

# 앱인벤터 (Appinventor)



앱인벤터는 구글이 제작하다가, 현재는 MIT가 관리하는 오픈소스 앱메이커이다. 사용하려면 구글 계정이 필요하다.

컴퓨터 프로그래밍을 처음 접하는 사람들이 쉽게 안드로이드 운영 체제용 응용 소프트웨어를 만들 수 있게 해준다. 스크래치와 매우 비슷한 그래픽 인터페이스를 사용하므로 사용자들이 코드 블록을 끌어당기거나 붙여서 안드로이드 장치에서 실행할 응용 프로그램들을 만들 수 있다.

앱인벤터는 2010년 7월 12일에 요청을 통해 사용이 가능하게 되었으며 2010년 12월 15일에 공식 출시되었다. 2011년 후반기에 구글은 소스 코드를 공개하고 서버를 종료하였으나, 앱인벤터 제작자에 의해 MIT로 넘어가게 되었다. MIT 버전은 2012년 3월에 시작되었다. 그러다 2013년 12월 6일에 MIT는 앱인벤터 2를 출시하였다.

앱인벤터의 프로젝트를 추출하면 \*.aia 형식의 파일이 나오고, 앱인벤터의 추가 기능은 \*.aix의 파일 형식을 가지고 있다.

Java와 안드로이드 SDK를 다루지 않아도 안드로이드 앱 개발이 가능하다는 것이다. 앱인벤터로 개발한 앱을 구글 플레이에 올릴수 있으며 유료 앱으로 올려도 상관없다.

섬세한 애플리케이션 개발은 불가능하다. 원래 기본적인 앱 개발을 위해 만들어진 사이트이다 보니 스프라이트 객체 개념이 없고 객체를 나타내려고 해도 제한이 있기 때문에 애플리케이션을 만들 수 있는 범위가 한정적이다.

## 앱인벤터

- 원래 구글이 제공한 오픈 소스 웹 애플리케이션으로, 지금은 MIT에 의해 관리
- 초보자들도 쉽게 안드로이드 운영 체제용 응용 소프트웨어를 만들 수 있게 해 줌
- 앱을 개발할 수 있는 직관적이고 시각적 프로그래밍 환경
- <http://appinventor.mit.edu>



### 앱인벤터시작하기

1. 앱 인벤터를 서비스하는 인터넷 주소(URL)는 <http://ai2.appinventor.mit.edu/>이다.
2. 접속한후 구글 계정으로 로그인이 한다.
3. '새 프로젝트 시작하기' 버튼을 클릭하고 프로젝트 이름을 입력하고 '확인'을 클릭하여 새로운 프로젝트를 생성한다.

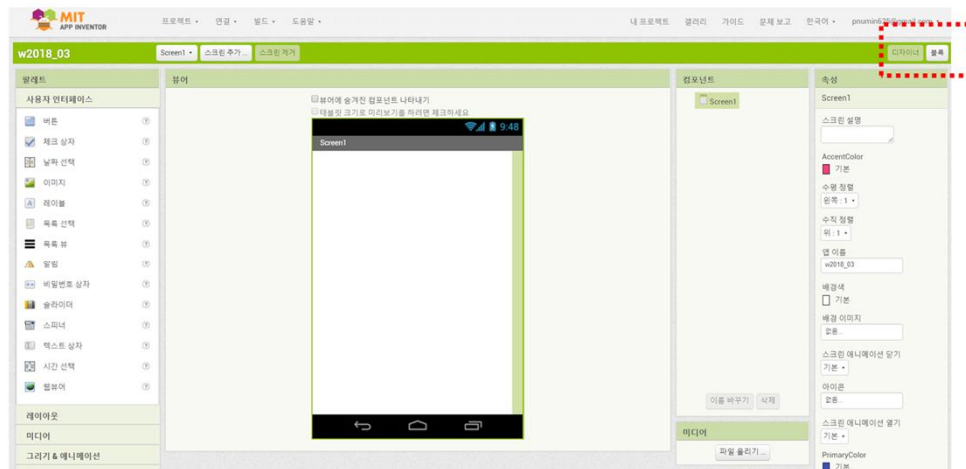
※ 프로젝트 이름은 영문대소문자와 '\_'로만 입력함.

※ 한글명, 기타 띄어쓰기, 특수문자 등은 모두 프로젝트 이름으로 입력할 수 없음

## 앱인벤터 – 순차



# 앱인벤터 화면 구성 - 디자이너



앱인벤터의 프로젝트 화면 구성은 디자이너와 블록 화면으로 구분되어 있다. 프로젝트를 시작하면 디자이너 화면에서 시작되며 우측 상단의 '블록' 버튼을 클릭하여 블록 화면으로 전환 할 수 있다. 역시 반대로 우측 상단의 '디자이너' 버튼을 클릭하여 블록화면에서 디자이너 화면으로 전환 할 수 있다.

디자이너 화면은 팔레트, 뷰어, 컴포넌트, 미디어, 속성창으로 구성된다.

- **팔레트**: 앱 제작에 필요한 컴포넌트의 집합으로 필요한 컴포넌트를 드래그하여 뷰어에 추가하여 사용
- **뷰어**: 레이아웃, 컴포넌트의 배치 등 앱이 실행시 표시되는 화면
- **컴포넌트**: 팔레트에서 드래그하여 뷰어에 추가된 컴포넌트의 목록
- **미디어**: 앱에서 사용할 오디오, 이미지 파일등이 등록된 목록, 사용자가 직접 추가
- **속성**: 선택한 컴포넌트의 상세 속성을 변하는 창

# 앱인벤터 컴포넌트

사용자 인터페이스	
<ul style="list-style-type: none"> <li>버튼</li> <li>체크 상자</li> <li>날짜 선택</li> <li>이미지</li> </ul>	<p><b>버튼</b></p> <p>버튼은 클릭을 하면 연결된 동작을 수행하는 컴포넌트입니다. 다양한 모양으로 변경할 수 있으며, 클릭을 가능하게 할지 (활성 여부)도 디자이너 또는 블록 에디터에서 변경할 수 있습니다.</p> <p><a href="#">자세히 알아보기</a></p>
<ul style="list-style-type: none"> <li>레이블</li> <li>목록 선택</li> <li>목록 뷰</li> <li>알림</li> </ul>	<p><b>레이블</b></p> <p>레이블은 텍스트 속성에 지정된 글을 화면에 표시하는 컴포넌트입니다. 디자이너 또는 블록 에디터에서 텍스트의 모양이나 위치를 변경하거나 다른 속성들을 지정할 수 있습니다.</p> <p><a href="#">자세히 알아보기</a></p>
<ul style="list-style-type: none"> <li>텍스트 상자</li> <li>시간 선택</li> <li>웹뷰어</li> </ul>	<p><b>텍스트 상자</b></p> <p>사용자가 텍스트를 입력할 수 있는 상자입니다. 초기값이나 사용자가 입력한 텍스트는 텍스트 속성에 지정됩니다. 만약 값이 비어있다면, 힌트 속성의 값이 흐리게 텍스트 상자에 표시되 무엇을 입력해야 할지 사용자에게 알려주는 역할을 합니다.</p>



## 앱인벤터 컴포넌트

### • 버튼

- 버튼은 클릭을 하면 연결된 동작을 수행하는 컴포넌트
- 다양한 모양으로 변경할 수 있음
- 클릭을 가능하게 할지 (활성 여부)도 디자이너 또는 블록 에디터에서 변경할 수 있음

### • 레이블

- 텍스트 속성에 지정된 글을 화면에 표시하는 컴포넌트
- 디자이너 또는 블록 에디터에서 텍스트의 모양이나 위치를 변경하거나 다른 속성들을 지정할 수 있음

### • 텍스트상자

- 사용자가 텍스트를 입력할 수 있는 상자
- 초기값이나 사용자가 입력한 텍스트는 텍스트 속성에 지정
- 여러 줄 속성은 한 줄 이상 입력 하도록 상자를 변경함
- 숫자만 속성은 숫자 값만 입력하도록 제한

# 앱인벤터 컴포넌트

사용자 인터페이스

버튼

체크 상자

날짜 선택

이미지

레이블

목록 선택

목록 뷰

알림

비밀번호 상자

슬라이더

스피너

텍스트 상자

시간 선택

웹뷰어

레이아웃

알림

알림 컴포넌트는 다음과 같은 함수를 사용하여 경고창, 메시지, 임시 경고창 화면에 표시하거나 안드로이드 로고를 생성합니다:

- 메시지창 나타내기: 메시지를 화면에 표시합니다. 사용자가 버튼을 누르면 사라집니다.
- 선택 대화창 나타내기: 메시지와 함께 사용자가 선택 할 수 있는 두 개의 버튼 (예를 들어, 예 또는 아니오)을 나타냅니다. 두 개 중 하나를 선택하면 선택 한 후 이벤트가 발생합니다.
- 텍스트 대화창 나타내기: 사용자가 메시지에 대한 응답을 입력하도록 합니다. 입력을 하면 텍스트 입력 후 이벤트가 발생합니다.
- 경고창 나타내기: 짧은 시간이 지나면 자동으로 사라지는 일시적인 경고창을 표시합니다.
- 오류 로그: 안드로이드에 오류 메시지 로그를 남깁니다.
- 정보 로그: 안드로이드에 정보 메시지 로그를 남깁니다.
- 경고 로그: 안드로이드에 경고 메시지 로그를 남깁니다.
- 대화창 (경고창이 아닌)의 메시지들은 다음과 같은 HTML 태그를 사용하여 모양을 지정할 수 있습니다: <b>, <big>, <blockquote>, <br>, <cite>, <dfn>, <div>, <em>, <small>, <strong>, <sub>, <sup>, <tt>, <u>
- 또한 폰트 태그를 사용하여 색상을 변경 할 수도 있습니다. 예를 들어, <font color="blue">. 이용 가능한 색상은 aqua, black, blue, fuchsia, green, grey, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow 입니다.

자세히 알아보기

## 앱인벤터 컴포넌트

- 알림 컴포넌트는 다음과 같은 함수를 사용하여 경고창, 메시지, 임시 경고창 화면에 표시하거나 안드로이드 로고를 생성한다.
  - 메시지창 나타내기: 메시지를 화면에 표시하고 사용자가 버튼을 누르면 사라짐
  - 선택 대화창 나타내기: 메시지와 함께 사용자가 선택 할 수 있는 두 개의 버튼 (예를 들어, 예 또는 아니오)을 나타내고 두 개 중 하나를 선택 하면 선택 한 후 이벤트가 발생
  - 텍스트 대화창 나타내기: 사용자가 메시지에 대한 응답을 입력하도록 함으로 입력을 하면 텍스트 입력 후 이벤트가 발생
  - 경고창 나타내기: 짧은 시간이 지나면 자동으로 사라지는 일시적인 경고창을 표시
  - 오류 로그: 안드로이드에 오류 메시지 로그를 남김
  - 정보 로그: 안드로이드에 정보 메시지 로그를 남김
  - 경고 로그: 안드로이드에 경고 메시지 로그를 남김

## 앱인벤터 컴포넌트(센서)

센서	
가속도 센서	가속도 센서
바코드 스캐너	흔들림을 감지하여 3차원 공간에서의 가속도 근사값을 측정하는 보이지 않는 컴포넌트입니다(SI 단위 $m/s^2$ 사용). 컴포넌트는 다음과 같이 구성되어 있습니다:
시계	<ul style="list-style-type: none"> <li><b>x가속도:</b> 스마트폰이 지면에 평평하게 놓여있을 경우 0의 값을 가집니다. 스마트폰이 오른쪽으로 기울어지면 (즉, 왼쪽 면이 들려지면) 양의 값을, 왼쪽으로 기울어지면 (즉, 오른쪽 면이 들려지면) 음의 값을 가지게 됩니다.</li> <li><b>y가속도:</b> 스마트폰이 지면에 평평하게 놓여있을 경우 0의 값을 가집니다. 스마트폰의 아래쪽이 들려지면 양의 값을, 반대로 위쪽이 들려지면 음의 값을 가지게 됩니다.</li> <li><b>z가속도:</b> 디스플레이가 위쪽을 바라본 채로 스마트폰이 지면과 평행하게 놓여져 있을 때 -9.8 (중력 가속도, 단위는 미터/초<sup>2</sup>) 값을 가집니다. 지면과 수직하게 놓여져 있다면 0, 지면과 평행하면서 디스플레이가 아래쪽을 바라본다면 +9.8이 됩니다. 이 값은 스마트폰이 중력을 따르거나 거울러 움직이는 가속 운동에 영향을 받을 수 있습니다.</li> </ul>
GyroscopeSensor	
위치 센서	
NFC	
방향 센서	
Pedometer	
근접 센서	
소셜	
저장소	

[자세히 알아보기](#)

### 앱인벤터 컴포넌트

#### • 가속도 센서

-흔들림을 감지하여 3차원 공간에서의 가속도 근사값을 측정하는 보이지 않는 컴포넌트이다.

## 앱인벤터 컴포넌트(미디어)



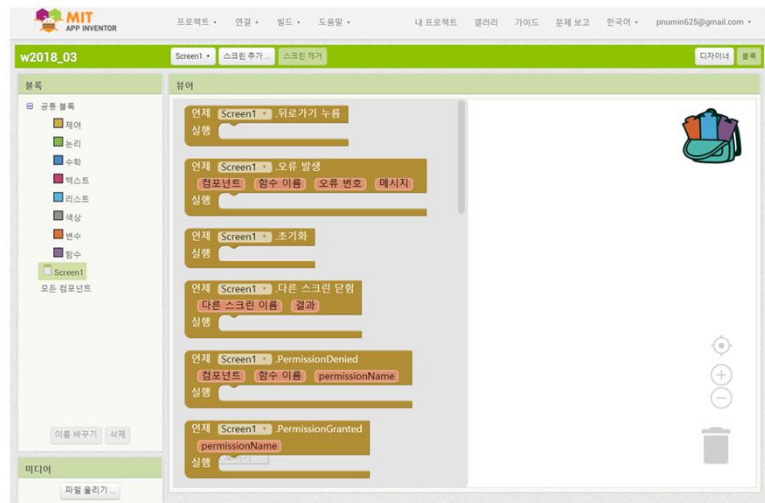
### 앱인벤터 컴포넌트

- 음성 변환

- 음성 변환 기능을 사용하여 글을 말로 바꾸어주는 컴포넌트이다.



# 앱인벤터 - 블록



블록영역에는 내장된 블록과 팔레트에서 추가한 컴포넌트인 블록들이 포함된다. 블록 뷰어는 블록영역에서 프로그래밍 할 수 있는 영역이다. 디자이너 화면에 배치된 컴포넌트들을 사용자가 원하는 동작을 하도록 구성하는 블록과 블록을 연결하여 작업을 한다. 왼쪽의 컴포넌트 리스트 중에서 하나의 컴포넌트를 클릭하면, 선택된 컴포넌트가 가질 수 있는 블록 리스트가 오른쪽에 나타난다. 스크린 영역의 컴포넌트가 가질 수 있는 블록의 종류는 이벤트 블록(노란색), 함수 블록(보라색), 속성 블록(초록색)으로 구분된다.

- 이벤트 블록(노란색)
  - 사용자가 특별한 행동을 했을 때의 반응을 설정하고자 할 때 사용
- 함수 블록(보라색)
  - 프로그램 내에서 이미 만들어 놓은 블록들의 조합
  - 공동블록의 함수 블록을 이용하여 사용자가 직접 함수를 만들 수도 있음
- 속성 블록(초록색)
  - 각 컴포넌트들의 색상, 크기, 위치, 속도 등의 속성 값을 변경하고자 할 때 사용
- 속성 블록(연한초록색)
  - 각 컴포넌트들의 색상, 크기, 위치, 속도 등의 속성 값을 가져올 때 사용

# 해결문제

본인을 소개하는 소개 앱을 만들어보세요.  
핸드폰을 흔들면 자료가 입력이 되고  
입력된 내용을 메시지 창을 이용해서 보여주고  
소개 내용을 음성으로 읽어주도록 작성



- 분해
  - 핸드폰을 흔든다.
  - 학생정보를 입력한다.
  - 입력된 정보를 음성으로 알려준다.
- 패턴인식
  - 핸드폰 흔들기 : 가속도 센서
  - 학생정보 입력 : 텍스트 상자
  - 입력된 정보 출력 : 버튼 -> 알림, 알림 내용 변수저장
  - 음성읽기 : 변수에 저장된 알림 내용
- 추상화
  - 디자이너 화면구성

## 해결문제 - 디자인

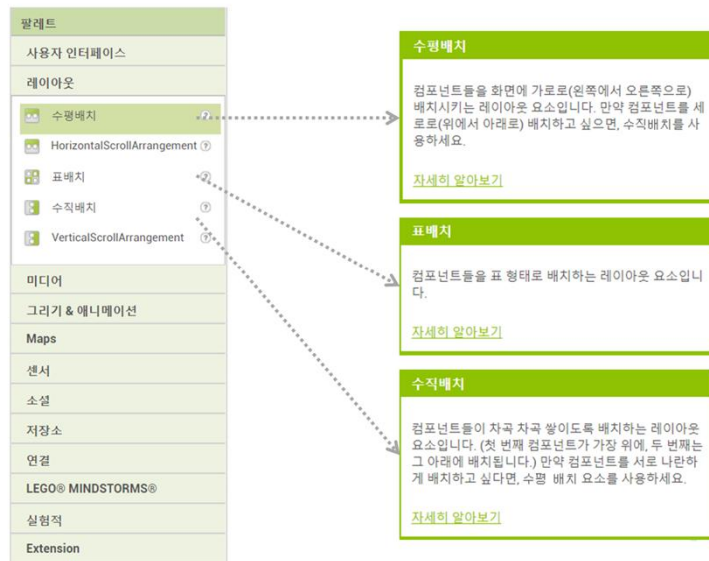


Come with  
**PNU!**

## 앱인벤터 – 순차(연산)



# 앱인벤터 컴포넌트 - 레이아웃

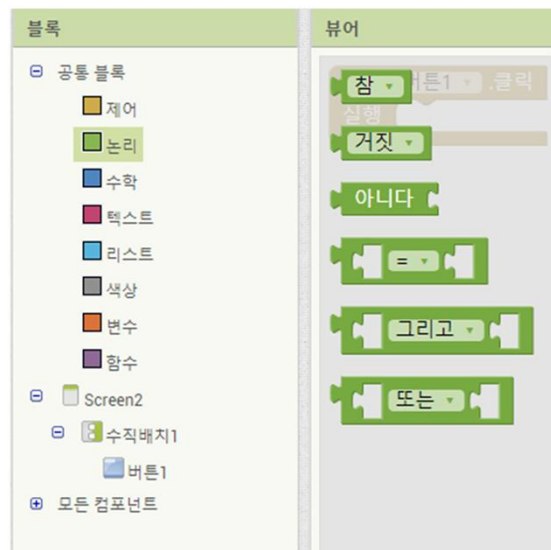


## 앱인벤터 컴포넌트

**레이아웃**은 컴포넌트들의 배열을 지정한다.

- 수평배치  
컴포넌트들을 화면에 가로로(왼쪽에서 오른쪽으로) 배치시키는 레이아웃 요소
- 표배치  
컴포넌트들을 표 형태로 배치하는 레이아웃 요소
- 수직배치  
컴포넌트들이 차곡 차곡 쌓이도록 배치하는 레이아웃 요소  
컴포넌트를 세로로(위에서 아래로) 배치

# 앱인벤터 논리블록



## 앱인벤터 공통 블록

:각 구성 요소에는 자체 이벤트, 메서드 및 속성에 고유 한 블록 집합이 있다.

## [논리 블록]

### 참, 거짓 블록

-상수 값 참(true), 거짓(false)를 나타냄

### 아니다 블록

-논리 부정을 실행 해, 입력이 true의 경우는 false를 돌려 주어, 입력이 false의 경우는 true를 돌려줌

### =, ≠ 블록

-인수가 동일한 지 여부를 테스트함

### 그리고 블록

-모든 논리 조건 집합이 참인지 여부를 테스트하여 테스트 된 모든 조건이 참인 경우에만 결과가 참(true)이 됨

### 또는 블록

-논리 조건 집합 중 하나라도 참인지 여부를 테스트하여 하나 이상의 테스트 된 조건이 참이면 결과가 참

# 앱인벤터 수학블록



Come with  
PNU!

## [수학 블록]

- 숫자
- $=$ ,  $\neq$ ,  $>$ ,  $\geq$ ,  $<$ ,  $\leq$
- 최대 최소
- sqrt, abs, -, log,  $e^x$ , round, ceiling, floor
- 계수의 나머지
- sin, cos, tan, asin, acos, atan
- 임의의 정수

# 앱인벤터 텍스트 블록



## [텍스트 블록]

### "" 블록

-텍스트 문자열 입력

### 합치기 블록

-입력된 문자열을 합쳐서 단일 문자열을 만듦

### 길이 블록

-문자열의 공백을 포함하여 문자 수를 반환

### 비어있나요? 블록

-문자열 길이가 0이면 참(true)을 반환하고 그렇지 않으면 거짓(false)을 반환

### 텍스트 비교하기 블록

-알파벳, 가나다 순으로 문자열 비교

### 다듬기 블록

-입력 문자열 앞이나 뒤에 공백을 제거하고 결과를 반환

### 대문자/소문자 블록

-입력 문자열을 대문자나 소문자로 변환

### 분할 텍스트 블록

-구분 텍스트로 문자열 나누기



# 앱인벤터 변수 블록



## [변수 블록]

### 전역변수 초기화 블록

- 전역 변수를 만드는 데 사용
- 전역 변수는 모든 프로시저 또는 이벤트에서 사용되므로 이 블록은 독립적임

### 가져오기 블록

- 변수 값을 가져옴

### 지정하기 블록

- 변수 값을 설정함

### 지역변수 초기화 블록

- 실행하는 프로시저에서만 사용되는 새 변수를 만들 수 있는 변경자

# 해결문제

표준체중을 계산하는 대표적인 방법으로 BMI법이 있습니다.

BMI는 신체질량지수로 신장과 체중의 비율을 계산한 객관적인 체중지수입니다.

BMI법에 따른 표준체중 공식은 다음과 같습니다.

표준체중 = 키(m) x 키(m) x BMI(여자 21, 남자 22)

비만을 진단하기 위한 BMI 지수 계산법은 다음과 같습니다.

$BMI = \text{체중(kg)} / (\text{키(m)} \times \text{키(m)})$

키와 몸무게를 입력 받아서 BMI법에 따른 표준체중과 BMI지수를 계산하는 앱을 만들어 보세요. 단, 흔들면 초기화되도록 하세요.

- 분해
  - 키와 몸무게 입력한다.
  - 여자와 남자 표준체중을 구한다.
  - BMI 지수를 구한다.
  - 초기화한다.
- 패턴인식
  - 핸드폰 흔들기 : 가속도 센서
  - 키와 몸무게 입력 : 텍스트 상자
  - 결과 출력 : 버튼 -> 레이블로 표시
- 추상화
  - 디자이너 화면구성

# 앱인벤터 디자인

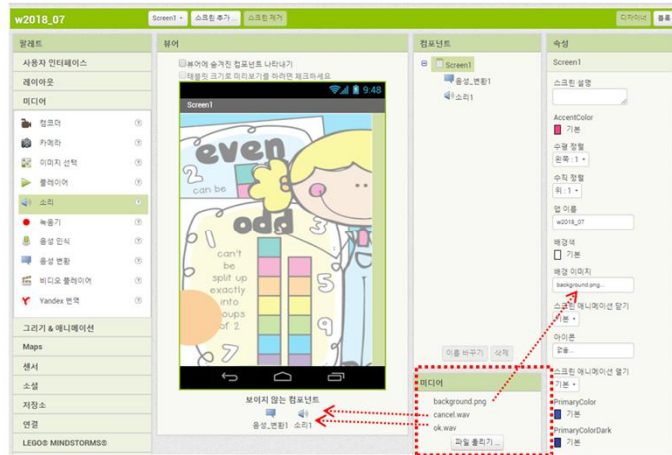


Come with  
PNU!

## 앱인벤터 – 순차(선택)



# 앱인벤터 미디어 사용하기



앱 제작에 필요한 사진, 음악, 영상들을 넣어 놓는 곳

## 앱인벤터 미디어 컴포넌트

### • 소리

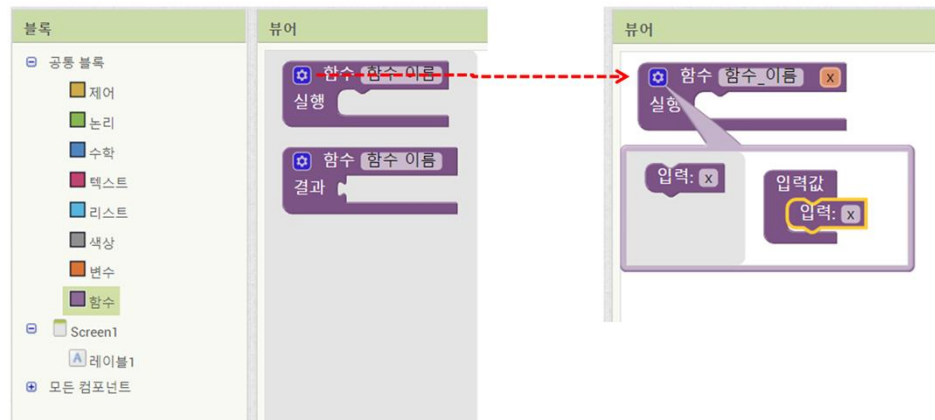
- 소리 파일을 재생하는 멀티미디어 컴포넌트
- 재생할 소리 파일의 이름은 디자이너나 블록 에디터에서 지정할 수 있음
- 효과음과 같은 짧은 소리 파일을 재생하는데 최적화
- ※ 플레이어 컴포넌트는 노래와 같은 길이가 긴 소리를 재생하는데 적합

# 앱인벤터 조건 만들기

**[제어 블록]**

- 22

# 앱인벤터 함수 블록



## 함수

:프로 시저는 특정 작업을 수행하는 지침 집합

### [함수블록]

- 함수이름을 입력하여 새로운 함수 생성
- 함수호출시 입력 값을 줄 수 있고 결과값을 반환 할 수 있음

# 해결문제

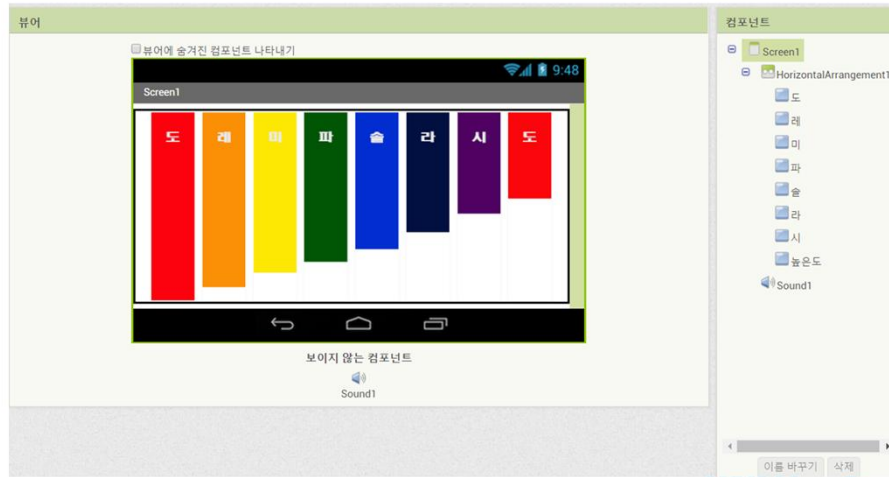
핸드폰으로 실로폰을 만들어 보세요.



- 분해
  - 각 음계에 해당하는 이미지를 누른다.
  - 해당 이미지에 맞는 음을 들려준다.
- 패턴인식
  - 이미지를 누른다. : 이미지 스프라이트
  - 이미지에 해당하는 음을 들려준다 : 함수 처리
- 추상화
  - 디자이너 화면구성



# 해결문제 - 디자인



# 해결문제

## 홀짝 게임앱을 만들어 보세요.

홀짝 버튼을 누르면 랜덤수를 자동으로 생성하고  
맞으면 ok.wav 틀리면 cancel.wav 소리가 나도록 하세요.



- 분해
  - 홀짝버튼을 선택하여 사용자입력을 정한다.
  - 컴퓨터가 랜덤수를 생성한다.
  - 생성된 랜덤수가 홀수인지 짝수인지 구분한다.
  - 결과를 화면에 표시하고 결과에 맞는 소리를 들려준다.
- 패턴인식
  - 핸드폰 흔들기 : 가속도 센서
  - 홀짝선택 : 버튼 -> 랜덤수 생성
  - 홀짝이 눌러졌을 때 처리할 일 -> 함수 처리
- 추상화
  - 디자이너 화면구성

# 해결문제 - 디자인



come with  
PNU!

## 해결문제 - 블록 (패턴인식)



## 앱인벤터 – 순차(반복)



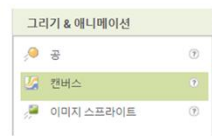
## 앱인벤터 반복문



# 해결문제 - 디자인



# 앱인벤터 컴포넌트 - 캔버스



**캔버스**

터치 가능한 2차원 패널로 그림을 그릴 수도 있고 스프라이트를 움직일 수도 있습니다.

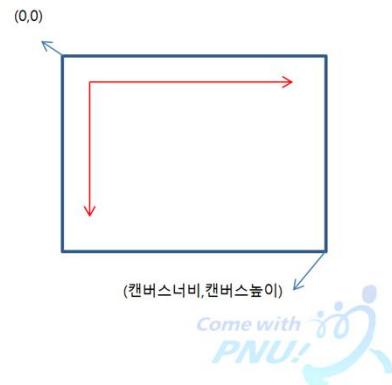
캔버스의 배경색, 페인트 색상, 배경 이미지, 너비, 높이를 디자이너 또는 블록 에디터에서 지정할 수 있습니다. 너비와 높이의 단위는 픽셀이며 반드시 0보다 커야 합니다.

캔버스에서의 위치는 (X, Y) 값의 쌍으로 표현되며,

- X는 캔버스의 왼쪽 모서리로부터 떨어진 픽셀 수
- Y는 캔버스의 위쪽 모서리로부터 떨어진 픽셀 수입니다

이벤트를 통해 캔버스의 어디가 터치되었는지, 또는 스프라이트 (이미지 스프라이트 또는 공)가 드래그 되었는지를 알 수 있습니다. 여러 함수들을 사용하면 점, 선, 원들을 그릴 수도 있습니다.

[자세히 알아보기](#)



## 앱인벤터 컴포넌트 캔버스

- 터치 가능한 2차원 패널로 그림을 그릴 수도 있고 스프라이트를 움직일 수도 있다.
- 캔버스의 배경색, 페인트 색상, 배경 이미지, 너비, 높이를 디자이너 또는 블록 에디터에서 지정할 수 있다.
  - 너비와 높이의 단위는 픽셀이며 반드시 0보다 커야 한다.
- 캔버스에서의 위치는 (X, Y) 값의 쌍으로 표현된다.
  - X는 캔버스의 왼쪽 모서리로부터 떨어진 픽셀 수
  - Y는 캔버스의 위쪽 모서리로부터 떨어진 픽셀 수
- 이벤트를 통해 캔버스의 어디가 터치되었는지, 또는 스프라이트 (이미지 스프라이트 또는 공)가 드래그 되었는지를 알 수 있다.
- 여러 함수들을 사용하면 점, 선, 원들을 그릴 수도 있다.



# 앱인벤터 이미지스프라이트

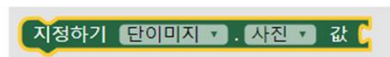
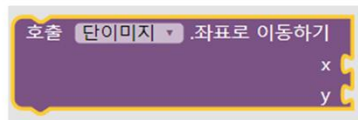


## 이미지 스프라이트

이미지 스프라이트는 캔버스에 놓을 수 있습니다. 터치나 드래그에 반응하며 다른 스프라이트 (공, 이미지 스프라이트) 나 캔버스의 모서리와 상호작용 할 수 있습니다. 또한 지정된 속성에 따라 움직일 수도 있습니다. 사진 속성에 지정된 값에 따라 이미지 모양이 달라집니다 (만약 표시 속성이 거짓이라면 보이지 않게 됩니다).

예를 들어 이미지 스프라이트를 매 1000밀리초 (1초) 마다 왼쪽으로 10픽셀씩 움직이게 하려면, 속도를 10 [픽셀]로, 간격을 1000 [밀리초]로, 방향을 180 [도], 회전 여부를 참으로 지정하면 됩니다. 회전 속성이 참이라면 스프라이트의 방향이 변하는데 따라 이미지가 회전하게 됩니다. 현재 회전된 스프라이트 중첩 검사는 회전되지 않은 상태의 위치를 사용합니다. 따라서 값이 낮거나 높고 넓은 스프라이트들의 위치값은 다소 부정확할 수도 있습니다. 스프라이트의 모든 속성들은 언제든지 프로그램 조작으로 변경 될 수 있습니다.

[자세히 알아보기](#)



## 앱인벤터 컴포넌트 이미지스프라이트

- 이미지 스프라이트는 캔버스에 놓을 수 있다.
- 터치나 드래그에 반응하며 다른 스프라이트 (공, 이미지 스프라이트)나 캔버스의 모서리와 상호작용 할 수 있다.
- 지정된 속성에 따라 움직일 수도 있다.
- 사진 속성에 지정된 값에 따라 이미지 모양이 달라진다.  
-(만약 표시 속성이 거짓이라면 보이지 않게 됨)

# 앱인벤터 타이머



## 앱인벤터 컴포넌트

### 시계

- 스마트폰의 시계, 타이머, 그리고 시간 계산을 기능을 하는 제공하는 보이지 않는 컴포넌트
- **타이머 항상 작동** 이 체크 되었으면 해당 스크린이 열리자마자 시계가 작동
- **활성여부**는 활성화 설정
- **타이머 간격**은 타이머가 한 작업을 실행 하고 다음 작업을 하기까지 지연하는 값  
단위는 ms(밀리세컨드/밀리초)를 사용  
1000ms = 1s(초)

# 앱인벤터 음성인식



## 앱인벤터 컴포넌트 음성 인식

- 음성 인식 기능을 입력된 말을 글로 변환해주는 컴포넌트

# 해결문제

## 구구단 출력 앱을 만들어 보세요.

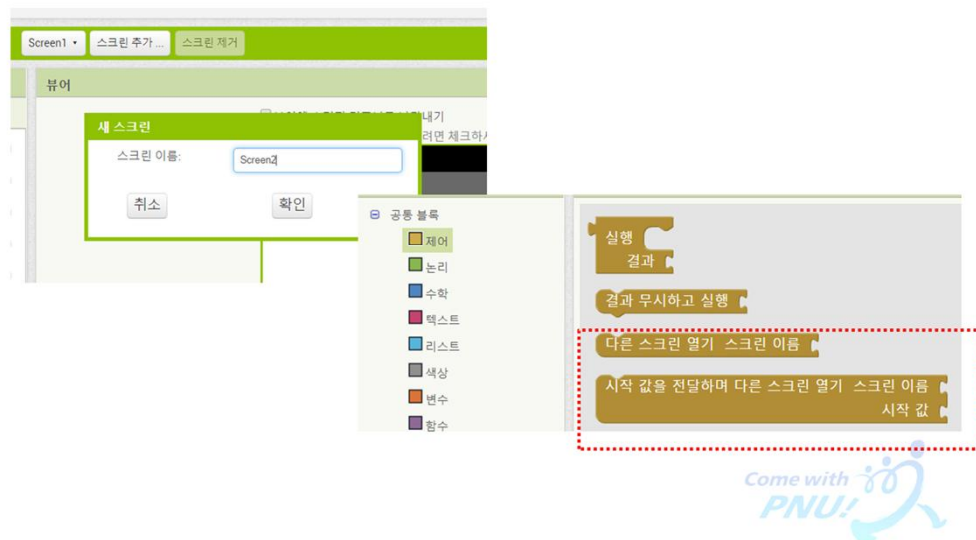
시작버튼을 누르면 숫자그림이 랜덤으로 나타나고 해당 숫자그림을 선택하면 구구단을 출력

음성인식버튼을 누르면 숫자를 입력하고 숫자가 입력되면 해당 구구단을 출력하고 숫자가 아닌 경우 숫자입력 안내

구구단읽기 버튼을 누르면 구구단이 출력된 경우 구구단을 음성으로 읽기



## 앱인벤터 화면전환



안드로이드에서는 액티비티(Activity)라고 하는 것이 앱인벤터에서는 스크린(Screen)이라고 한다.  
앱인벤터의 스크린은 앱에 표시되는 화면으로 스크린간의 이동은 앱에서 화면 전환이 됩니다.

1. 스크린 추가하기
2. 추가 한 스크린 디자인 하기
3. 스크린 이동 블록 작업하기  
제어에서 다른 스크린 열기 블록을 추가

# 해결문제

구구단 학습 앱을 만들어 보세요.

랜덤으로 구구단을 생성하여  
사용자가 답을 입력하여  
정답인지 확인할 수 있도록 한다.  
몇번 정답을 맞추었는지 표시한다.



# 해결문제

## 바나나 잡기 게임 앱을 만들어 보세요.

왼쪽/오른쪽 버튼으로 원숭이를 움직임  
바나나와 폭탄은 랜덤으로 아래로 내려오기  
바나나와 폭탄이 아래에 닿으면 다시 위에서 나타나기  
이때, 그림이 바나나와 폭탄이 랜덤으로 나타나기  
원숭이와 부딪힌 그림이 바나나이면 점수가 증가하고 폭탄이면 점수가 감소



# 바나나 게임





## 바나나 / 폭탄 랜덤 움직이기



Come with  
**PNU!**

## 원숭이 움직이기



전역변수 초기화 점수 값 0

전역변수 초기화 점수 값 0

# 리스트



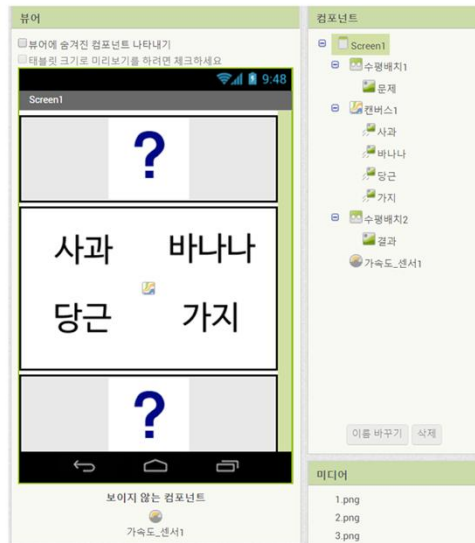
# 해결문제

## 과일 이름을 맞추는 학습 앱을 만들어 보세요.

핸드폰을 흔들면 그림이 랜덤으로 사진이 나옴  
단어이미지(a?.png)를 선택하면 선택된 단어이미지의 배경이 빨간  
색(b?.png)으로 변경  
단어이미지가 한번 선택되면 다른 단어이미지는 선택되지 않도록  
그림과 단어 이미지가 같으면 결과에 O, 그렇지 않으면 X이미지가  
나오도록 함



# 과일 이름 맞추기



## 문제분석

- 핸드폰을 흔들면 그림이 랜덤으로 사진이 나오
- 단어이미지(a?.png)를 선택하면 선택된 단어 이미지의 배경이 빨간색(b?.png)으로 변경
- 단어이미지가 한번 선택되면 다른 단어이미지는 선택되지 않도록
- 그림과 단어 이미지가 같으면 결과에 O, 그렇지 않으면 X이미지가 나오도록 함

# 리스트 선언

전역변수 초기화 문제과일 값 빈 리스트 만들기

전역변수 초기화 답안과일 값 빈 리스트 만들기

언제 Screen1 초기화

실행

- 리스트에 항목 추가하기 리스트 가져오기 global 문제과일
  - item "1.png"
  - item "2.png"
  - item "3.png"
  - item "4.png"
- 리스트에 항목 추가하기 리스트 가져오기 global 답안과일
  - item 사과
  - item 바나나
  - item 당근
  - item 가지



# 과일이미지 나오기





# 결과만들기

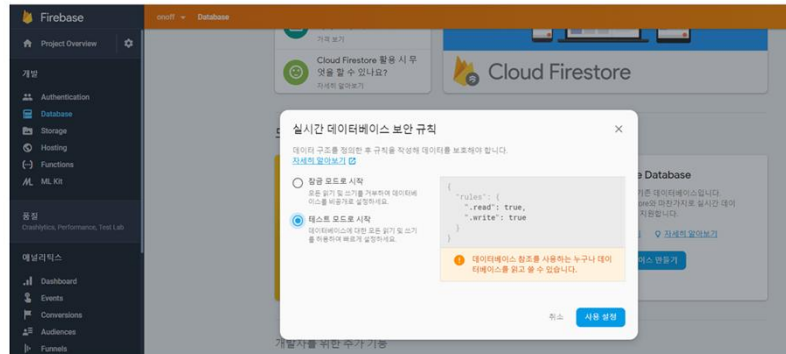


# 파이어베이스 연결하기



# 파이어베이스 설정

- 파이어베이스 콘솔로 이동하여 프로젝트 추가
  - <https://firebase.google.com>
- 대시보드에서 Realtime Database 선택
  - 데이터베이스 만들기
  - 테스트모드로 시작

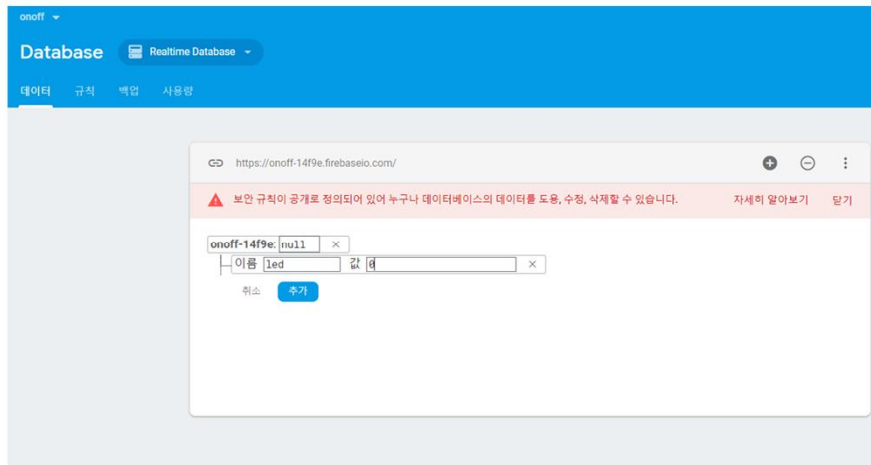


## 파이어베이스

파이어베이스(Firebase)는 2011년 파이어베이스(Firebase, Inc)사가 개발하고 2014년 구글에 인수된 iOS, Android 및 웹 앱을 개발할 수 있도록 구글이 지원하는 개발 플랫폼입니다.

파이어베이스는 실시간 데이터 동기화를 지원하며 클라우드 기반 데이터베이스 솔루션을 제공합니다. 실시간 데이터베이스는 파이어베이스의 고유한 데이터베이스로서, 여러 클라이언트에서 실시간으로 상태를 동기화해야 하는 모바일 앱을 위한 효율적이고 지연 시간이 짧은 솔루션입니다.

# 파이어베이스 자료 추가

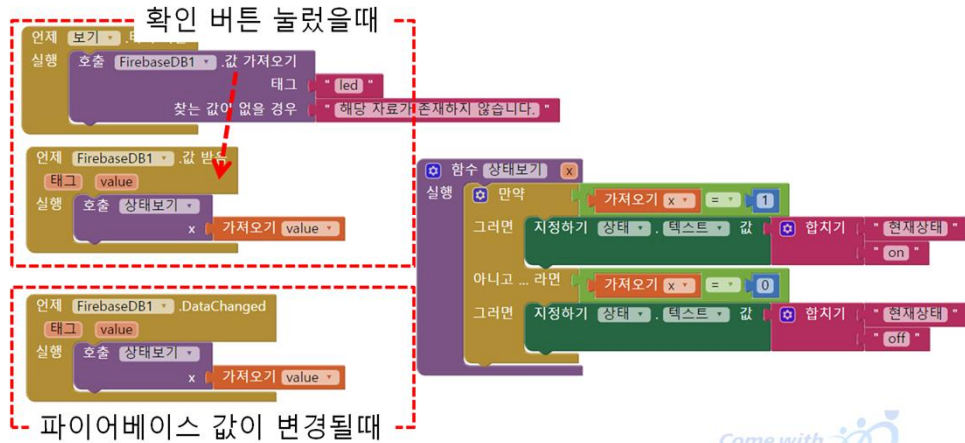


Come with 80  
PNU!

# 앱인벤터에서 파이어베이스 연동



## 앱인벤터 파이어베이스 값가져오기



## 앱인벤터 파이어베이스 값변경

