

App Inventor for
Android



App Inventor for Android

google が公開しているAndroidアプリの開発環境「App Inventor for Android」を使ってプログラミング入門をしてみよう。

まずは、入手とインストールです。

[App Inventor のサイト](#) を確認して、インストールします。

現時点では、英語だけの解説ですが、頑張って作業しましょう。

Mac Linux Windows それぞれの環境に合わせた解説があります。自分の環境を確認して作業します。

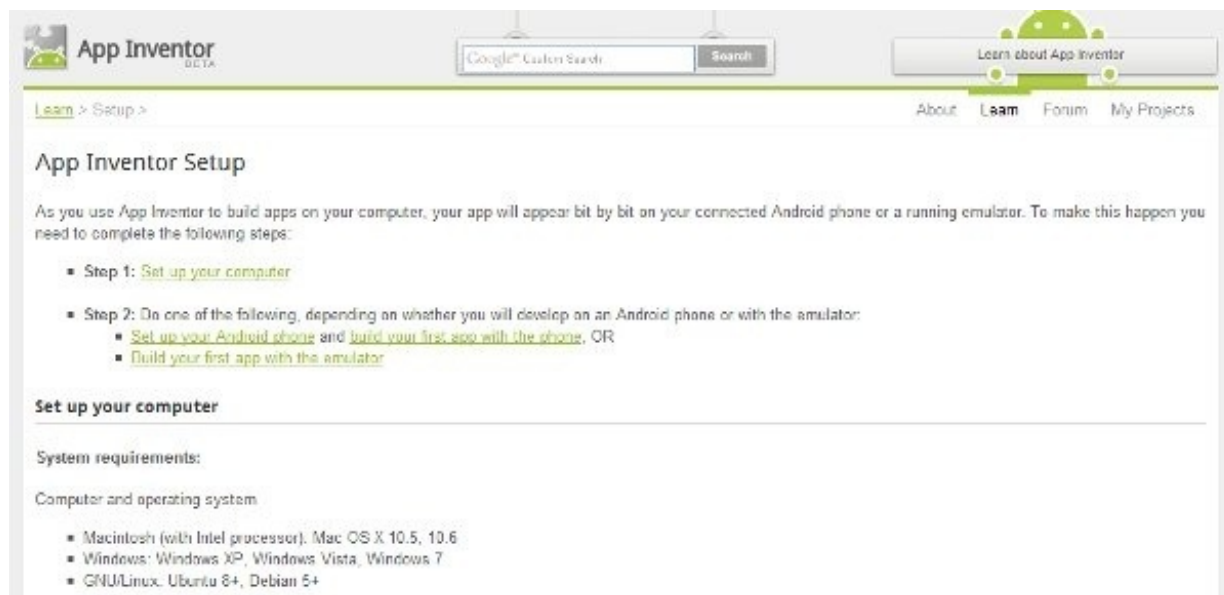
App Inventor は、各OS上で、最新のJava環境、ブラウザを利用して動作します。

App Inventor for Android をインストール Setup 1

App Inventor for Android のインストールについて説明します。

[英文公式ページ](#)の正確な日本語訳ではないのでご了承ください。

公式ページのGetStartedをクリックすると以下のページが表示されます。



インストール作業は、ここの指示に従って行います。

まず、Step 1です。

ここでは、使用しているコンピュータシステムについての、環境を確認します。

OSは、

- Macintosh (with Intel processor): Mac OS X 10.5, 10.6
- Windows: Windows XP, Windows Vista, Windows 7
- GNU/Linux: Ubuntu 8+, Debian 5+

に対応しています。

自分で使用するコンピュータのOSは、どれか確認してください。OSにあわせたインストールファイルをダウンロードします。

ブラウザ（インターネット閲覧用ソフトウェア）は、

- Mozilla Firefox 3.6 or higher
- Apple Safari 5.0 or higher
- Google Chrome 4.0 or higher
- Microsoft Internet Explorer 7 or higher

に対応します。使用しているブラウザの名称を確認します。一般的には、ブラウザのヘルプメニューから確認することができます。

つぎに、Java環境について確認します。App Inventor for Android の実行には、Java 6 環境が必要です。

Java は、www.java.com からダウンロードできます。

Java環境のテストも同じサイトで可能です。

あなたと JAVA,
今すぐダウンロード

無料 Java のダウンロード

開発者向け:
[Java 研修トレーニング](#) / [Java 認定資格](#)

» [Javaとは?](#) » [Javaの有無のチェック](#) » [サポート情報](#)

Java in Action (英語サイト)

Java + Alice Java + Livescribe Java + Neil Young Java + Blu-ray Disc Java + Mifos Java + Amazon Kindle

言語の選択 | [Javaについて](#)
[プライバシー](#) | [利用規定](#) | [お問い合わせ](#) | [免責事項](#)

ORACLE

いきなり「無償Javaのダウンロード」ではなく、その下に小さくある [Javaの有無のチェック](#) で状況を確認することができるので、こちらを選択します。

これをクリックすると以下のように切り替わります。

Java のバージョンの確認

お使いのオペレーティングシステムに推奨されたバージョンの Java をインストールしていることを確認してください。

Java のバージョンの確認

[サポート情報](#)

注: 新たに Java ソフトウェアをインストールした場合は、インストール状況を確認する前にブラウザを再起動する必要があります。ブラウザのすべてのウィンドウをいったん閉じてから、再開裂してください。

ORACLE

中央の「Javaのバージョンの確認」をクリックします。以下のように表示されれば、最新のJava環境がインストールされています。



[Java in Action](#) [ダウンロード](#) [ヘルプセンター](#)

Java のダウンロード一覧

ほかのコンピュータまたはオペレーティングシステム用の Java をダウンロードする場合は、下のリンクをクリックしてください。

[Java のダウンロード一覧](#)

ヘルプ資料

- » [Java とは?](#)
- » [エラーメッセージ](#)
- » [旧バージョンの削除](#)
- » [その他のヘルプ](#)

Java のバージョンを確認しました。

正常な設定です。

推奨バージョンの Java がインストールされています (Version 6 Update 25).

[言語の選択](#) | [Java について](#) | [お問い合わせ](#)
[プライバシー](#) | [利用規定](#) | [お問い合わせ](#) | [免責事項](#)

ORACLE

この表示にならないときは、各OSについてのインストール方法などを確認してJavaのインストール作業を行ってください。この作業が終了するまでは、App Inventor for Android のインストール作業に進まないでください。

App Inventor for Android をインストール Setup 2

Java環境が最新のものと同確認できたら、App Inventor for Android をインストールします。

自分の使用環境にあわせたインストールガイドを選んで作業します。ここでは、Windowsを例に説明します。

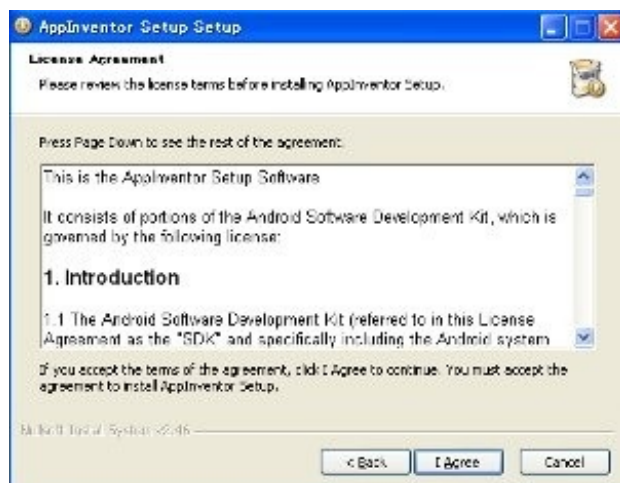
以下のように表示されたサイトの「download」をクリックするとファイルのダウンロードが開始されるので、ファイルを自分でファイルを確認しやすいフォルダに保存します。



ダウンロードが完了したら、インストール作業の権限のあるユーザで実行して、インストール作業を行います。起動すると以下のような表示になります。「Next」をクリックして次に進みます



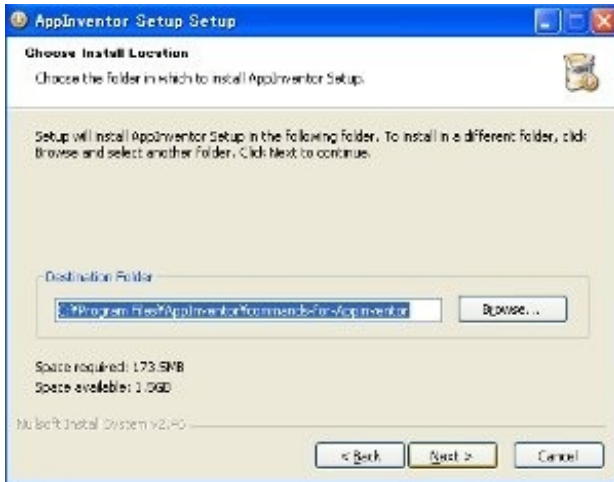
ライセンスに関する確認です。「I Agree」をクリックすると先に進めます。気になる方は、熟読してください。次に進みます。



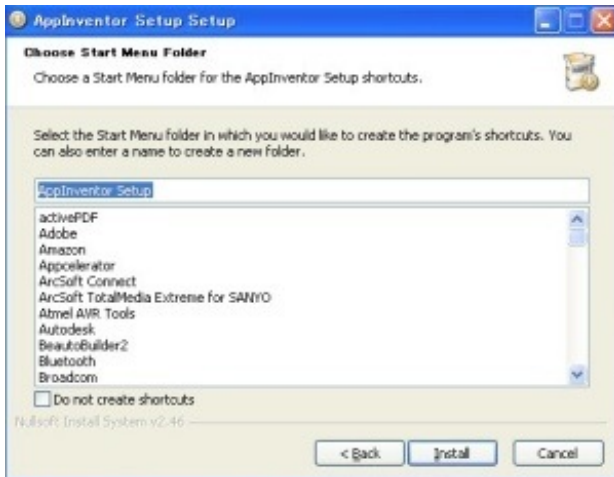
インストール先の確認です。

インストール先は、通常 `C:\Program Files\Appinventor\commands-for-Appinventor` になります。

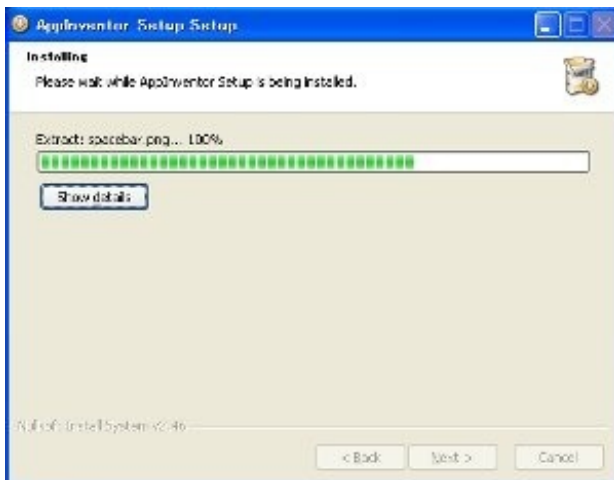
変更したいときは、ここで指定します。



次の画面は、Start Menu への登録に関するものです。このままで、問題ないと思いますが、自由に変更することができます。



ここで「Install」をクリックするとインストールが開始されます。もし、変更したいことがあれば、「<Back」で戻って再設定してください。



インストールが終了するまで待ちましょう。



お疲れさまでした。これで作業終了です。

実機を接続する際は、デバイスドライバファイルをインストールします。

- T-Mobile G1* / ADP1
- T-Mobile myTouch 3G* / Google Ion / ADP2
- Verizon Droid (not Droid X)
- Nexus One
- Nexus S

は、最初からインストールされています。

リストに無い場合は、[obtain and install a driver manually](#)の手順に従ってインストールします。

App Inventor for Android を起動

App Inventor for Android の起動は、ブラウザから行います。
ブラウザを起動して、[App Inventor for Android のサイト](#)を開きます。
以下のような表示が出たときは、googleアカウントでSign in します。



The easiest way to create apps for your phone!

App Inventor is built on the idea that you do not need to be a developer to build great mobile applications. Instead of code, App Inventor allows you to visually design applications and use blocks to specify application logic.



Use the GPS-location sensor to build location aware apps
For example, build an app to help you remember where you parked your car.



Make your apps communicate by using the phone functionality
For example, build an app that periodically texts "missing you" to your loved ones.



Integrate with the web to build mashup applications
For example, build an app that talks to your favorite website like Twitter.

Sign in with your
Google Account

Email:

Password:

Stay signed in

[Can't access your account?](#)

Don't have a Google Account?
[Create an account now](#)

©2011 Google - [About](#) - [Privacy](#) - [Terms](#)

Sign in すると以下のように開発環境が表示されます。

これで、インストールは終了です。App Inventor for Androidの世界を楽しみましょう！！

はじめてのプログラム

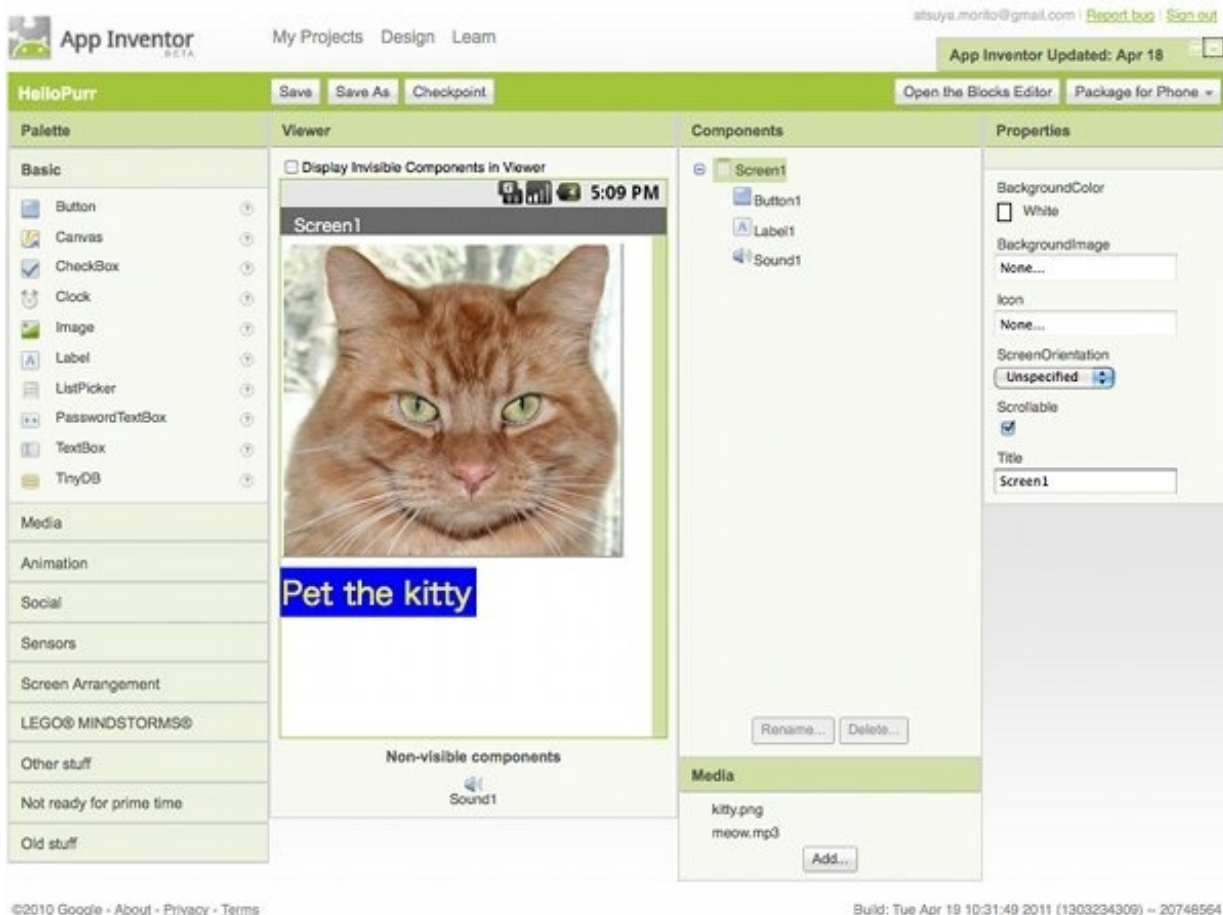
C言語や、Java言語でも、はじめてのプログラムは、Hello World! でした。

App Inventor は、違うようです。

チュートリアルで最初に作るのは、猫の写真タッチすると鳴き声があるというものです。

App Inventor の画面構成は、以下のようになっています。

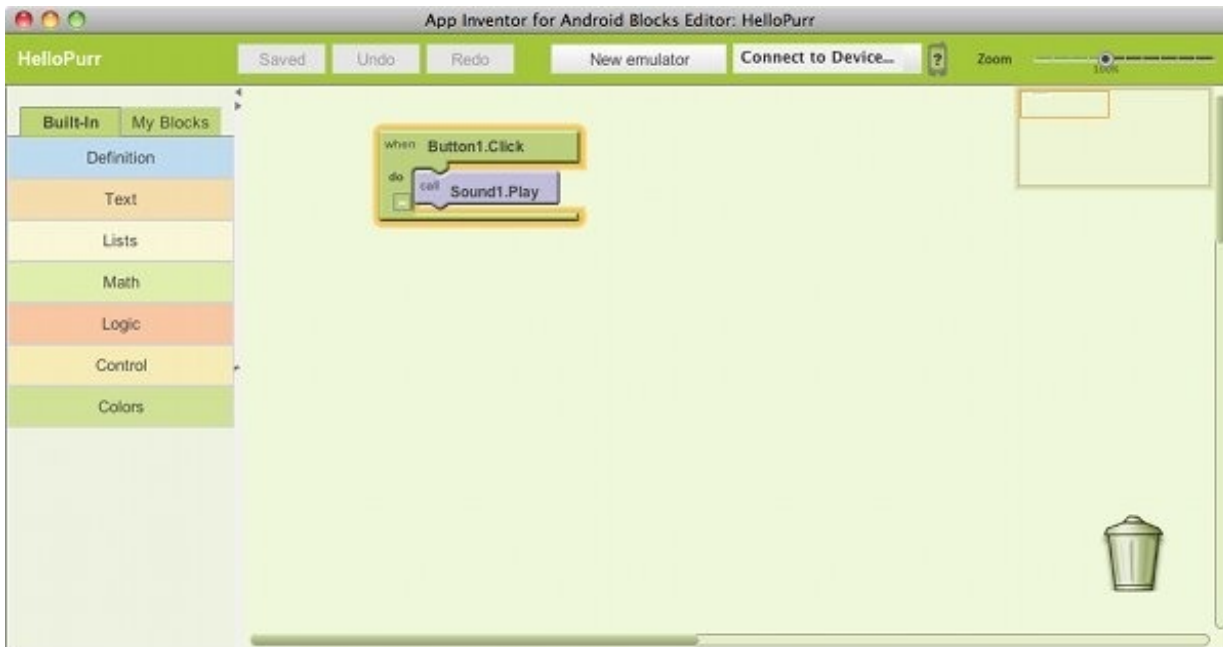
中央のViewerが実行時の画面です。ここに、左側のPaletteから、必要なパーツを並べてアプリの動作イメージを作ります。



配置したパーツは、必要に応じて、Components Properties などを変更します。

次にボランをクリックしたなどの、操作に対応する処理を作成します。

上部にあるOpen the Blocks Editor ボタンをクリックして、Block Editor を起動します。



左側の部分（名前があるのかな？）から、動きを表すブロックを選んで、配置することで処理を表現します。この例では、

When Button1.Click（Button1 を Click したとき）

Sound1 Play を call（呼び出して）do（実行する）

という処理になります。

中学生の英語レベルでなんとなく分かるような気がします。

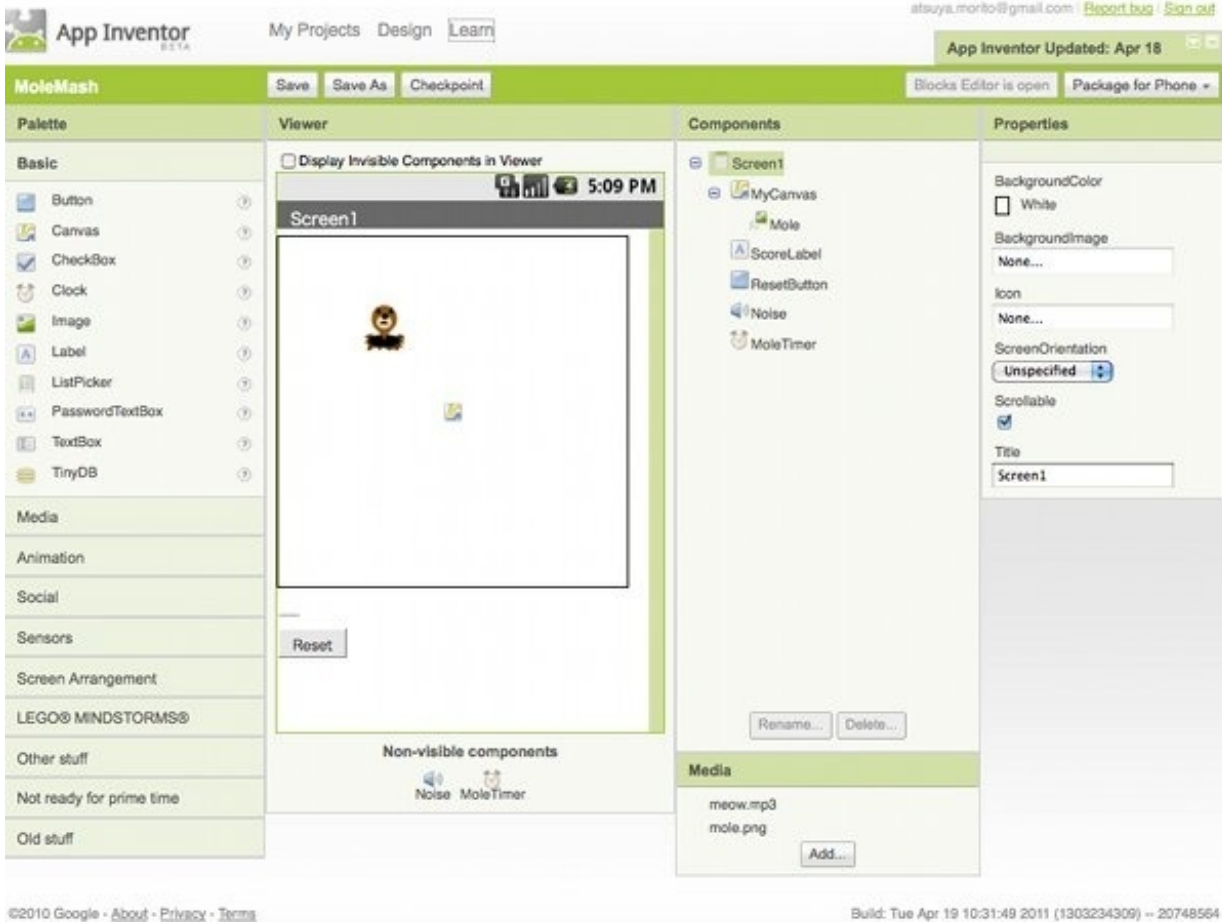
プログラムと思わないで、文章を作るような感じで考えた方が良いでしょう。

処理を考える

チュートリアルの中にある [Mole Mash](#) を例にして、処理の様子を考えてみます。

英文で、それまでのチュートリアルからみると、多少説明が簡略化されているので、はじめて操作すると、うまく動作できなくて、諦めなくなっちゃう気がします。

この文章を書きしてみようと思ったきっかけは、このチュートリアルがあったからかも知れません



チュートリアルにしたがって、このようにパーツを配置して各種設定を行います。必要な画像と音源は、Web からダウンロードして用意します。配置したパーツは、指示されたとおりに名前も変えます。名前を意味のある分かりやすいものにすることで、Blocks Editor での作業も分かりやすいものになります。Timer を配置して、定期的になにか処理を行えるようになっています。

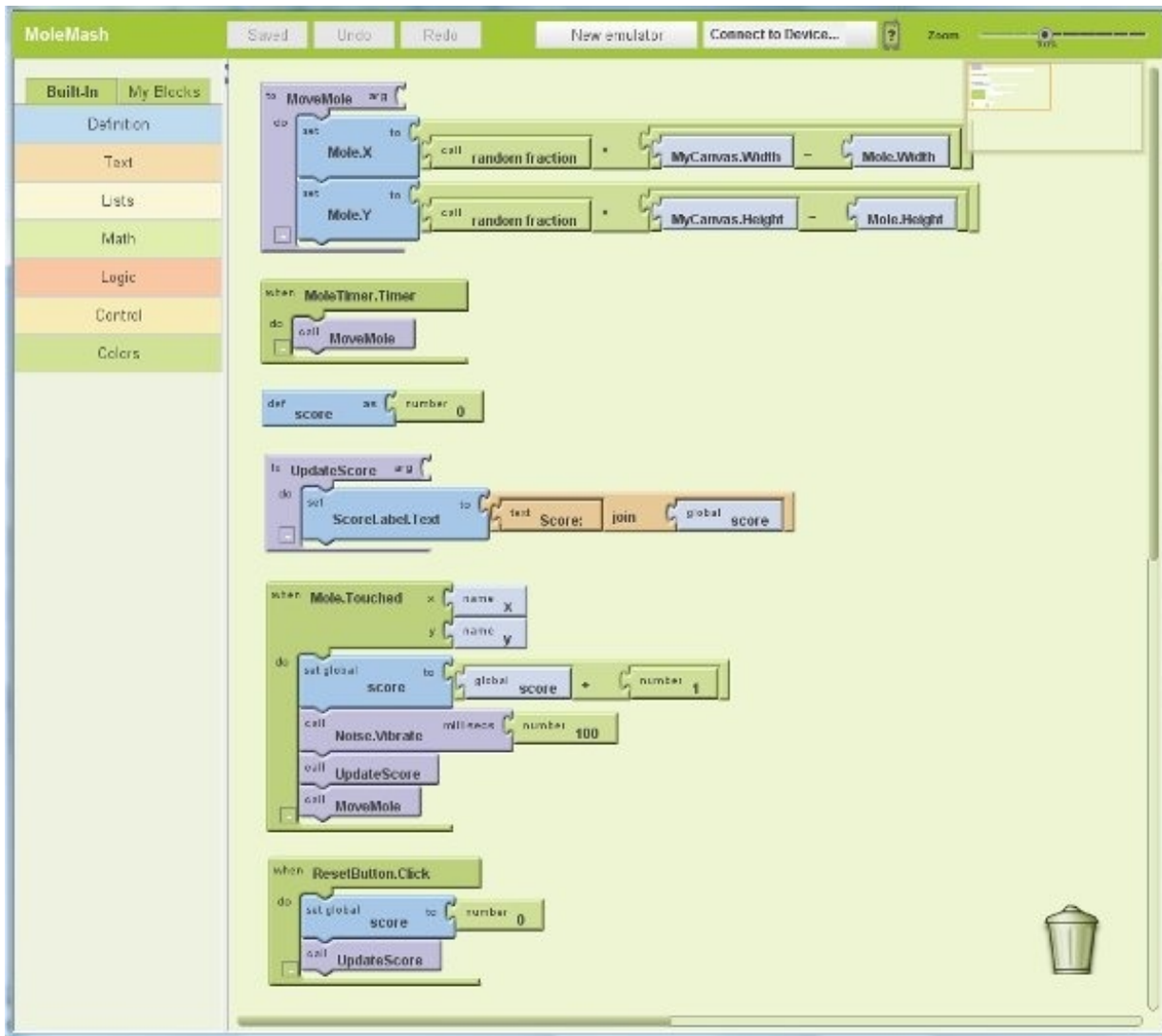
つぎは、Blocks Editor で処理を表現します。

まず、Blocks Editorの起動です。

右上にある、Open the Blocks Editor をクリックすると、Javaアプリケーションがダウンロード開始されます。環境によっては、ダウンロードの確認、起動してよいかなどのメッセージが出ます

自分で使いたくてダウンロードしているので、迷わず起動します。

しばらくたつと起動します。



今までは、ブラウザでの作業でしたが、ここからは、別ウィンドウで起動されたアプリケーションでの作業です。

上の画像は、MoleMushを完成したときのものです。これだけで、Androidアプリとして動作します。

[チュートリアル \(英語\)](#) にしたがって作業をすすめて完成できるのは、英語が得意な人と、プログラム作成経験者かな？と思います。

最近、プログラマは、理系の職業では、無くなってきていると聞いたことがあります。この環境では、とくにそう感じます。デザイナーでのデザイン、Blocks Editorでの作業、どちらもとくに理系の昔ながらのプログラミング作業には、程遠い感じです。

さて、Blocks Editorで作成する処理について動作を考えてみます。

このチュートリアルで作っているのは、MoleMash、もぐらたたき、ですね。ゲームなので、アプリを起動すると自動的にモグラが動き回って、うまくモグラをたたけたら（エミュレータでは、マウスでクリック）、音がしてスコアが更新される。という処理です。

自動的に動くという処理では、タイマを使います。デザイナーで、パーツを配置したときに、MoleTimerを作りました。ここで、一定時間ごとにタイマが動くように設定してあります。

左側にあるMy Blocksタブをクリックすると、デザインとして配置したパーツに関するブロックが表示されます。



このように表示されたブロックから、使いたいものを選んで配置していくことで、処理を作ります。

自動的にモグラが動き回って
という処理は、

```
when MoleTimer.Timer  
do [モグラを動かす]
```

というブロックになります。

じゃあ、[モグラを動かす] ってどうすんの??ということになるので、

Built-InタグのDefinitionをクリックして、ひとまとまりの関数として定義します。

これが、

```
to MoveMole arg
```

```
do set Mole.x to [乱数で適当な位置を作る]
```

```
  set Mole.y to [乱数で適当な位置を作る]
```

というブロックで定義されています。

[乱数で適当な位置を作る] というブロックは、Mathブロックから選んで作ります。

モグラを叩いたときの処理なども同様に考えられます。

計算してみる

日本のプログラム入門だと、計算して結果を表示するというのが、定番な気がするので、App Inventor for Android でもやってみます。

某IP***アプリで一時話題になっていたBMIを求めてみましょう。

アプリの動作は、身長と体重を入力してから、〔BMI計算〕と書いてあるボタンを押すと計算結果が表示される。

という感じでしょうか？

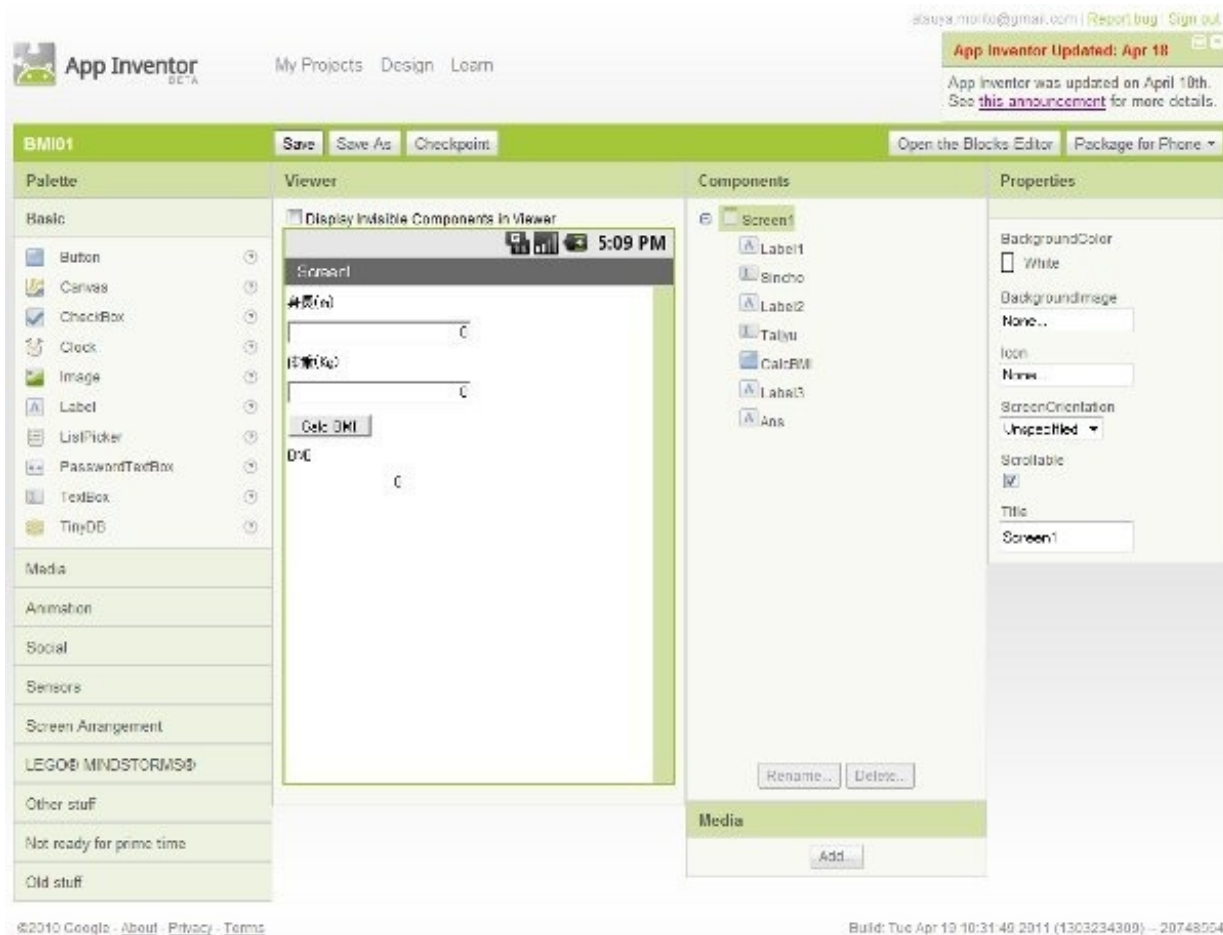
さて、では、新規作成です。

デザインとして配置するのは、テキストボックスとボタンですね。

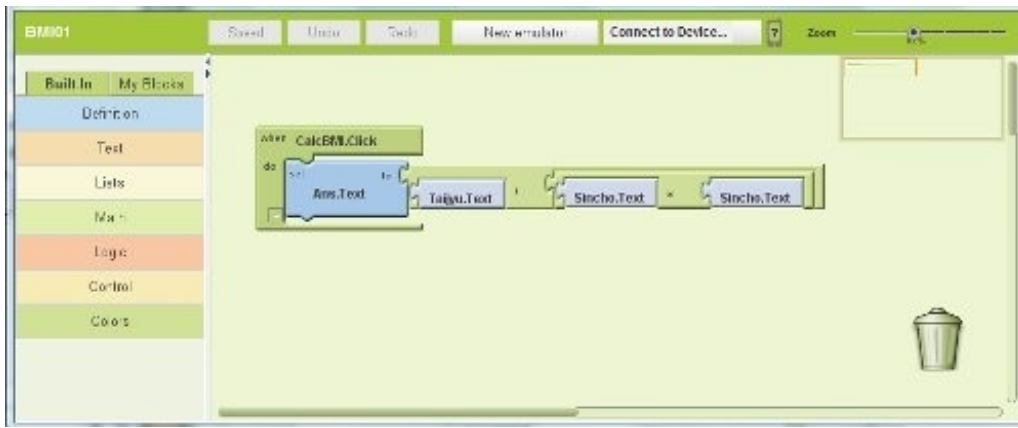
$BMI = \text{体重(Kg)} \div (\text{身長} \times \text{身長})$

なので、テキストボックス 2つ を入力に

それぞれの項目名などのラベルとともに配置します。



名前は、あとでわかりやすいようにつけます。テキストボックスのプロパティでは、NumbersOnly にチェックをつけておくと、入力ミスが軽減されます。



ボタンを押したら、ラベルにBMIを計算して表示する処理を記述して完成です。

when CalcBMI.click

do set ANS,Text to Taijuu.Text / Shincho.txt × Shincho.txt

のような処理となります。

実行して動作を確認しましょう。

らくちんです。今までの経験で良くあったのは、テキストボックスに入力した数値で計算しようとすると、数字（文字）なので数値化してから計算しようという、初心者には???な手間でした。

App Inventor for Androidでは、普通にそのまま計算できてしまいます。

計算式の作り方も（）でくるのではなく、2つの値を計算して行って全体にするという感じなので、慣れれば分かりやすいし、式の間違いも少ないかも知れません。