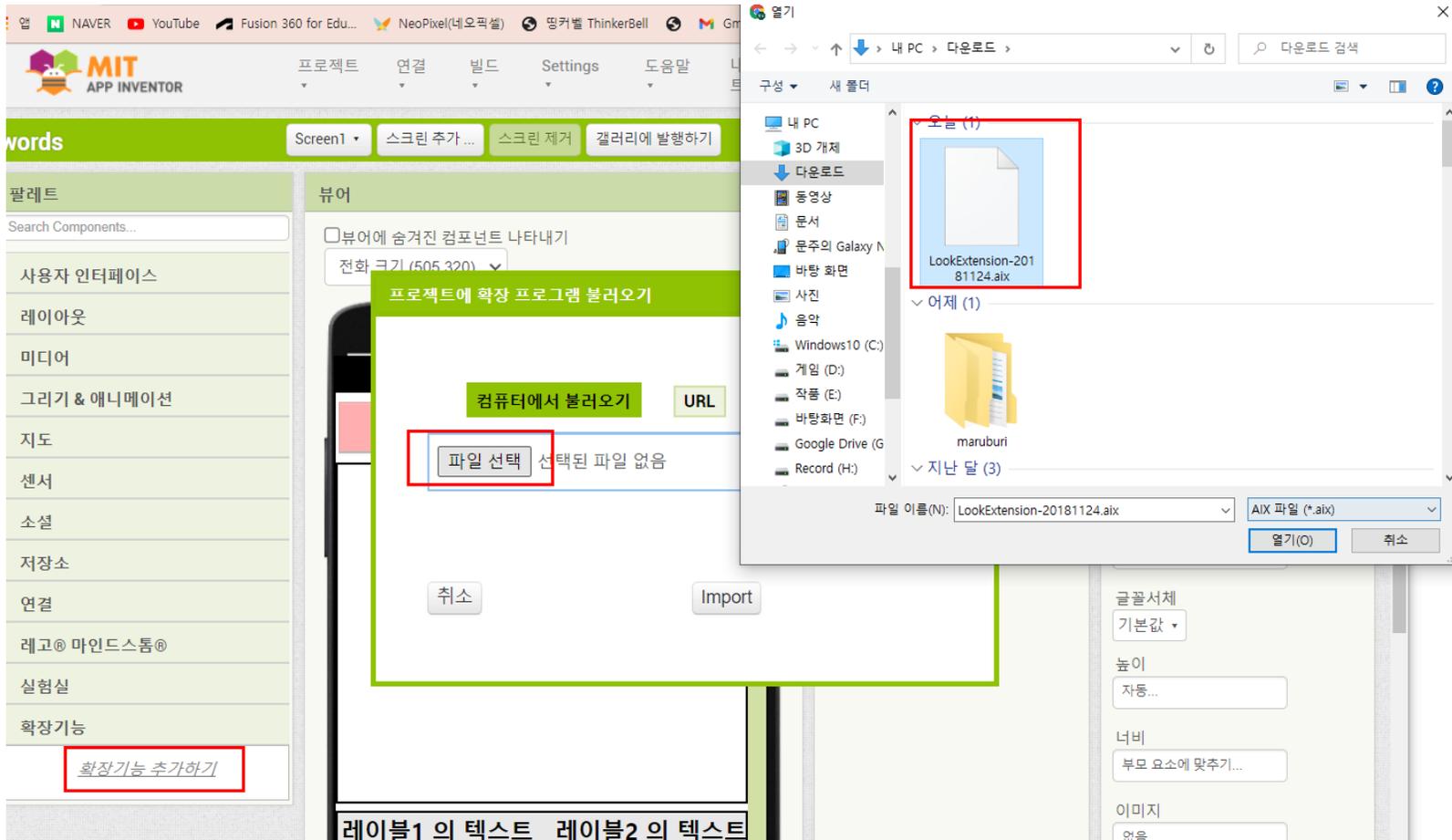
### 앱 인벤터 확장 기능:

컴포넌트에서 제공하지 않는 기능을 확장시킴  
앱 인벤터에서 공식적으로 제공하기도 하고  
개발자들이 만들어 공유하기도 함

#### Supported:

Name	Description	Author	Version	Download .aix File	Source Code
BluetoothLE	Adds as Bluetooth Low Energy functionality to your applications. See <a href="#">BluetoothLE Documentation and Resources</a> for more information.	MIT App Inventor	20200828	<a href="#">BluetoothLE.aix</a>	<a href="#">Via GitHub</a>
LookExtension	Adds object recognition using a neural network compiled into the extension.	MIT App Inventor	20181124	<a href="#">LookExtension.aix</a>	<a href="#">Via GitHub</a>
PersonalAudioClassifier	Use your own neural network classifier to recognize sounds with this extension.	MIT App Inventor	20200904	<a href="#">PersonalAudioClassifier.aix</a>	<a href="#">Via GitHub</a>
PersonallImageClassifier	Use your own neural network classifier to recognize images with this extension.	MIT App Inventor	20210315	<a href="#">PersonallImageClassifier.aix</a>	<a href="#">Via GitHub</a>
PosenetExtension	Estimate pose with this extension.	MIT App Inventor	20200226	<a href="#">Posenet.aix</a>	<a href="#">Via GitHub</a>
FaceMeshExtension	Estimate face landmarks with this extension.	MIT App Inventor	20210414	<a href="#">Facemesh.aix</a>	<a href="#">Via GitHub</a>

# 앱 제작의 실제7 – 영단어 학습



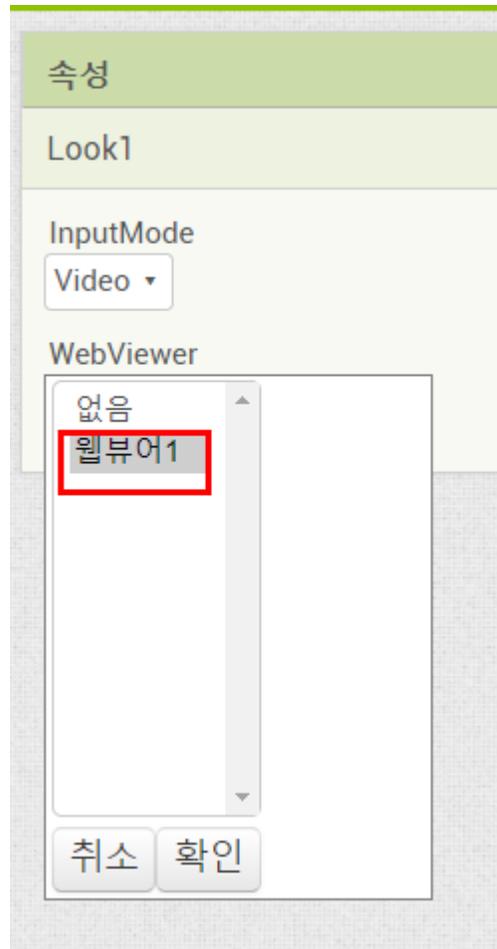
다운 받은 aix 파일을  
'확장 기능 추가하기'로  
가져오기

# 앱 제작의 실제7 – 영단어 학습



확장기능 추가

## 앱 제작의 실제7 – 영단어 학습



웹뷰어1 추가

## 앱 제작의 실제7 – 영단어 학습

The image shows a screenshot of an application development environment. On the left, a tree view displays the project structure. Under 'Screen1', '버튼1' (Button1) is highlighted with a red box. Below it, 'Look1' is also highlighted with a red box. On the right, a visual programming canvas shows a flowchart. A yellow box contains the event '언제 버튼1 .클릭했을때' (When Button1 is clicked) and the action '실행 호출 Look1 .ClassifyVideoData' (Execute Call Look1 .ClassifyVideoData). Red arrows point from the '버튼1' and 'Look1' boxes in the tree view to their respective elements in the flowchart. At the bottom, there are warning and error indicators (0 each) and a 'Show Warnings' button.

Screen1

- 버튼1
- 웹뷰어1

수평배치1

- 레이블1
- 레이블2

버튼2

얀덱스번역1

음성변환1

Look1

언제 버튼1 .클릭했을때

실행 호출 Look1 .ClassifyVideoData

0 0

Show Warnings

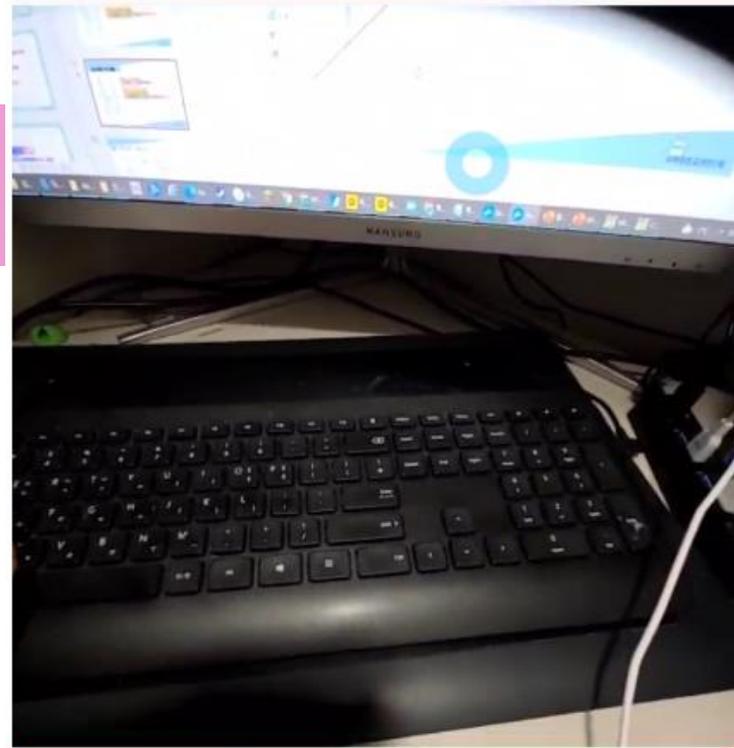
# 앱 제작의 실제7 – 영단어 학습

The image displays a visual programming environment for an application. On the left is a component palette with the following items: 색상 (Color), 변수 (Variable), 함수 (Function), Screen1, 버튼1 (Button1), 웹뷰어 (Webviewer), 수평배치1 (Horizontal container), 레이블1 (Label1), 레이블2 (Label2), 버튼2 (Button2), 안텍스번역1 (Text translation), 음성변환1 (Voice conversion), and Look1. The main workspace contains two event-driven logic blocks. The first block is triggered by '언제 버튼1 .클릭했을때' (When Button1 is clicked) and contains an '실행 호출 Look1 .ClassifyVideoData' (Execute Call Look1 .ClassifyVideoData) block. The second block is triggered by '언제 Look1 .GotClassification' (When Look1 .GotClassification) and contains a '결과' (Result) block, followed by an '실행 지정하기 레이블1 . 텍스트 값' (Execute Assign Label1 . Text Value) block, and finally a '가져오기 결과' (Fetch Result) block. Red arrows indicate data flow: one arrow points from '레이블1' in the component palette to the '지정하기' block, another points from the '결과' block to the '가져오기' block, and a third points from the '호출' block to the '결과' block. At the bottom, there are two status indicators: a yellow warning triangle with '0' and a red 'X' with '0'.

## 앱 제작의 실제7 – 영단어 학습

스마트폰에서  
사진인식버튼을 누르면  
레이블1에 각 사물과 비율  
10개가 리스트의 형식으로  
나오는 것을 발견할 수 있다.

사물 인식 버튼



```
[["keyboard", "0.89600"], ["mouse",  
'0.04224'], ["crossword puzzle",  
'0.01831'], ["remote control",  
'0.01239'], ["display", "0.00618"],  
["laptop", "0.00255"], ["sunglasses",  
'0.00237'], ["modem", "0.00223"],  
["printer", "0.00189"], ["pen", "0.00149"]]
```

발음 듣기

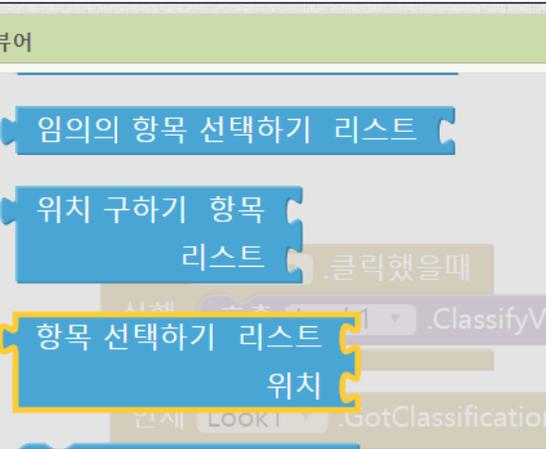
## 앱 제작의 실제7 – 영단어 학습

### 유의점

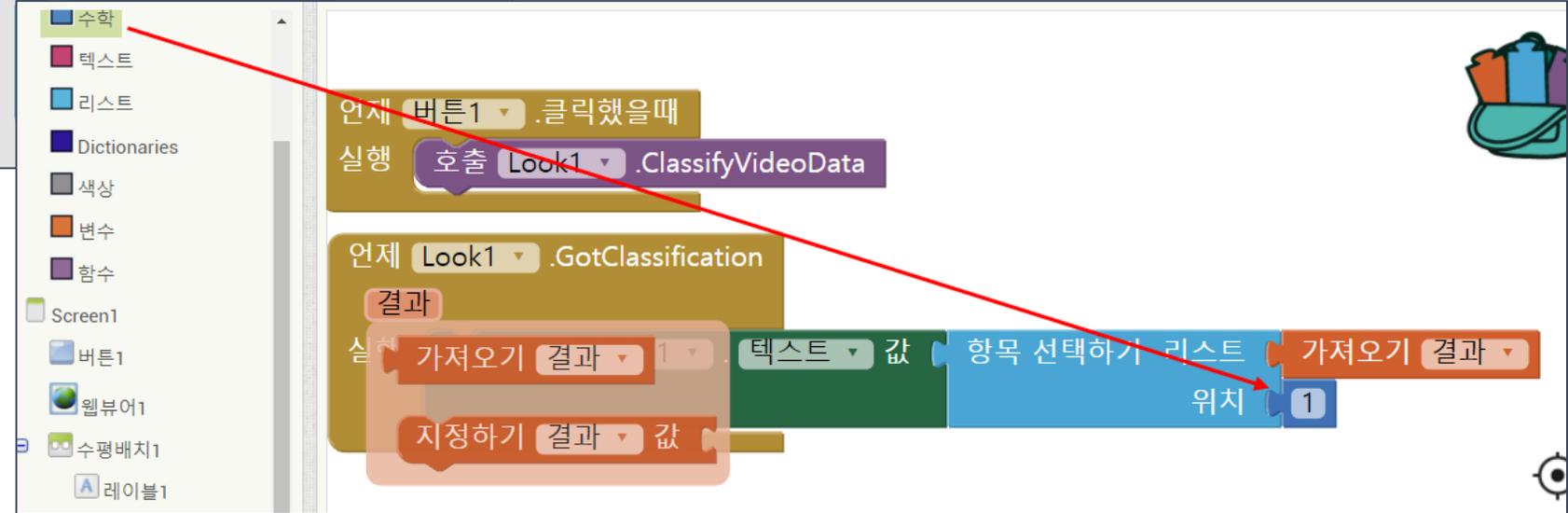
- 미국의 사물들을 기준으로 한 것이라, 사물 인식이 정확하게 안 될 수도 있음



# 앱 제작의 실제7 – 영단어 학습

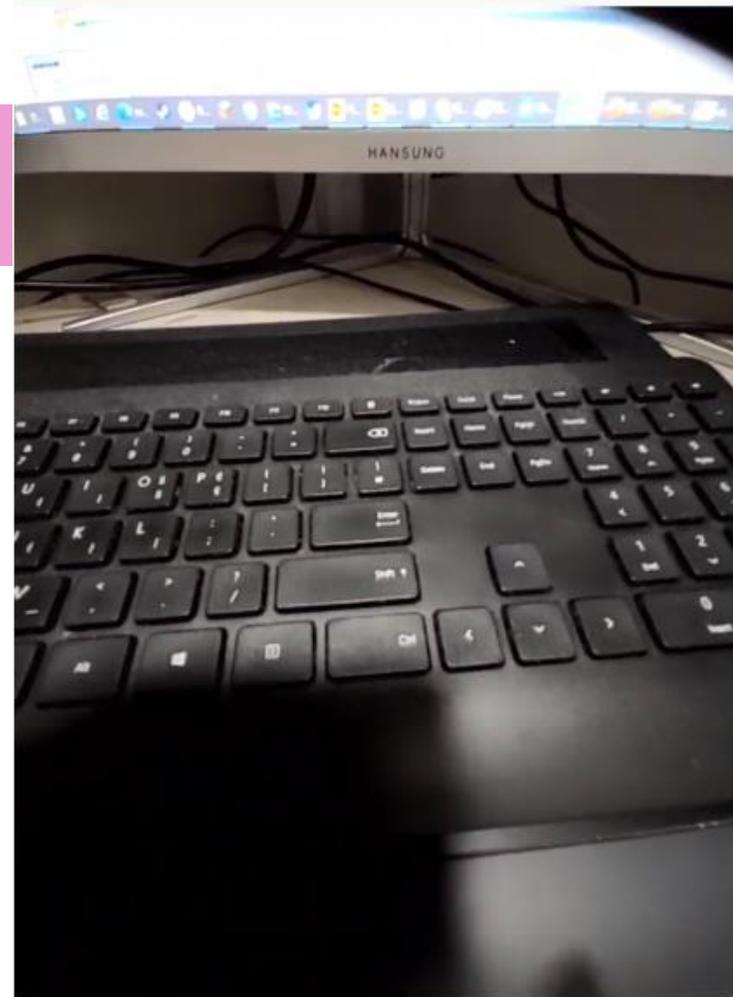


정확도가 가장 높은 첫 번째  
리스트 항목만 가지고 온다.



## 앱 제작의 실제7 - 영단어 학습

첫번째 리스트만 나타난다.



["keyboard", "0.70313"] 레이블2 의 텍스트

발음 듣기

## 앱 제작의 실제7 – 영단어 학습

어떤 사물인지만 필요하므로  
첫 번째 리스트 안에서도 첫 번째  
항목만 가져온다

언제 버튼1 ▾ .클릭했을때

실행 호출 Look1 ▾ .ClassifyVideoData

언제 Look1 ▾ .GotClassification

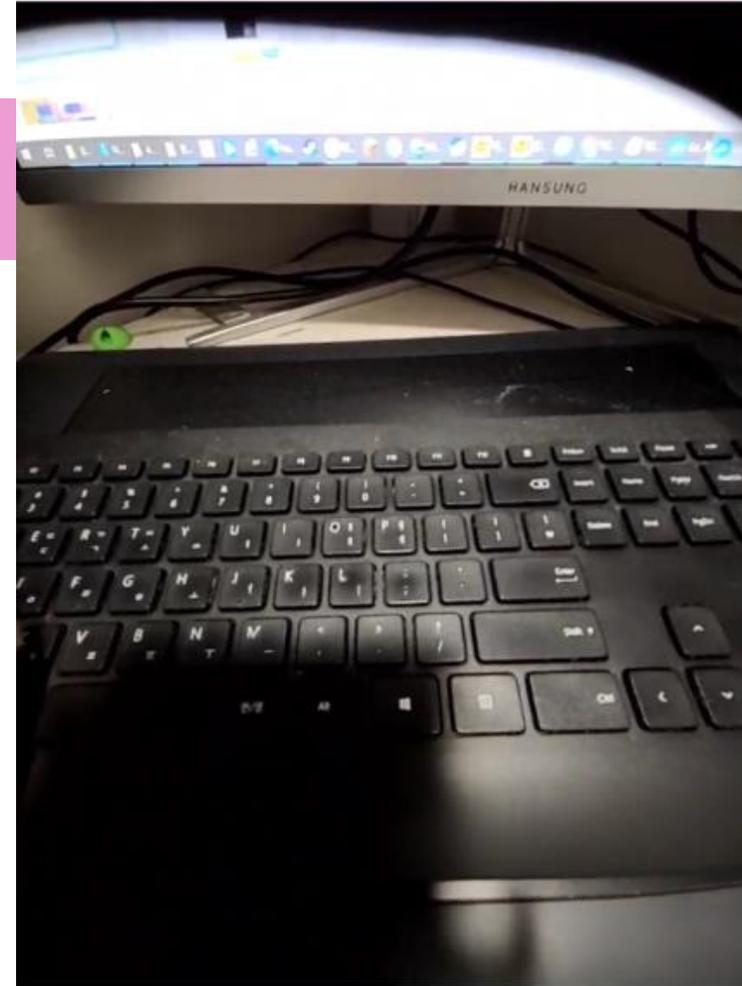
결과

실행 지정하기 레이블1 ▾ . 텍스트 ▾ 값 ▶ 항목 선택하기 리스트 ▶ 항목 선택하기 리스트 ▶ 가져오기 결과 ▾  
▶ 위치 1 ▶ 위치 1 ▶ 위치 1

사물 인식 버튼

## 앱 제작의 실제7 - 영단어 학습

첫번째 리스트의 첫 번째 항목인  
사물의 이름만 나타난다.



keyboard 레이블2 의 텍스트

발음 듣기

# 앱 제작의 실제7 – 영단어 학습

레이블2에 한글로 표시하기 위해  
얀덱스 번역 블록을 가져온다.

The image shows a Scratch-like block editor interface. On the left is a 'Sprites' panel with categories: 수화, 텍스트, 리스트, Dictionaries, 색상, 변수, 함수, Screen1, 버튼1, 웹뷰어, 수평배치1, 레이블1, 레이블2, 버튼2, 안덱스번역1 (highlighted with a red box), 음성변환1, and Look1. The main workspace contains two event-driven code blocks:

- 언제 버튼1 .클릭했을때**  
실행 호출 Look1 .ClassifyVideoData
- 언제 Look1 .GotClassification**  
결과  
실행 지정하기 레이블1 . 텍스트 값 항목 선택하기 리스트 항목 선택하기 리스트 가져오기 결과 위치 1 위치 1

A red box highlights a custom block named '호출 안덱스번역1 .번역요청하기'. The block has three input fields: '번역언어코드' and '번역할텍스트'.

# 앱 제작의 실제7 – 영단어 학습

en-ko: 영어를 한국어로 번역  
([https://ko.m.wikipedia.org/wiki/ISO\\_639-1\\_%EC%BD%94%EB%93%9C\\_%EB%AA%A9%EB%A1%9D](https://ko.m.wikipedia.org/wiki/ISO_639-1_%EC%BD%94%EB%93%9C_%EB%AA%A9%EB%A1%9D))  
레이블1 텍스트 값을 번역한다.

수학

텍스트

리스트

Dictionaries

색상

변수

함수

Screen1

버튼1

웹뷰어1

수평배치1

레이블1

레이블2

버튼2

안텍스번역1

언제 버튼1 .클릭했을 때

실행 호출 Look1 .ClassifyVideo

언제 Look1 .GotClassification

결과

실행 지정하기 레이블1 . 텍스트 값

호출 안텍스번역1 .번역요청하기

번역언어코드 'en-ko'

번역할 텍스트 레이블1 . 텍스트

항목 선택하기 리스트

항목 선택하기 리스트

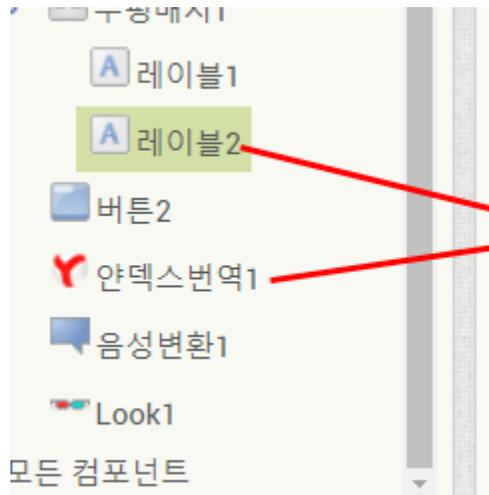
가져오기 결과

위치

위치

# 앱 제작의 실제7 - 영단어 학습

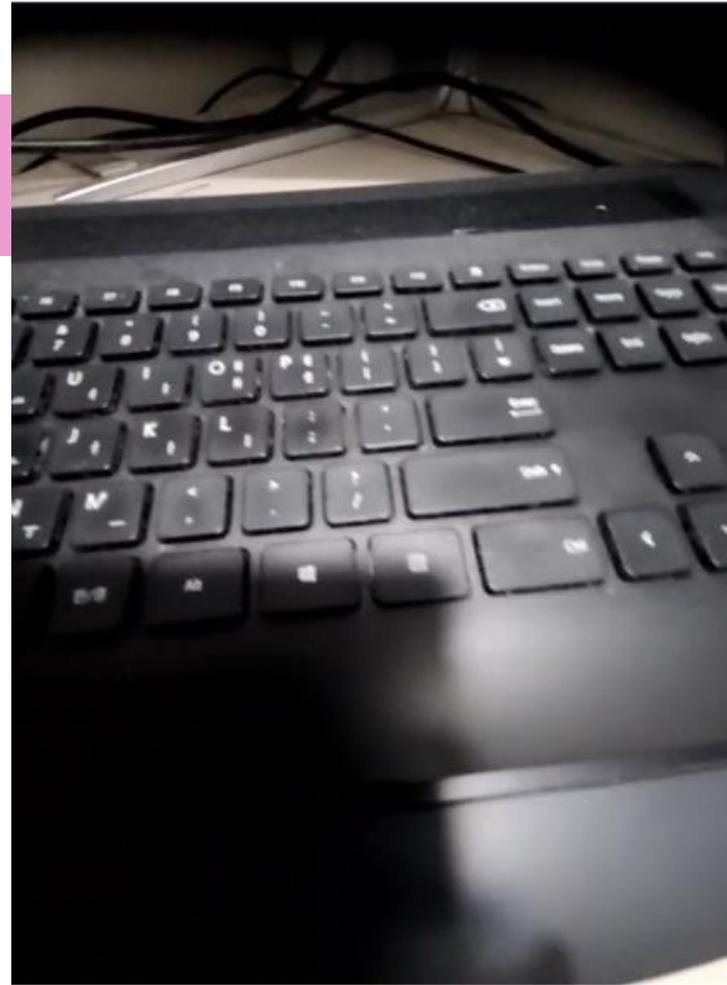
번역을 받으면 레이블2에  
번역 결과가 나오도록 한다.



## 앱 제작의 실제7 – 영단어 학습

영어 옆에 한글이 나온다.

사물 인식 버튼

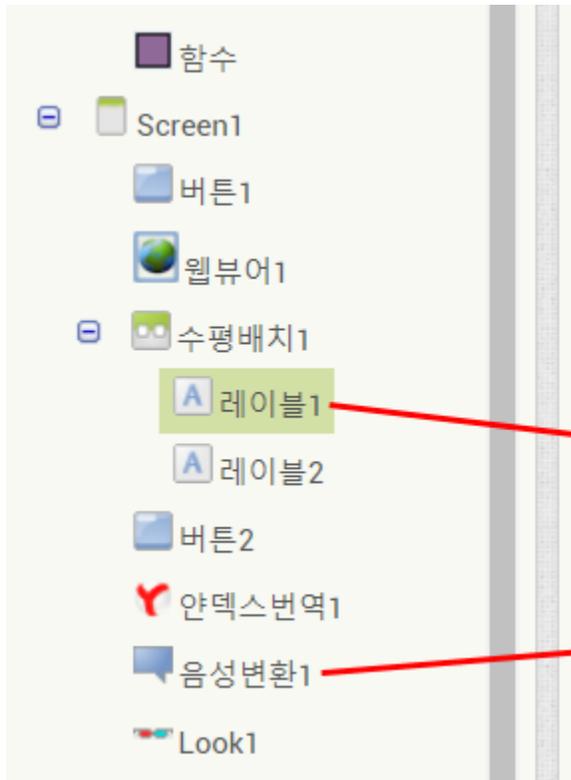


keyboard 키보드

발음 듣기

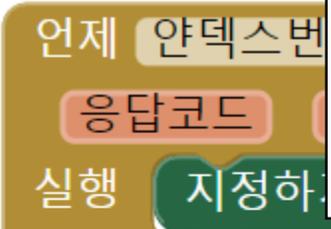
# 앱 제작의 실제7 – 영단어 학습

버튼 2를 누르면  
레이블1의 영어를 발음하도록 한다.



함수

- Screen1
  - 버튼1
  - 웹뷰어1
- 수평배치1
  - 레이블1
  - 레이블2
- 버튼2
- 안텍스번역1
- 음성변환1
- Look1



언제 안텍스번  
응답코드  
실행 지정하

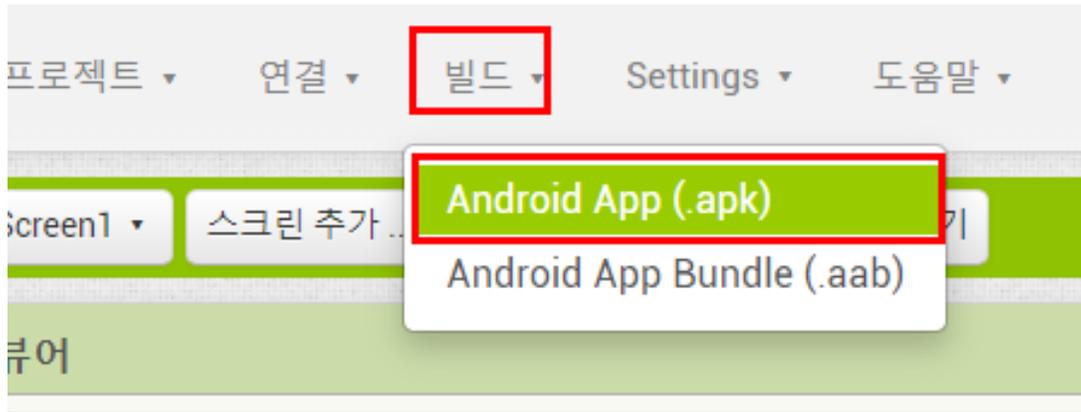


언제 버튼2 .클릭했을때  
실행 호출 음성변환1 .말하기  
메시지 레이블1 . 텍스트



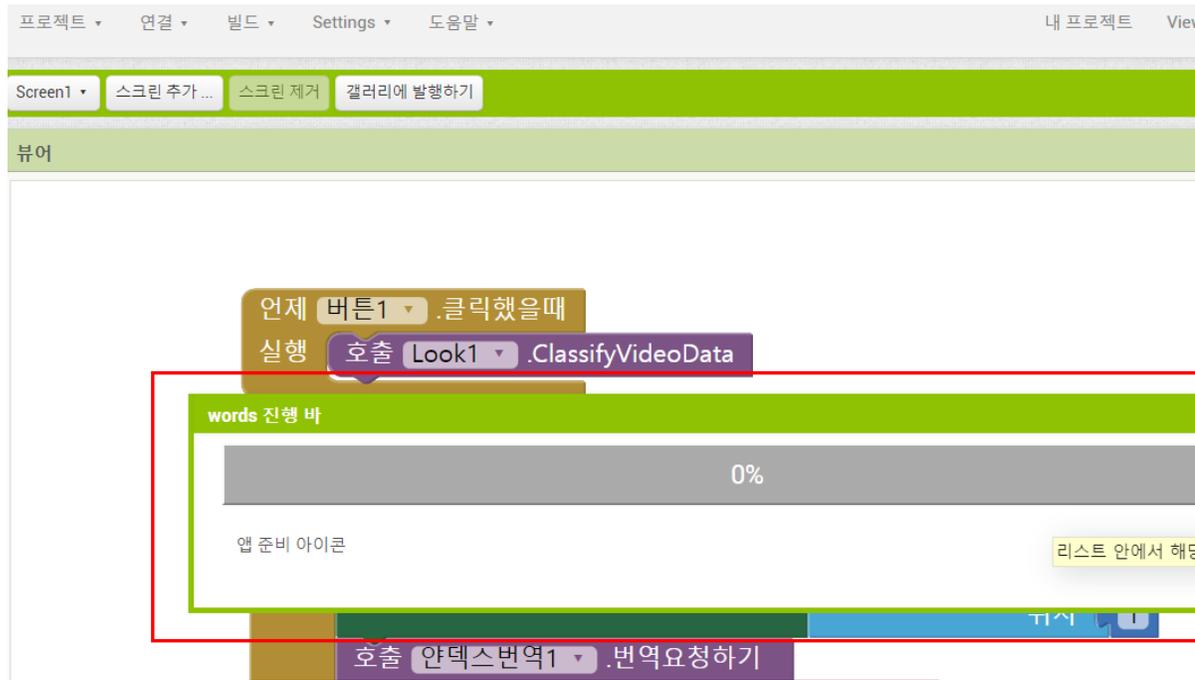
Warning: 0  
Error: 0

## 앱 제작의 실제7 – 영단어 학습



AI 컴패니언을 종료한 후  
빌드 – Android App(.apk)를 선택한다

# 앱 제작의 실제7 – 영단어 학습



바코드 생성까지 시간이 걸린다

# 앱 제작의 실제7 – 영단어 학습

Android App for words

  
Download .apk now



Click the button to download the app, right-click on it to copy a download link, or scan the code with a barcode scanner to install.  
Note: this link and barcode are only valid for 2 hours. See [the FAQ](#) for info on how to share your app with others.

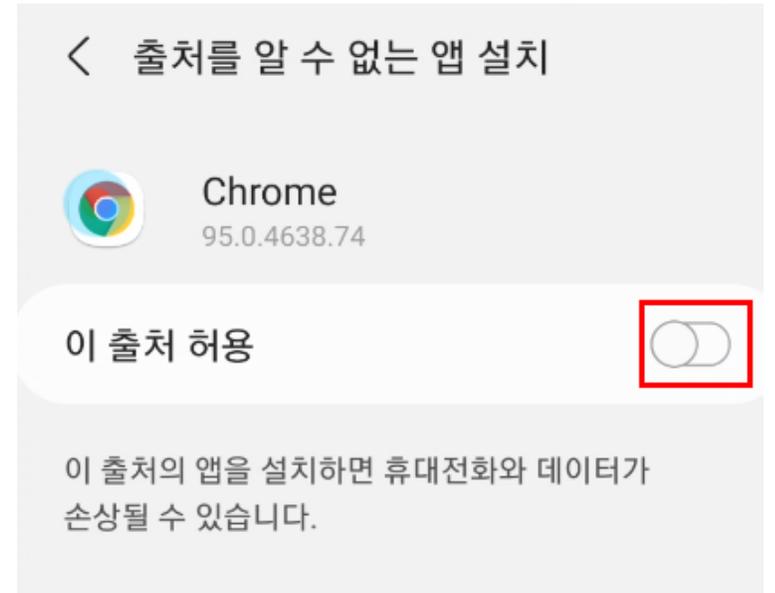
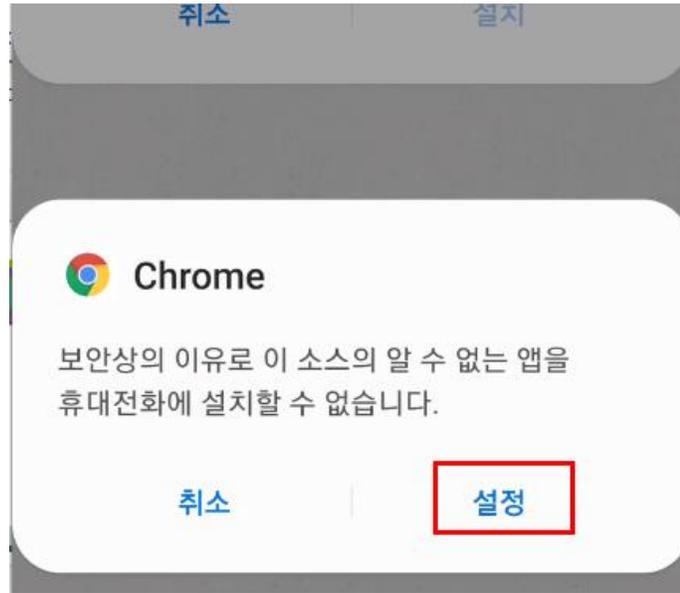
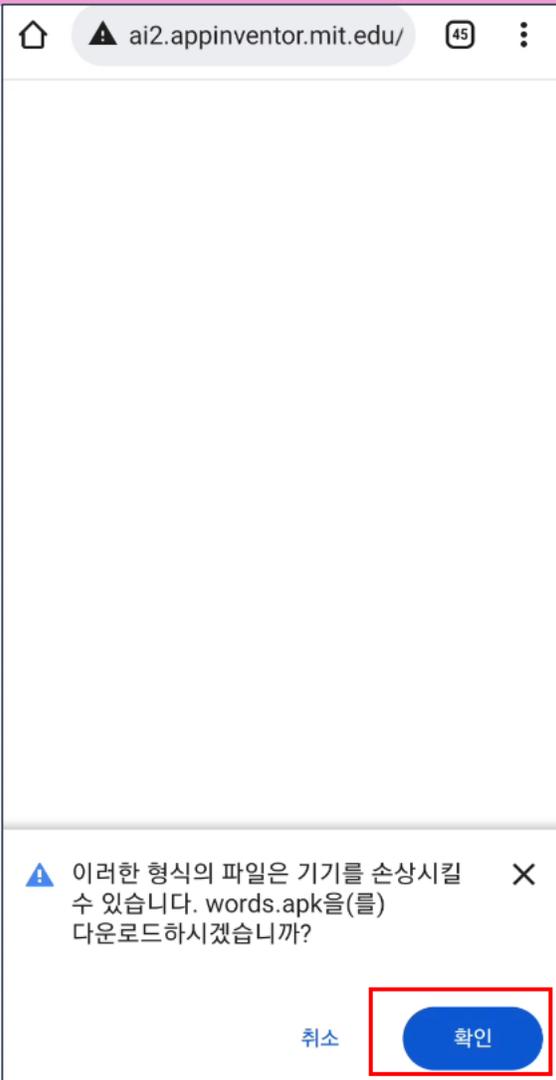
거절하기

번역할 텍스트   레이블1   텍스트

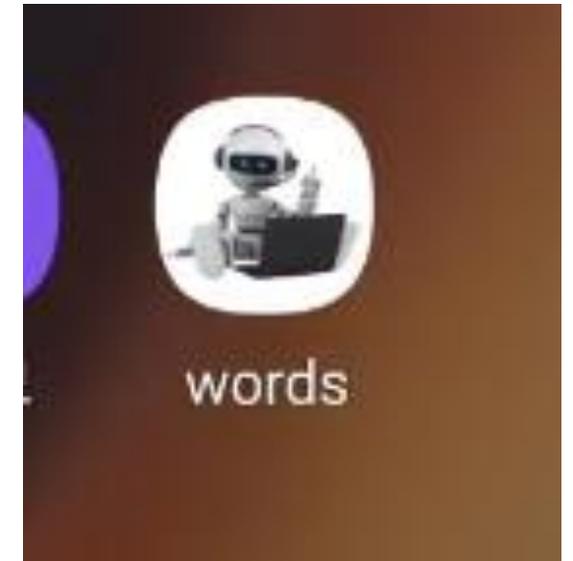
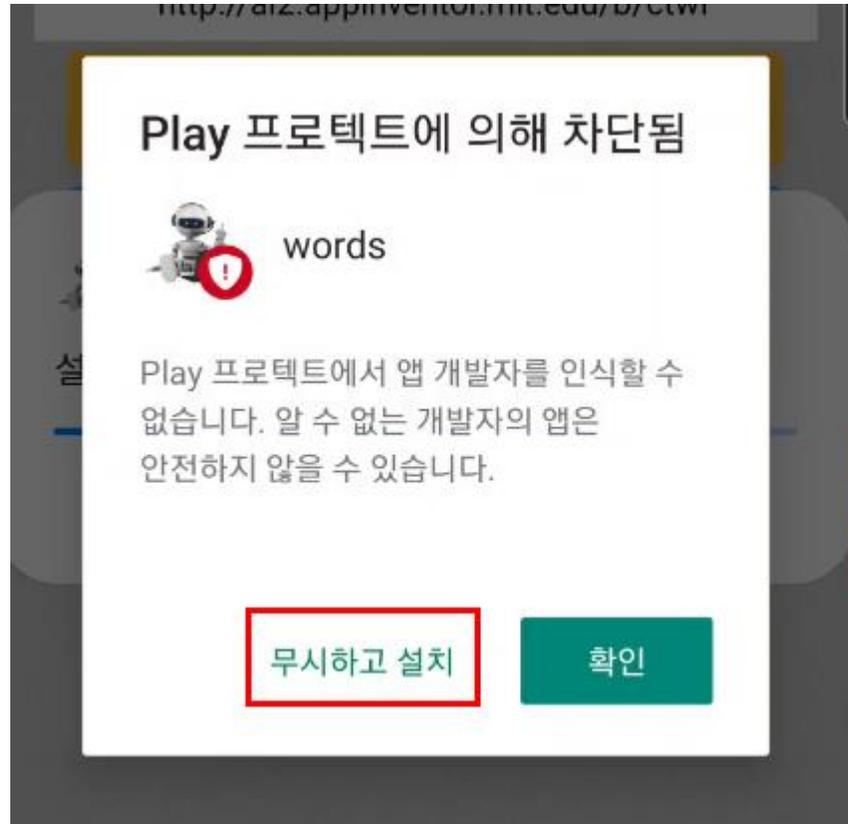
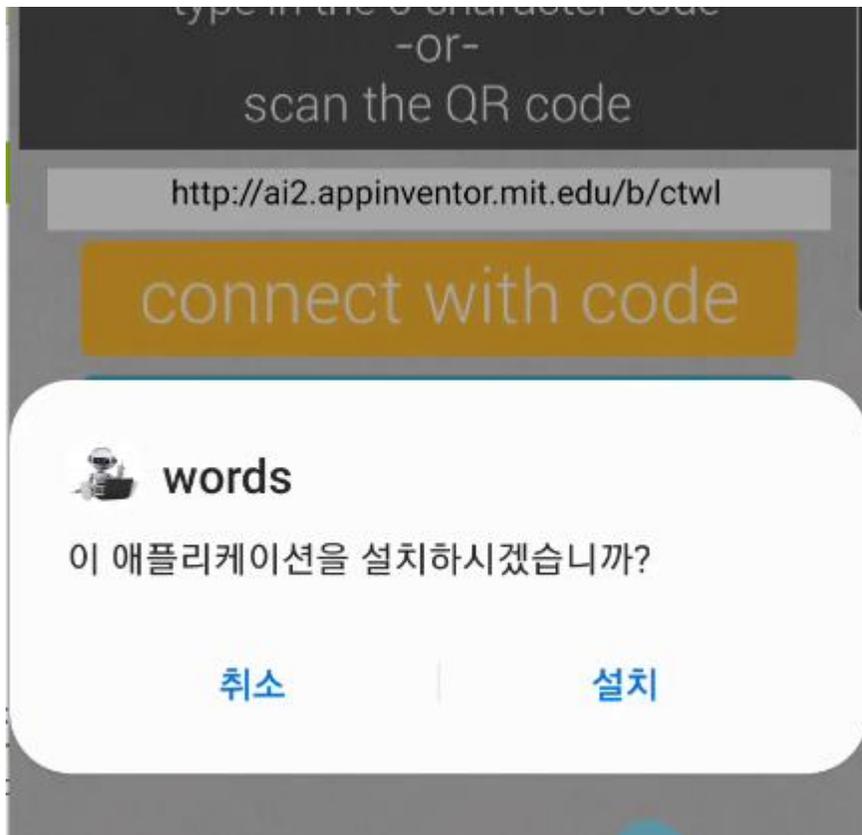
MIT AI2 COMPANION 앱으로  
QR코드를 찍는다.

항목 선택하  
리스트 안에서 해당 위치에  
1

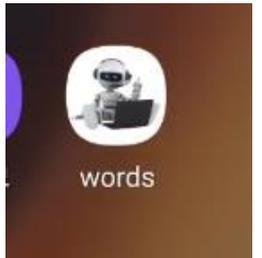
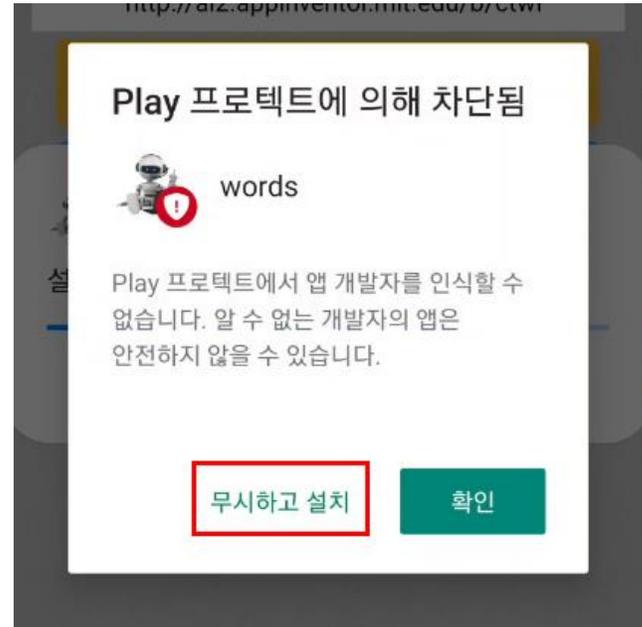
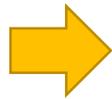
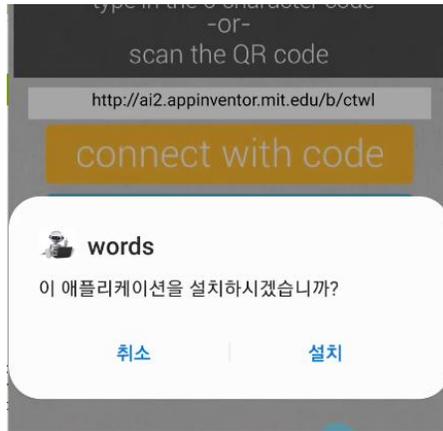
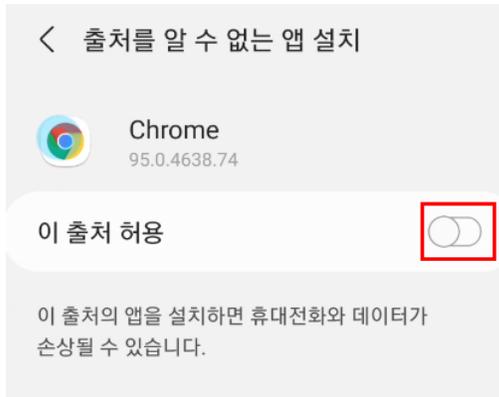
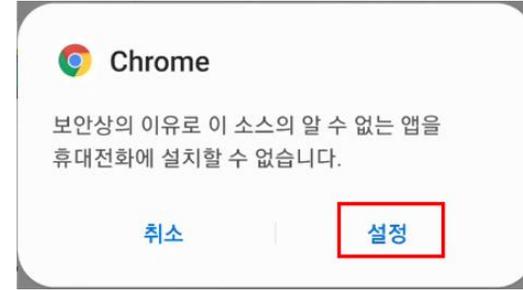
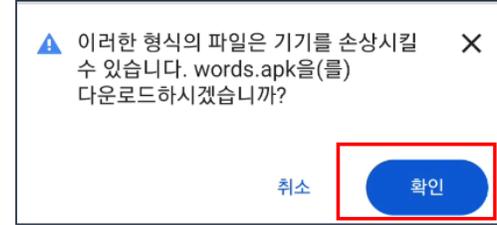
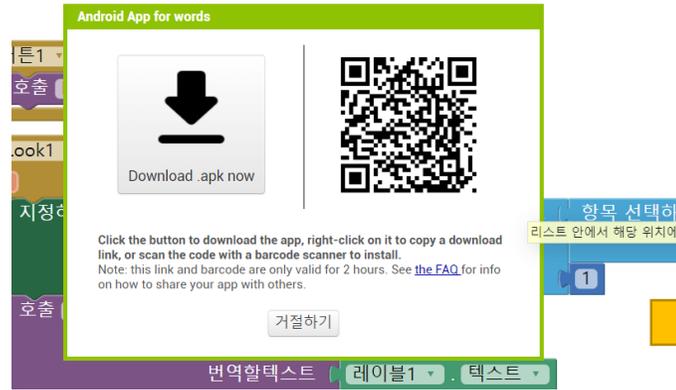
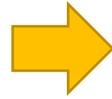
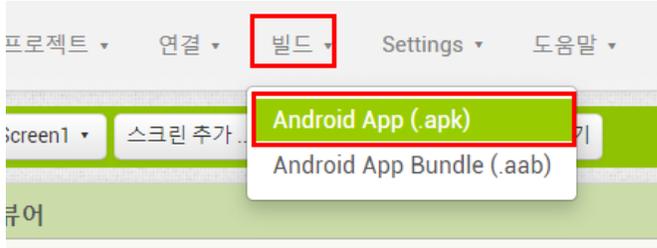
# 앱 제작의 실제7 – 영단어 학습



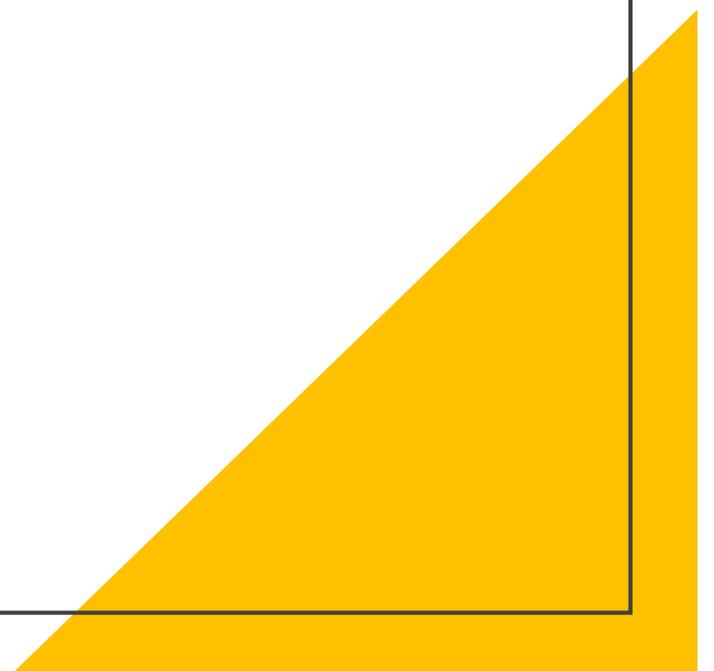
## 앱 제작의 실제7 – 영단어 학습



# 앱 제작의 실제7 – 영단어 학습



# 6. 마스크 분류 앱 만들기



## 앱 제작의 실제8 - 마스크 분류기

이미지 인식 확장 기능



### **Personallmage Classifier:**

개인이 학습시킨 이미지 분류(md1 파일) 모델 대로 이미지를 분류해 주는 기능

## 앱 제작의 실제8 - 마스크 분류기

### 마스크 경고 앱은 어떻게 작동할까?

- 1 얼굴을 카메라로 찍는다.
- 2 마스크 착용 유무를 분류한다.
- 3 마스크 착용 상태와 확률을 나타낸다.



# 앱 제작의 실제8 - 마스크 분류기



# 앱 제작의 실제8 – 마스크 분류기

<https://classifier.appinventor.mit.edu/>

Personal Image Classifier Train Test **Previous Version**

## Training Page

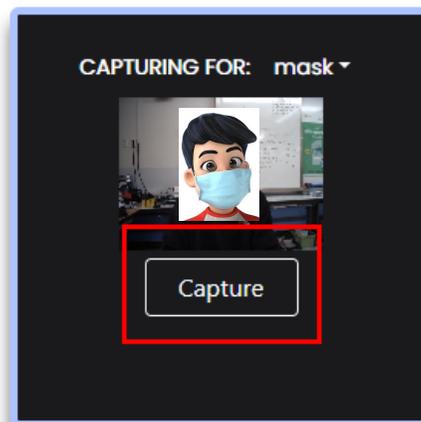
To get started, click the plus icon to add a classification and then use the "Capture" button or drag images into the capture box to add images to the selected classification. You can also upload previously generated data and models using the buttons below. When done, hit "Train"



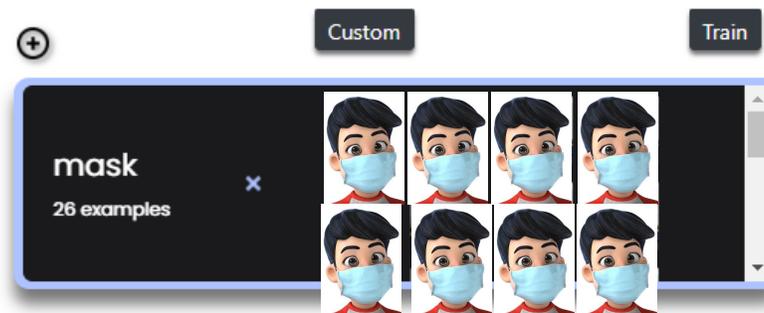
Upload Model

Upload Training Data

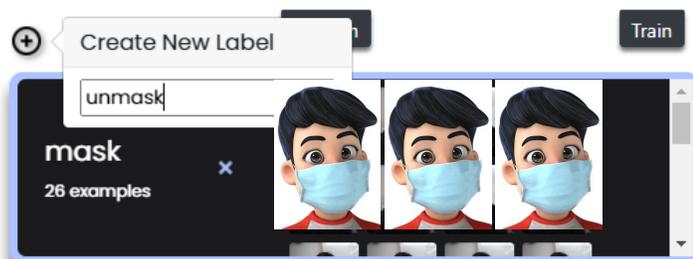
# 앱 제작의 실제8 – 마스크 분류기



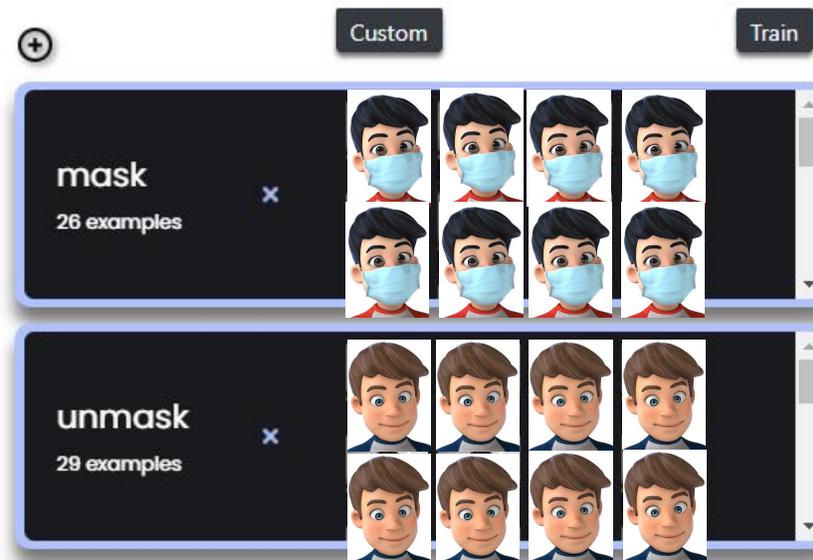
레이블 이름을 mask라고 한다  
Capture를 클릭해 사진을 찍는다  
(마스크를 쓴 상태의 사진)



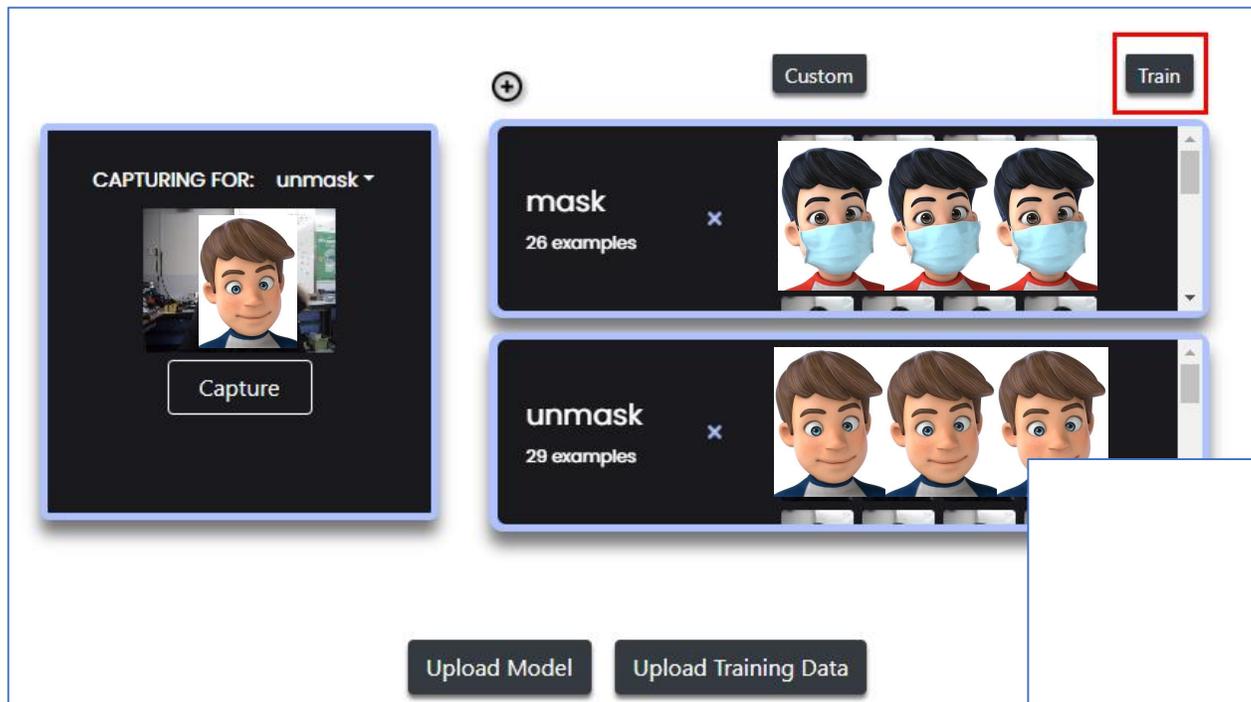
## 앱 제작의 실제8 – 마스크 분류기



레이블 이름을 unmask라고 한다  
Capture를 클릭해 사진을 찍는다  
(마스크를 벗은 상태의 사진)



## 앱 제작의 실제8 – 마스크 분류기



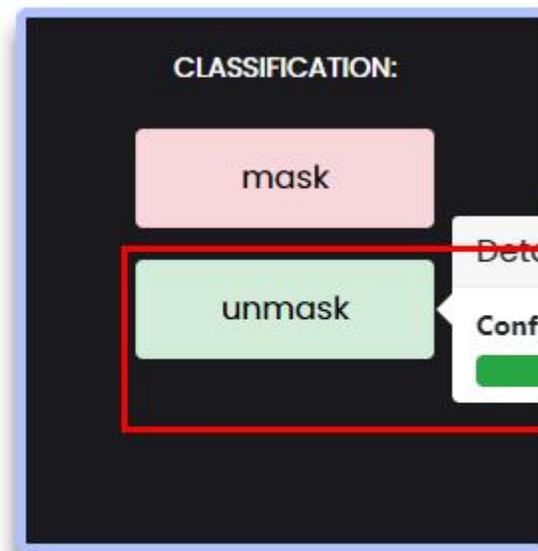
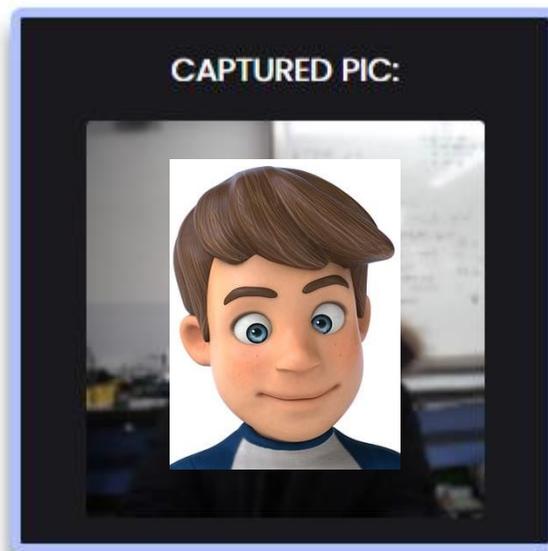
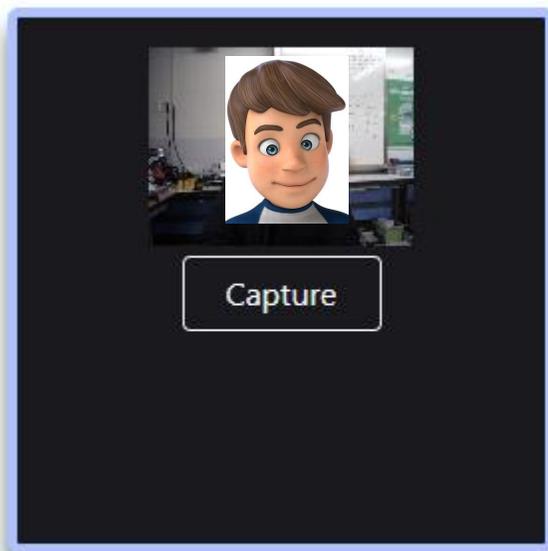
Train을 눌러서 학습시킨다.

Building custom model...



## 앱 제작의 실제8 - 마스크 분류기

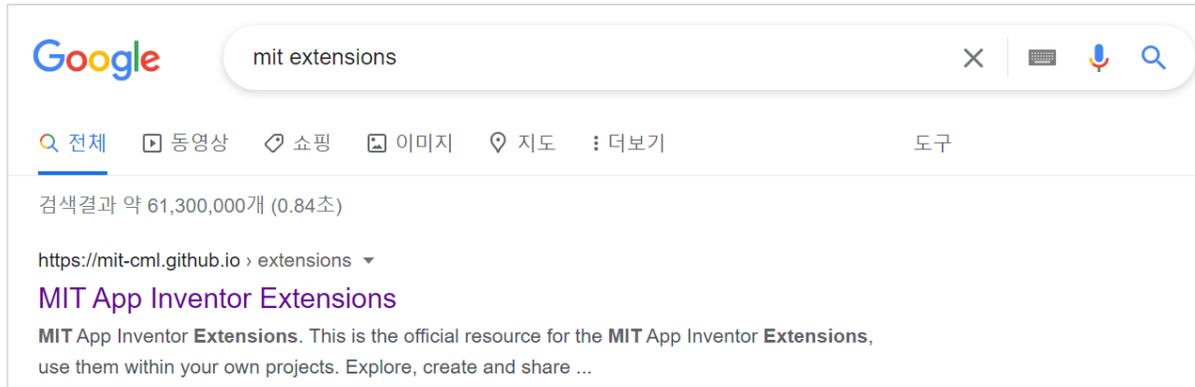
잘 학습되었는지 확인한 후에  
Export Model을 눌러 파일로 생성한다.



Export Model

Export Training Data

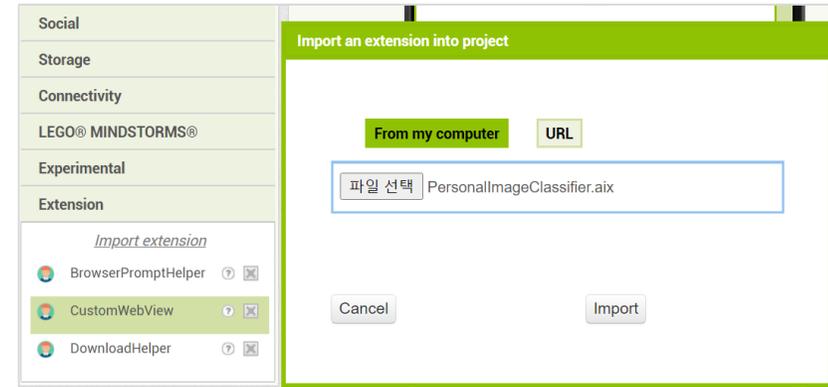
# 앱 제작의 실제8 – 마스크 분류기



## 1. MIT App Inventor Extensions 추가하기

Name	Description	Author	Version	Download .aix	Code
BluetoothLE	Adds as Bluetooth Low Energy functionality to your applications. See <a href="#">BluetoothLE Documentation and Resources</a> for more information.	MIT App Inventor	20200828	<a href="#">BluetoothLE.aix</a>	<a href="#">Via GitHub</a>
LookExtension	Adds object recognition using a neural network compiled into the extension.	MIT App Inventor	20181124	<a href="#">LookExtension.aix</a>	<a href="#">Via GitHub</a>
PersonalAudioClassifier	Use your own neural network classifier to recognize sounds with this extension.	MIT App Inventor	20200904	<a href="#">PersonalAudioClassifier.aix</a>	<a href="#">Via GitHub</a>
PersonallImageClassifier	Use your own neural network classifier to recognize images with this extension.	MIT App Inventor	20210315	<a href="#">PersonallImageClassifier.aix</a>	<a href="#">Via GitHub</a>
PosenetExtension	Estimate pose with this extension.	MIT App Inventor	20200226	<a href="#">Posenet.aix</a>	<a href="#">Via GitHub</a>
FaceMeshExtension	Estimate face landmarks with this extension.	MIT App Inventor	20210414	<a href="#">Facemesh.aix</a>	<a href="#">Via GitHub</a>

## 2. PersonallImageClassifier.aix 다운로드



## 3. Import Extension – 파일 업로드 – Import (15초 기다리기)

# 앱 제작의 실제8 - 마스크 분류기

레이아웃

미디어

그리기 & 애니메이션

지도

센서

소셜

저장소

연결

레고® 마인드스톰®

실험실

확장기능

*확장기능 추가하기*

Look

PersonallImageClassifier

사물 인식 버튼

PersonallImageClassifier1

레이블1 의 텍스트    레이블2 의 텍스트

발음 듣기

수평배치1

레이블1

레이블2

버튼2

안덱스번역1

음성변환1

Look1

PersonallImageClassifier1

최소간격

0

Model

model.mdl...

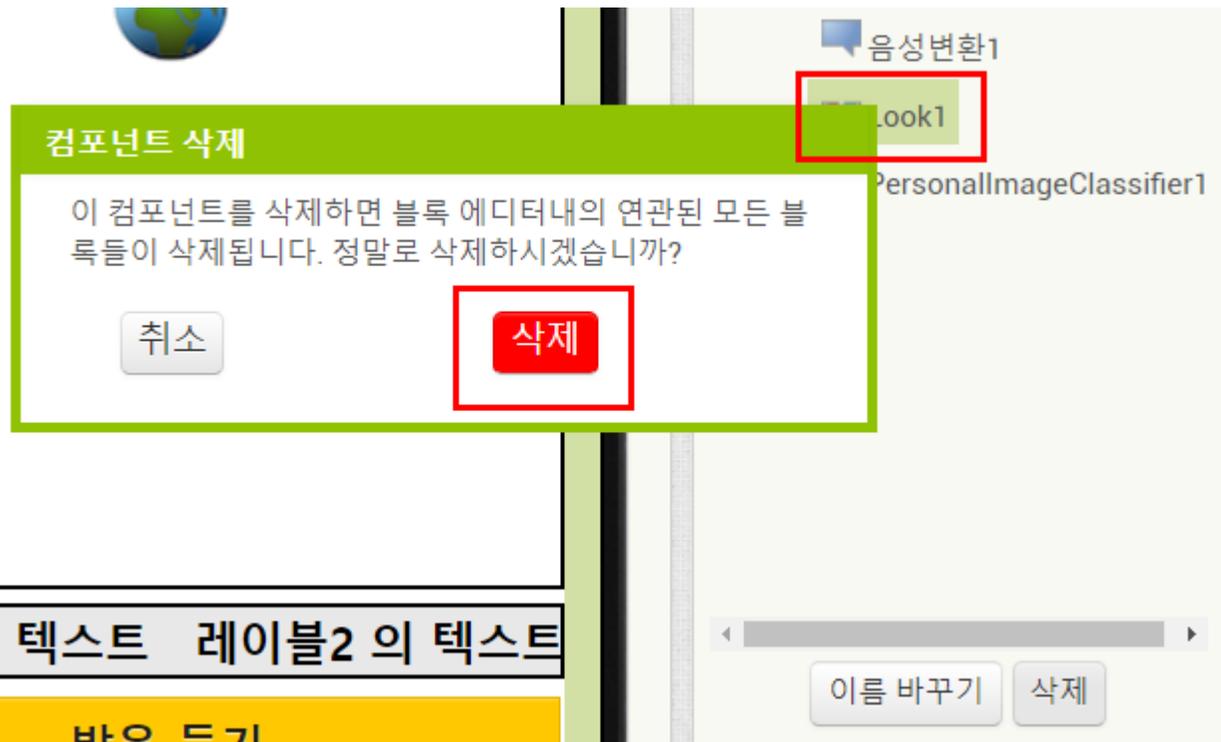
WebView

웹뷰어1...

이름 바꾸기    삭제

## 앱 제작의 실제8 - 마스크 분류기

Look 기능은 삭제한다.



블1의 텍스트 레이블2의 텍스트

반오 드기

이름 바꾸기

삭제

## 앱 제작의 실제8 - 마스크 분류기



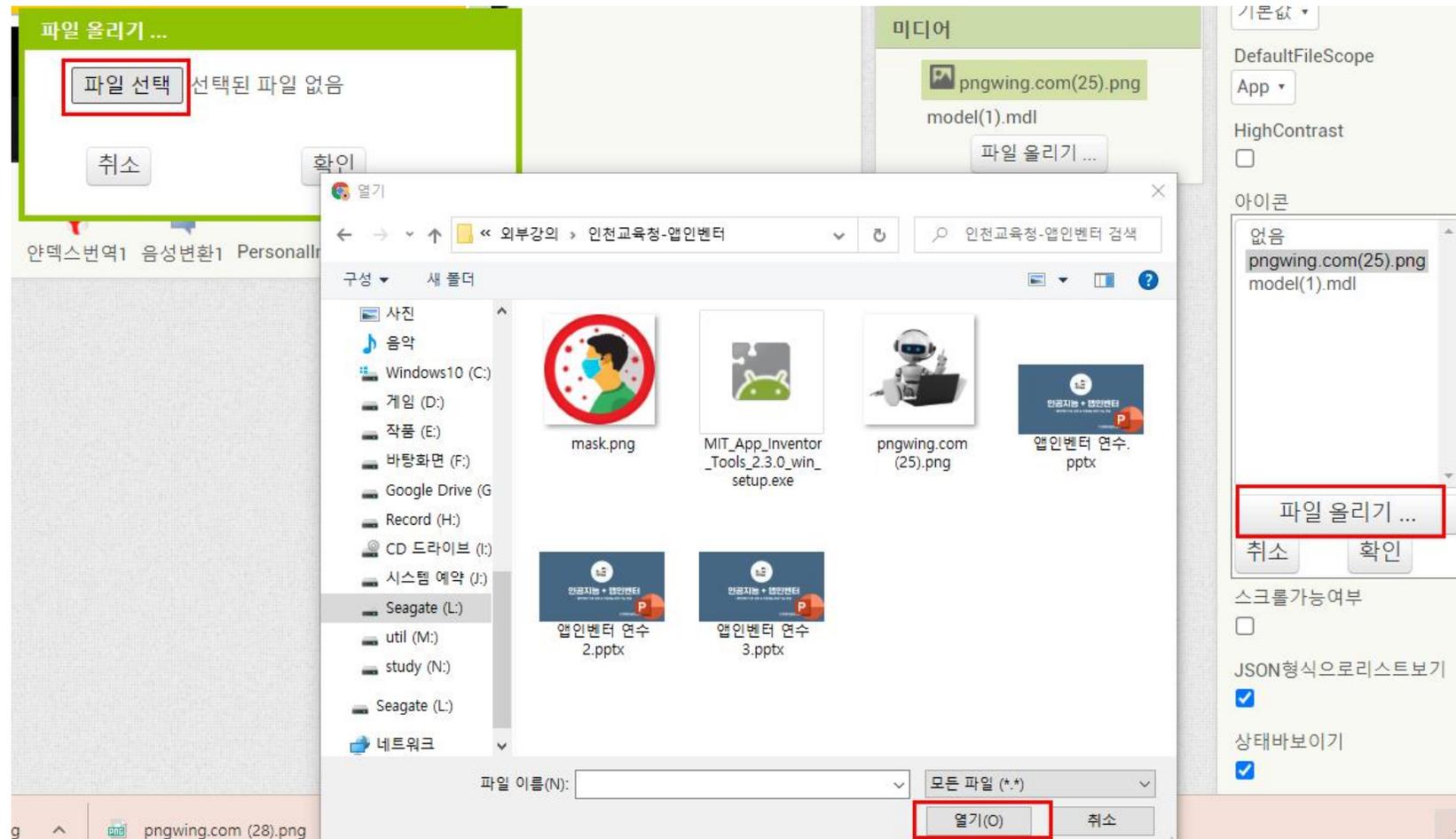
## 앱 제작의 실제8 – 마스크 분류기

The screenshot displays the interface of an application development tool, divided into two main sections: '컴포넌트' (Components) on the left and '속성' (Properties) on the right.

**컴포넌트 (Components):** This panel lists the components added to the application. The 'Screen1' component is highlighted with a red box. Below it, other components are listed, including '버튼1' (Button1), '웹뷰어1' (WebView1), '수평배치1' (HorizontalLayout1), '레이블1' (Label1), '레이블2' (Label2), '버튼2' (Button2), '얀덱스번역1' (YandexTranslate1), '음성변환1' (SpeechRecognition1), and 'PersonallImageClassifier1'.

**속성 (Properties):** This panel shows the properties for the selected 'Screen1' component. The '앱이름' (App Name) property is highlighted with a red box and contains the text 'mask'. Other visible properties include '스크린설명' (Screen Description), '강조색' (Accent Color) set to '기본값' (Default), '수평정렬' (Horizontal Alignment) set to '가운데 : 3' (Center : 3), and '수직정렬' (Vertical Alignment) set to '위 : 1' (Top : 1).

# 앱 제작의 실제8 – 마스크 분류기



## 앱 제작의 실제8 - 마스크 분류기

버튼1을 클릭하면 사물을 분류한다.

언제 **버튼1** .클릭했을때

실행 **호출** **PersonallImageClassifier1** .ClassifyVideoData

언제 **PersonallImageClassifier1** .GotClassification

**결과**

실행 **지정하기** **레이블1** . **텍스트** 값 **가져오기** **결과**

분류한 결과 값을 레이블1에 보여준다.

# 앱 제작의 실제8 – 마스크 분류기

언제 버튼1 ▾ .클릭했을때

실행 호출 PersonallImageClassifier1 ▾ .ClassifyVideoData

언제 PersonallImageClassifier1 ▾ .GotClassification

결과

실행 지정하기 레이블1 ▾ . 텍스트 ▾ 값 가져오기 결과 ▾

언제 버튼2 ▾ .클릭했을때

실행

호출 음성변환1 ▾ .말하기  
메시지 레이블1

버튼 2 안에 있던 것을 버림

## 앱 제작의 실제8 – 마스크 분류기

언제 버튼1 ▾ .클릭했을때

실행 호출 PersonallImageClassifier1 ▾ .ClassifyVideoData

언제 PersonallImageClassifier1 ▾ .GotClassification

결과

실행 지정하기 레이블1 ▾ . 텍스트 ▾ 값 가져오기 결과 ▾

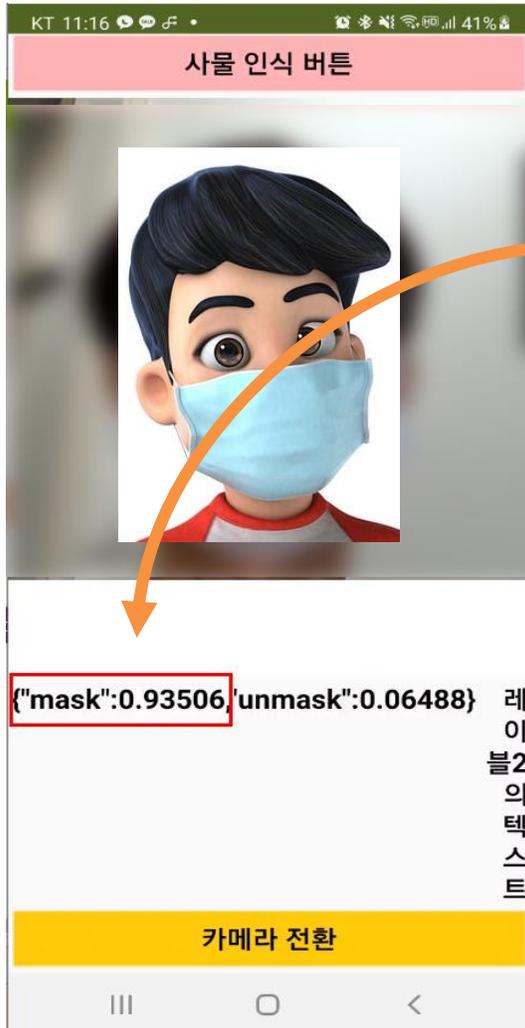
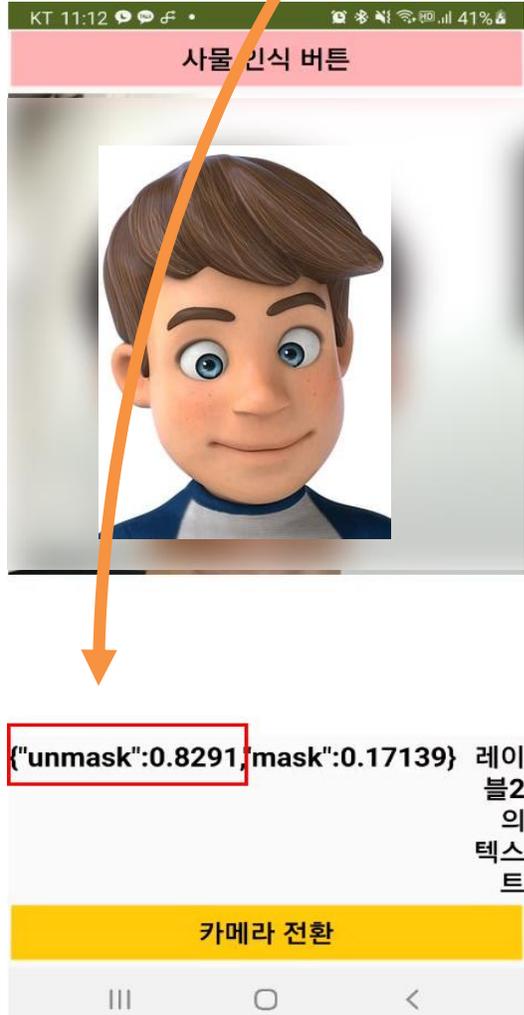
언제 버튼2 ▾ .클릭했을때

실행 호출 PersonallImageClassifier1 ▾ .ToggleCameraFacingMode

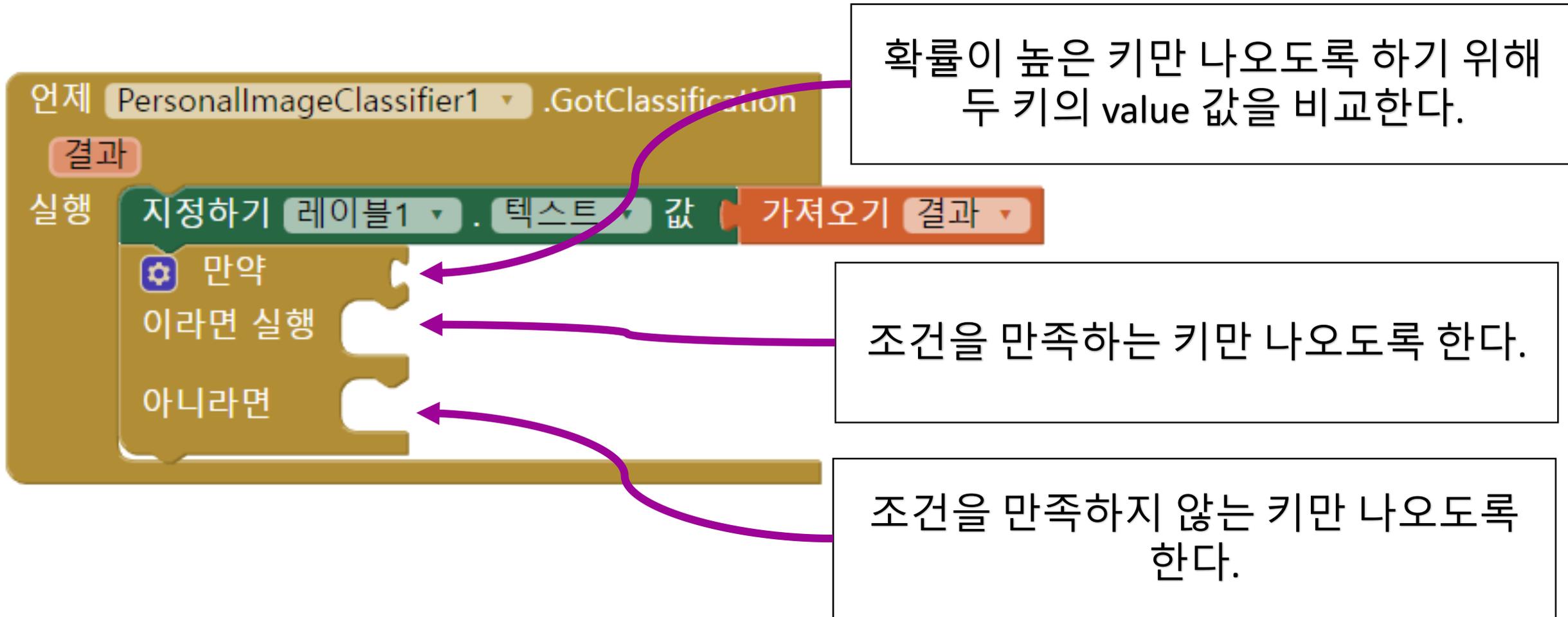
카메라 전환 기능 블록

# 앱 제작의 실제8 - 마스크 분류기

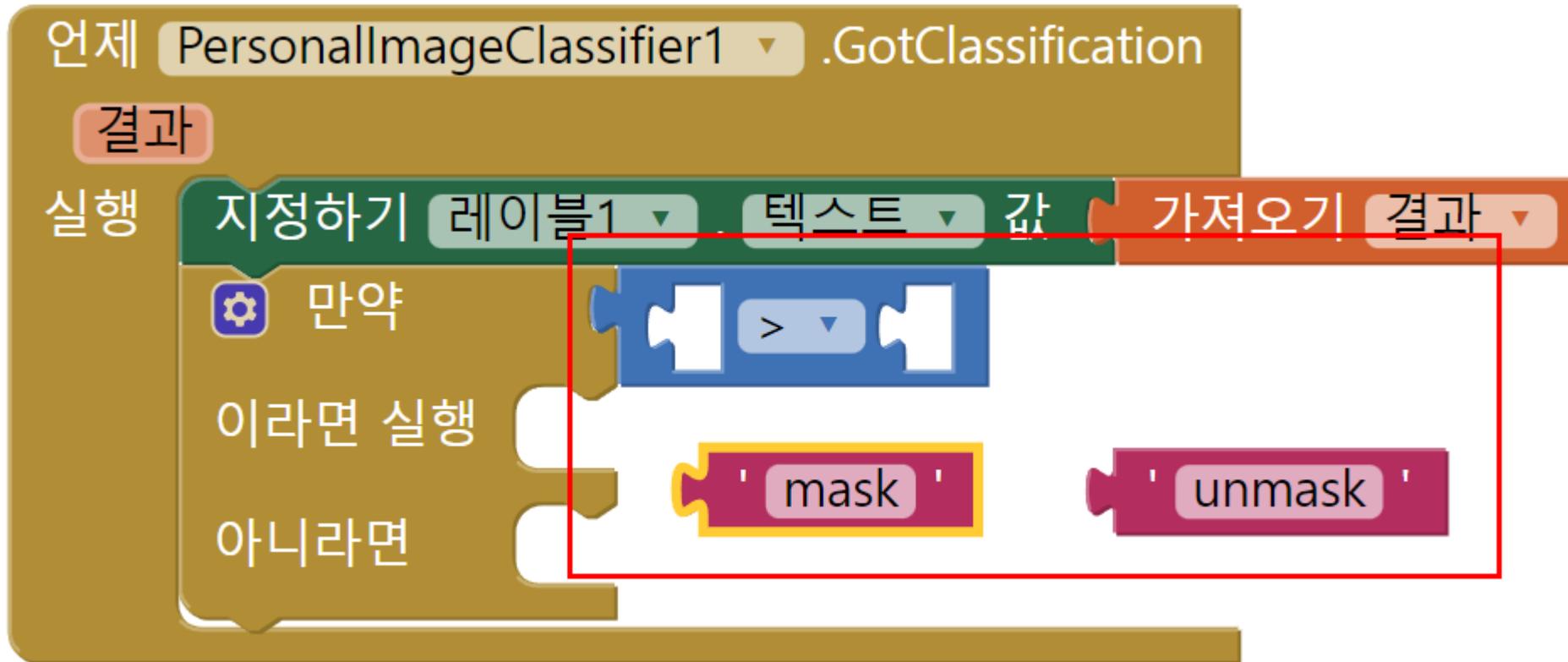
인식 결과가 딕셔너리의 형태로 나타난다  
{키:value}



## 앱 제작의 실제8 - 마스크 분류기



## 앱 제작의 실제8 - 마스크 분류기



# 앱 제작의 실제8 - 마스크 분류기

```
{ "mask": 0.93506, "unmask": 0.06488 }
```

언제 PersonallImageClassifier1 .GotClassification  
결과

실행 지정하기 레이블1 . 텍스트 값 가져오기 결과

만약

- get value for key 'mask' in dictionary or if not found 가져오기 결과
- >
- get value for key 'unmask' in dictionary or if not found 가져오기 결과

이러면 실행  
아니라면

# 앱 제작의 실제8 - 마스크 분류기

언제 PersonallImageClassifier1 .GotClassification

결과

실행

지정하기 레이블1 . 텍스트 값 가져오기 결과

만약

get value for key ' mask ' in dictionary 가져오기 결과  
or if not found ' not found '

> get value for key ' unmask ' in dictionary 가져오기 결과  
or if not found ' not found '

이러면 실행 지정하기 레이블1 . 텍스트 값 ' 마스크 착용 '

아니라면 지정하기 레이블1 . 텍스트 값 ' 마스크 미착용 '



## 앱 제작의 실제8 – 마스크 분류기

마스크 미착용 레이블2 의 텍스트

카메라 전환

# 앱 제작의 실제8 - 마스크 분류기

마스크 착용 0.7002

카메라 전환

실행

지정하기 레이블1 . 텍스트 값 가져오기 결과

만약

get value for key 'mask' in dictionary 가져오기 결과  
or if not found 'not found'

> get value for key 'unmask' in dictionary 가져오기 결과  
or if not found 'not found'

이러면 실행

지정하기 레이블1 . 텍스트 값 '마스크 착용'

지정하기 레이블2 . 텍스트 값  
get value for key 'mask' in dictionary 가져오기 결과  
or if not found 'not found'

아니라면

지정하기 레이블1 . 텍스트 값 '마스크 미착용'

# 앱 제작의 실제8 - 마스크 분류기

마스크 착용 **85.889**

카메라 전환

만약

get value for key  
in dictionary  
or if not found

' mask '

가져오기 결과

' not found '

>

get value for key  
in dictionary  
or if not found

' unmask '

가져오기 결과

' not found '

이러면 실행

지정하기 레이블1 . 텍스트 값

마스크 착용

지정하기 레이블2 . 텍스트 값

×

get value for key  
in dictionary  
or if not found

' mask '

가져오기 결과

' not found '

100

아니라면

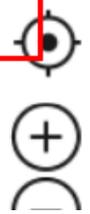
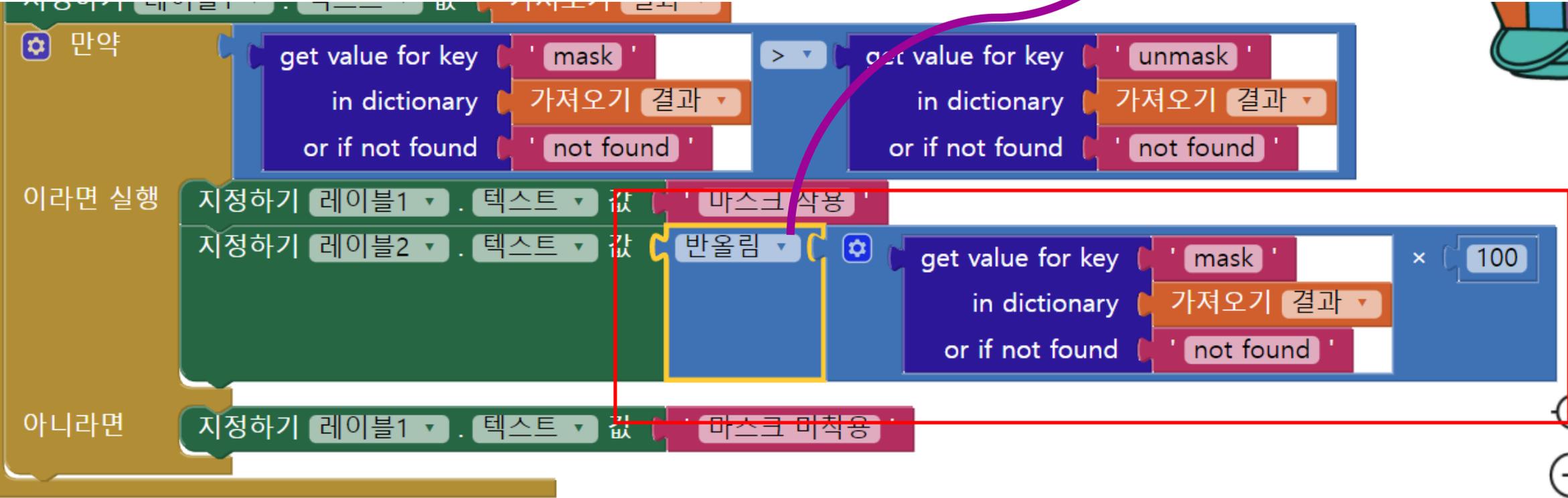
지정하기 레이블1 . 텍스트 값

마스크 미착용

# 앱 제작의 실제8 - 마스크 분류기

마스크 착용 94

카메라 전환



# 앱 제작의 실제8 - 마스크 분류기

마스크 착용 80%

카메라 전환

만약

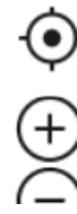
- get value for key 'mask' in dictionary or if not found 'not found' > get value for key 'unmask' in dictionary or if not found 'not found'

이러면 실행

- 지정하기 레이블1 . 텍스트 값 마스크 착용
- 지정하기 레이블2 . 텍스트 값 합치기 반올림 %
- get value for key 'mask' in dictionary or if not found 'not found' × 100

아니라면

- 지정하기 레이블1 . 텍스트 값 마스크 미착용

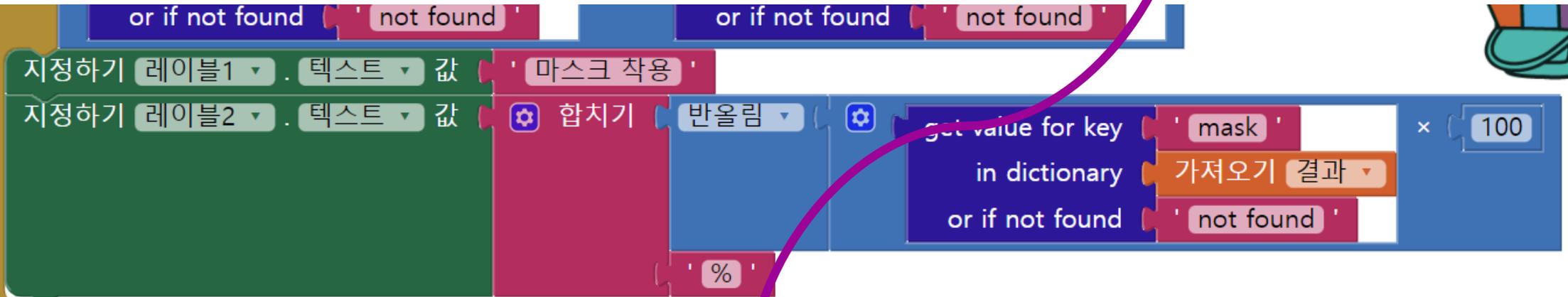


# 앱 제작의 실제8 - 마스크 분류기

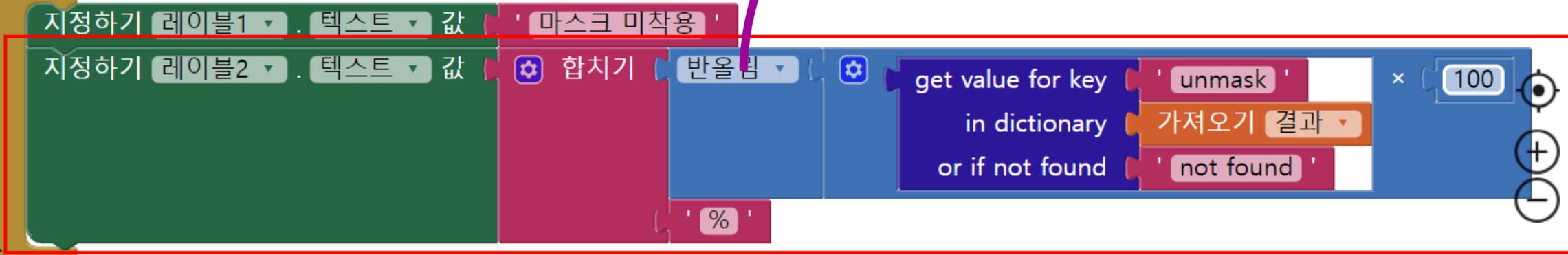
마스크 미착용 89%

카메라 전환

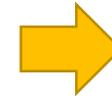
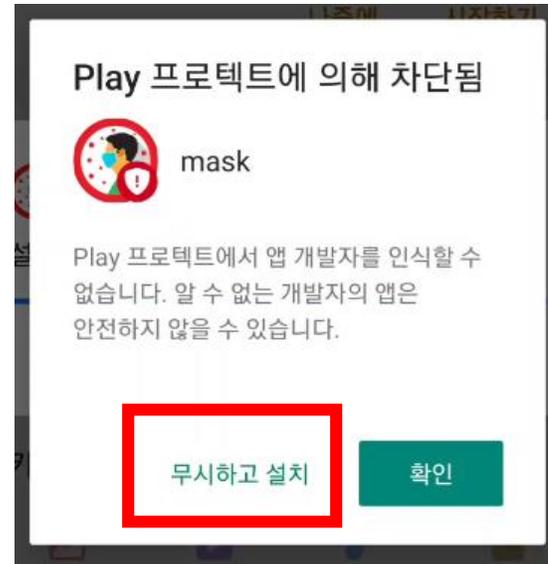
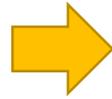
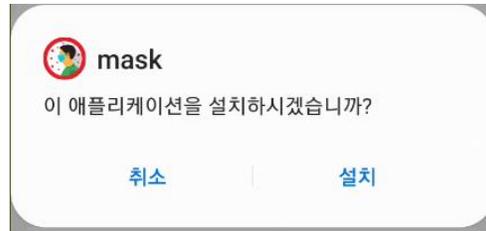
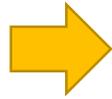
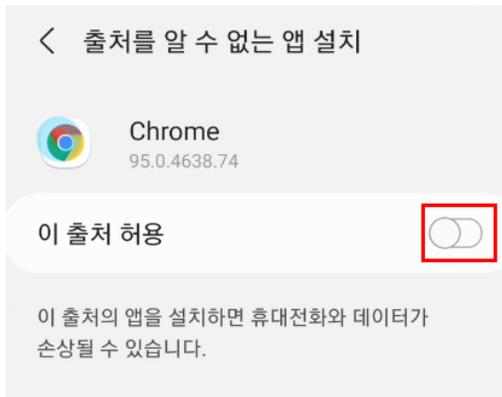
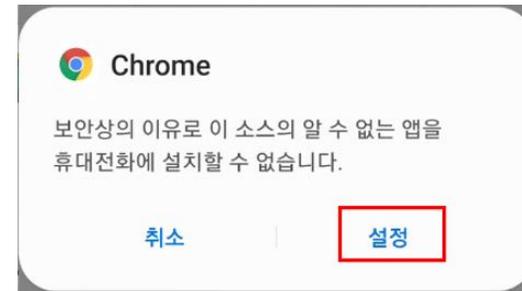
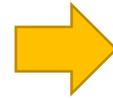
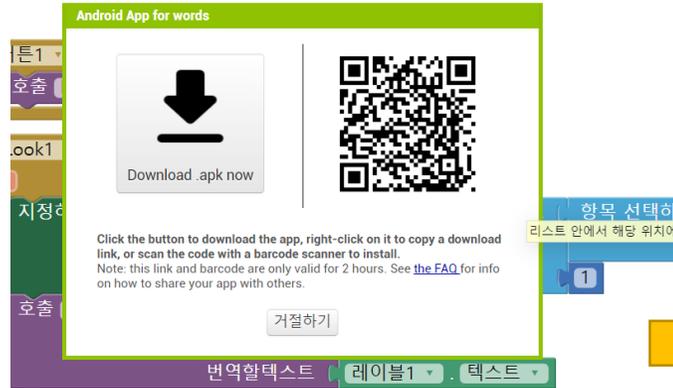
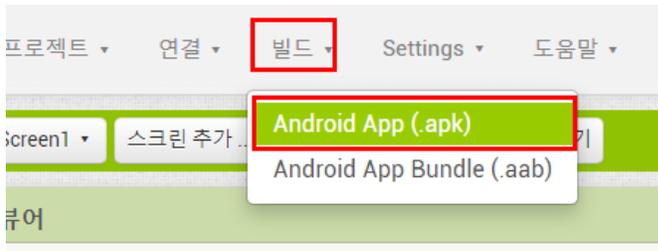
이러면 실행



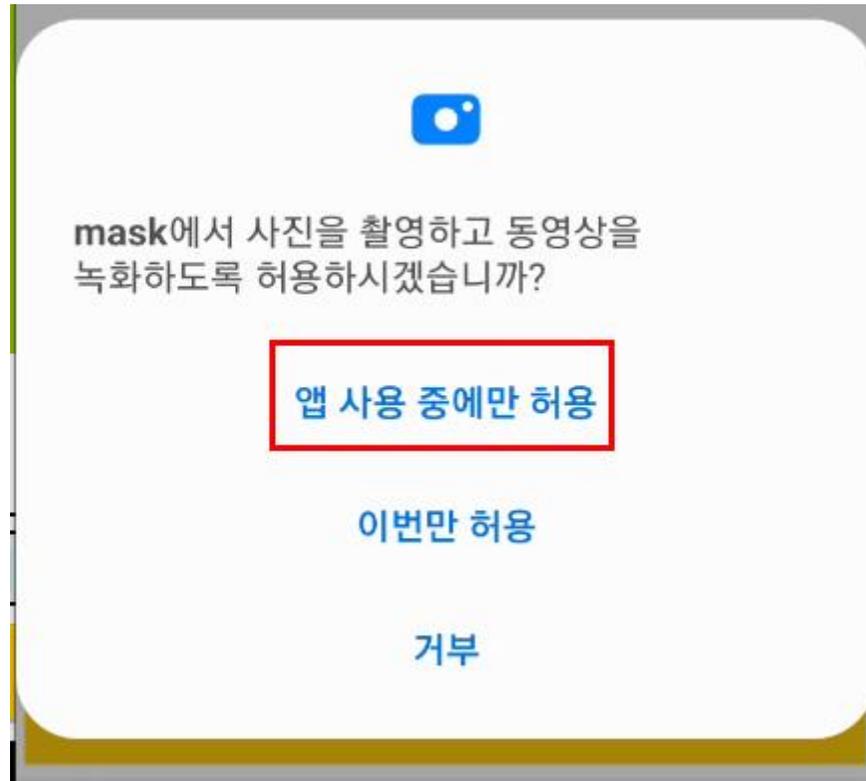
아니라면



# 앱 제작의 실제8 - 마스크 분류기



## 앱 제작의 실제8 - 마스크 분류기



## 앱 제작의 실제8 - 마스크 분류기

### 유의점

- 웹뷰어 화면이 안 보이고 RECORD라는 빨간 단추가 보인다면 확장 기능을 오디오분류로 잘못 가져온 것이다.
- 코딩에 틀린 게 없는데 컴패니언 앱에서 제대로 작동하지 않는다면 연결이 끊어진 것이니 다시 연결하도록 한다.
- 중간에 디자이너에서 수정하면 웹뷰어에 카메라 화면이 안 보이니 스마트폰과 연결을 끊고 다시 연결한다.

## 앱 제작의 실제8 - 마스크 분류기

### 유의점

- 연결이 끊어진 경우 앱인벤터 - 연결 - 다시 연결하기를 클릭한 후에 다시 앱인벤터 - 연결 - AI 컴패니언을 클릭한다. 스마트폰에서도 MIT 컴패니언 앱을 종료한 후에 다시 실행한다.