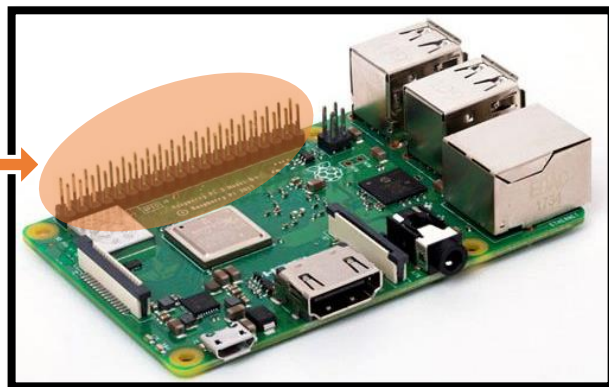


Raspberry Pi とは

イギリスのラズベリーパイ財団によって開発されている教育用ワンボードコンピュータ。

一昔前のパソコンと同等のスペックを持っており、上部にある **GPIO** と呼ばれるインターフェースを搭載しているのが特徴的。**GPIO ピン** から電子部品を繋いで制御する事ができ、電子工作によく用いられる。この科目では RaspberryPi 3B+ にラズパイ入門ボードを接続し、プログラムで電子部品の制御を行っていく。



Raspberry Pi OS の書き込み

パソコンを操作するには Windows や mac OS という OS（オペレーティングシステム）が必要なのと同様に、Raspberry Pi を操作するにも Raspberry Pi OS（旧 Raspbian）という OS が必要です。

Raspberry Pi OS は Linux をベースに作られた OS で、これを MicroSD カードに書き込んで本体に差し込むことにより、起動時に読み込まれて Raspberry Pi が操作できます。

下記の手順に従って自分の MicroSD カードに Raspberry Pi OS を書き込みましょう。

- ① ノートパソコンを起動し、配布された MicroSD カードを差し込む
- ② 実習機の裏側から LAN ケーブルを抜き、ノートパソコンに接続する。※授業終了時に元に戻すこと
- ③ ブラウザを起動し、[[Raspberry Pi OS – Raspberry Pi](#)]へアクセスする。
- ④ [Download for Windows]をクリックして Raspberry Pi Imager をダウンロードする。

Install Raspberry Pi OS using Raspberry Pi Imager

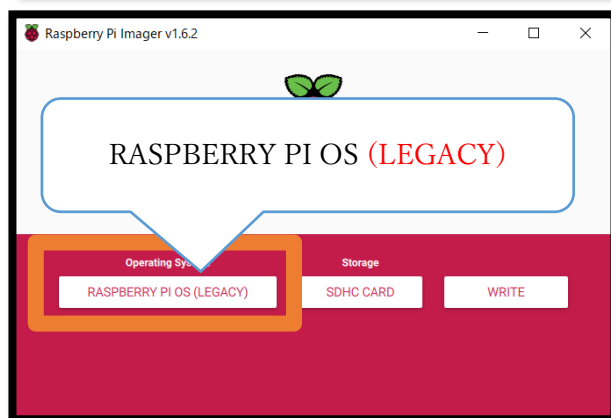
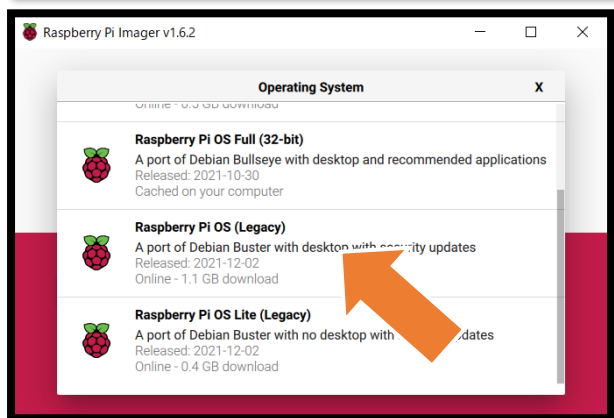
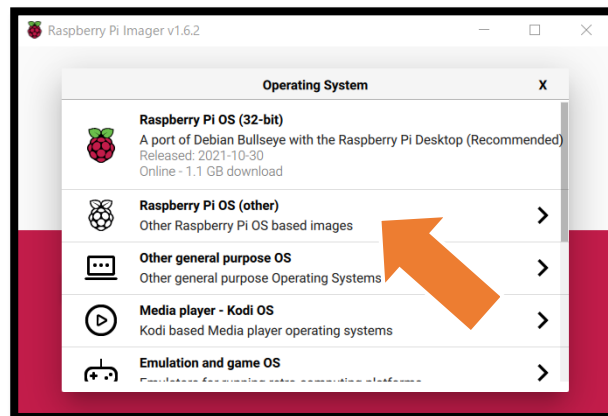
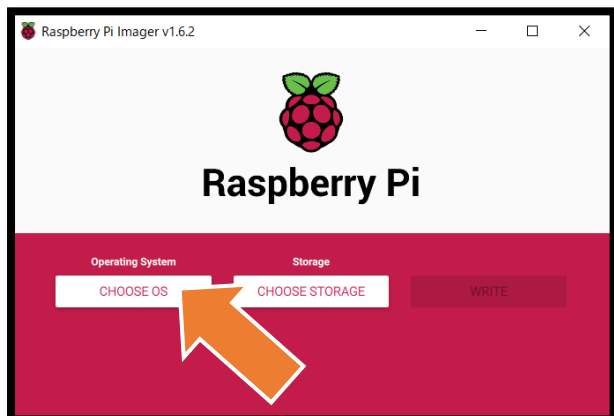
Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi. [Watch our 45-second video](#) to learn how to install an operating system using Raspberry Pi Imager.

Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

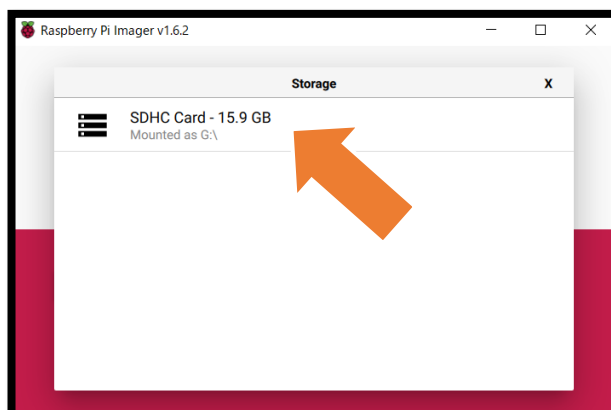
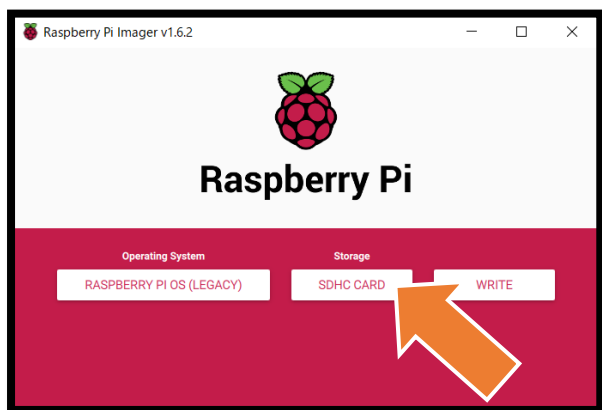
[Download for Windows](#)
[Download for macOS](#)
[Download for Ubuntu for x86](#)

- ⑤ ダウンロードした [imager_~~~.exe] を実行してインストールする。

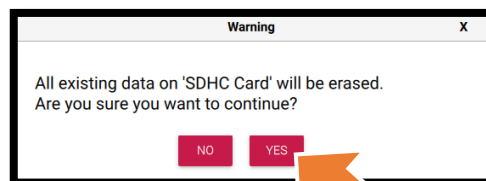
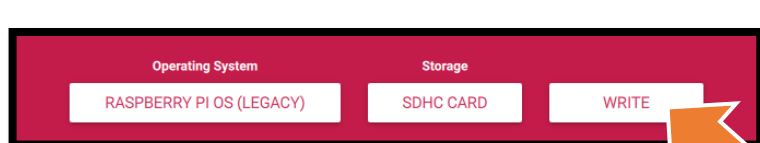
- ⑥ インストールした Raspberry Pi Imager を起動し、[CHOOSE OS]をクリックする。
その後、Operating System から [Raspberry Pi OS(other)] をクリックする。
その後、[Raspberry Pi OS (Legacy)] をクリックする。
Operating System が [RASPBERRY PI OS (LEGACY)] になっているか必ず確認すること。



- ⑦ [CHOOSE STORAGE] をクリックする。
その後、[SDHC Card – 15.9GB] をクリックする。※SD カードによって表記が違う場合あり。



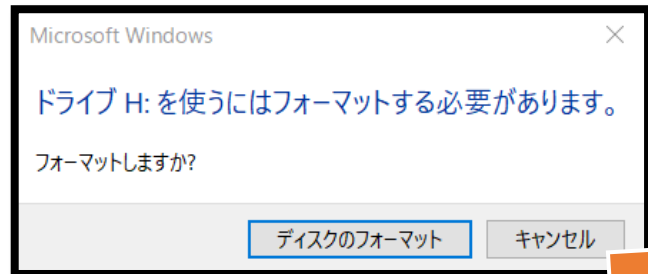
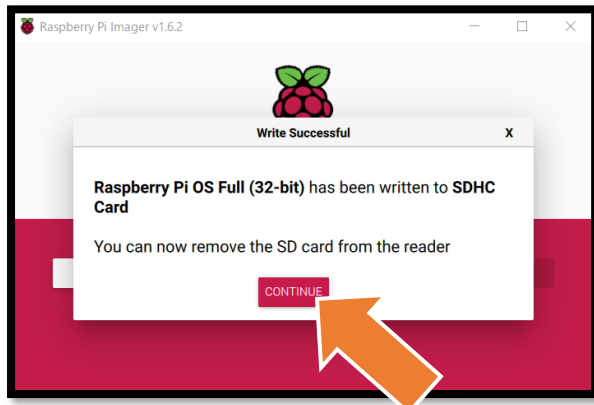
- ⑧ [WRITE] をクリックし、[YES] をクリックすると書き込みが始まるのでしばらく待機する。



- ⑨ 書き込みが完了すると成功とのウィンドウが表示されるので[CONTINUE]をクリックする。

その後、Raspberry Pi imager を閉じる。

※SD カードが読み込みなおされて Windows システムがフォーマットを要求してくる場合があるが、[キャンセル]をクリックして**フォーマットしないこと**。するともう一度やり直しになります。



Raspberry Pi 起動

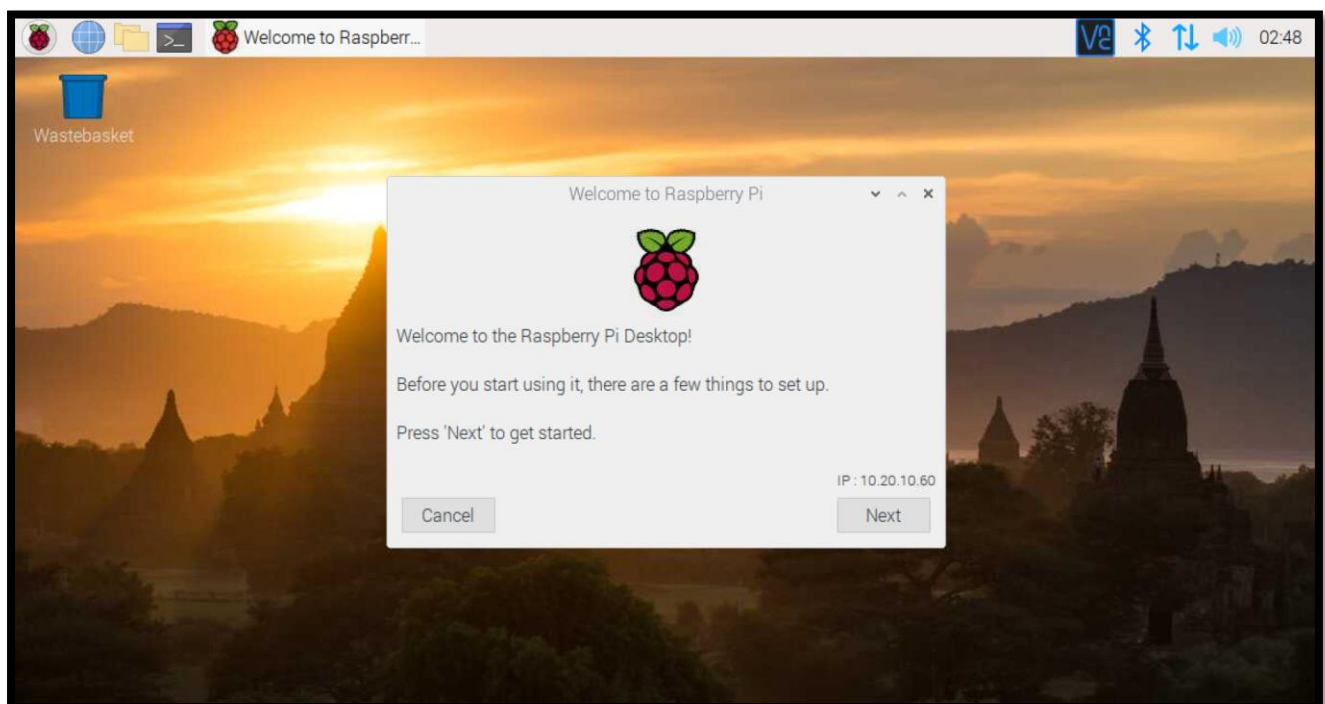
授業資料[0_RaspberryPi の取り扱い方]に記載している起動手順を見て、Raspberry Pi を起動します。

ただし、**初回授業だけ**はスムーズに環境構築を行う為、実習機に差し込んである LAN ケーブルを Raspberry Pi に差し込んで起動してください。**※授業終了時に元に戻すこと**

Raspberry Pi セットアップ

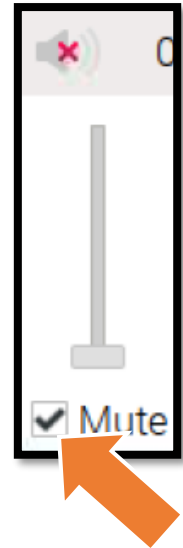
Raspberry Pi OS が起動すると以下のようなデスクトップ画面がディスプレイに表示されます。

下記の手順に従って、今後授業で演習していくためにセットアップを行いましょう。



① ウィザードの音声流れるので右上のサウンドからミュートにしておく。

② ウィザード初期画面の[Next]をクリックする。

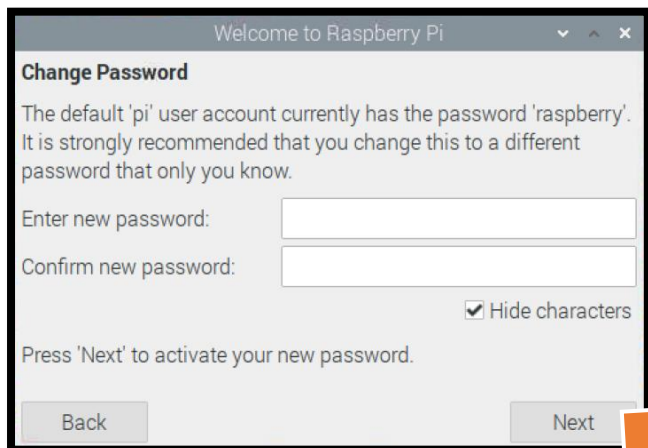


③ Set Country の Country を[Japan]に変更する。

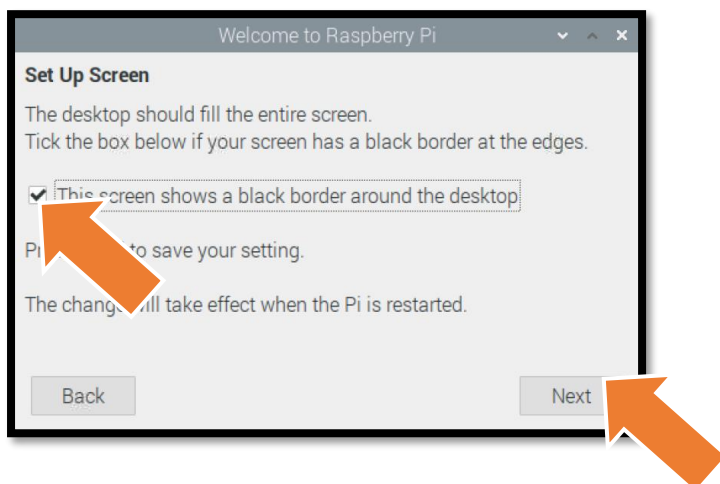
自動的に Language が[Japanese]、TimeZone が[Tokyo]に変更されるので[Next]をクリックする。



④ Change Password では未入力（デフォルト）のまま[Next]をクリックする。

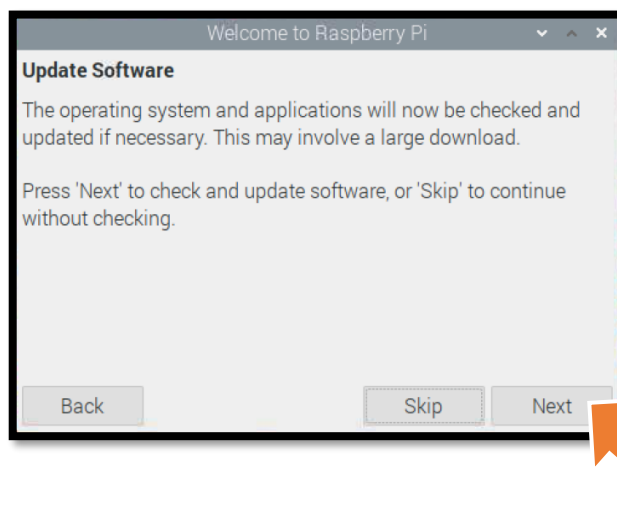
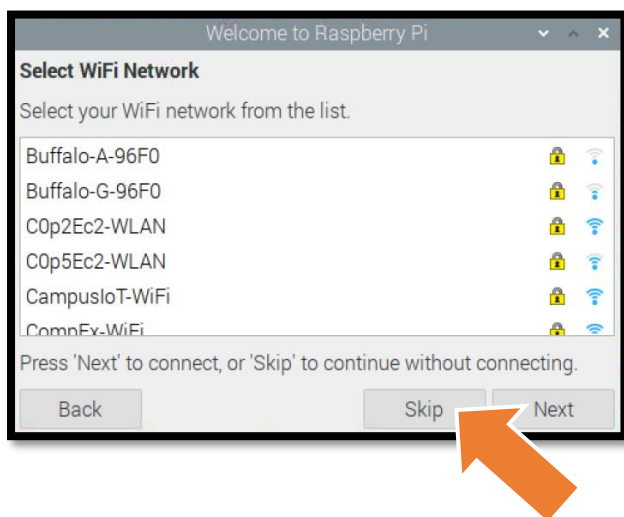


- ⑤ Set Up Screen では[This screen shows~]にチェックを入れて[Next]をクリックする。

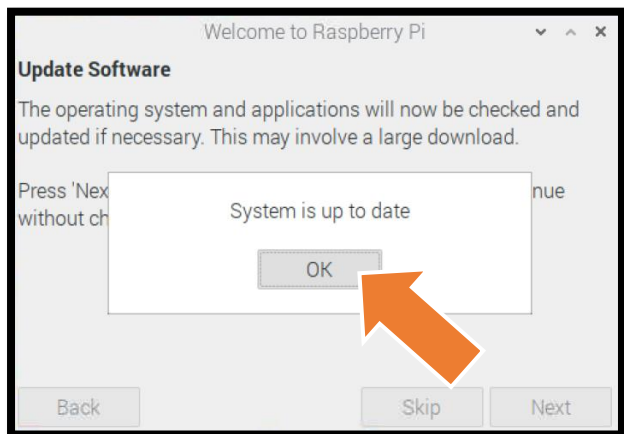


- ⑥ Select WiFi Network ではこの授業内で WiFi は使用しない為、[Skip]をクリックする。
その後、Update Software では[Next]をクリックするとシステムアップデートが始まるのでしばらく待機する（5~10 分掛かります）

※LAN ケーブルを Raspberry Pi 本体に接続しているか確認してアップデートを始めること。



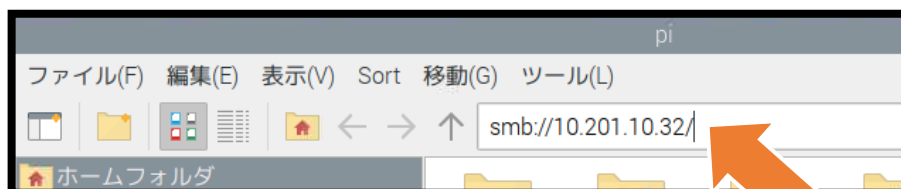
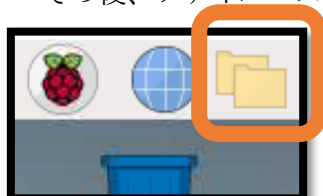
- ⑦ システムアップデートが完了したら[OK]をクリックする。
その後、Setup Complete では[Restart]をクリックすると、再起動が始まるので待機する。



環境構築

この授業では他科目で学んでいる Java を活用して Raspberry Pi の GPIO を制御します。
 下記の手順に従って演習をしていく為の環境構築を行きましょう。

- ① 左上部にあるフォルダのアイコンをクリックしてファイルマネージャーを開く。
 その後、ファイルパス欄に[<smb://10.201.10.32/>]と入力してエンターキーを押す。



- ② ユーザー名：学籍番号
 ドメイン：maetel
 パスワード：@から始まる配布パスワード
 を入力し、[期限なしで記憶する]を選択して
 接続するをクリックする。
 間違いなく入力出来ていれば学校の nt22
 (配布サーバ) に接続できる。

※何度か入力要求される場合がありますが、
 同様の内容を入力すれば大丈夫です。



- ③ [配布]->[IT]->[村上]とフォルダを開き、その中にある[電子制御演習]フォルダを右クリックして
 コピーし、デスクトップに貼り付ける。これにて今後の授業資料のダウンロードが完了となる。

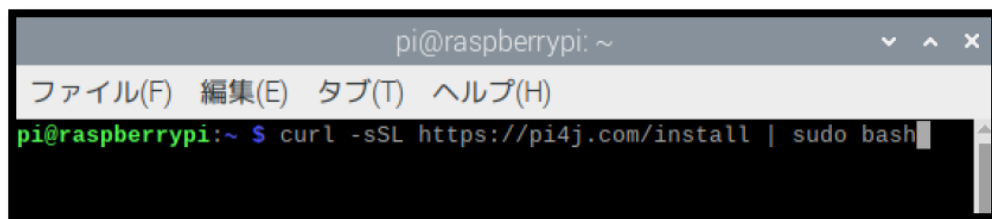


右クリック → 貼り付け

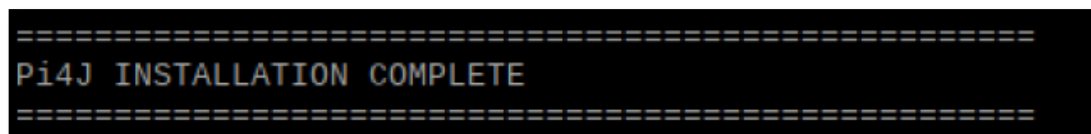
- ④ 左上部にある黒いアイコンをクリックして LXTerminal を開く。

その後、[`curl -sSL https://pi4j.com/install | sudo bash`] と入力し、エンターキーを押す。

Java で GPIO を制御する為の Pi4j パッケージのインストールが始まるのでしばらく待機する。



[Pi4J INSTALLATION COMPLETE] と表示されるとインストール完了となる。



- ⑤ 引き続き LXTerminal 上で [`sudo nano /etc/profile`] と入力してエンターキーを押す。

テキストファイルが開かれるが、[↓] キーを押し続けて最下部までカーソル移動させる。



その後、[`export CLASSPATH=./opt/pi4j/lib/*:/home/pi/java/lib/*;`] と入力する。

※打ち間違いのないよう確認すること

これにて Pi4j パッケージやライブラリをプログラムから使用することが可能となる。



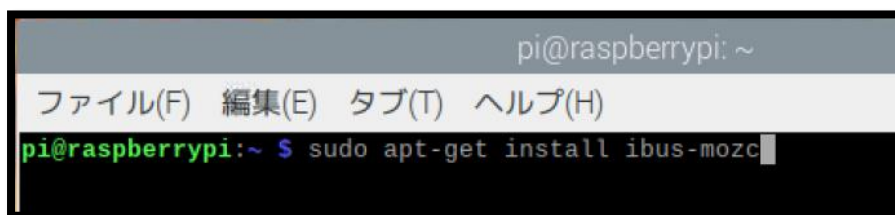
入力後、[Ctrl]キーを押しながら[O (オー)]キーを押す。

その後、エンターキーを押すとファイルの書き込みが完了する。



その後、[Ctrl]キーを押しながら[X]キーを押すとファイルが閉じられる。

- ⑥ 引き続き LXTerminal 上で[`sudo apt-get install ibus-mozc`]と入力しエンターキーを押す。



日本語入力用のソフトウェア mozc (モズク) のインストールが始まるのでしばらく待機する。

※以降、途中で続行するか問われた際には[Y]を入力しエンターキーを押してインストールを行う。

- ⑦ 引き続き LXTerminal 上で[`sudo apt install openjdk-11-jdk`]と入力しエンターキーを押す。



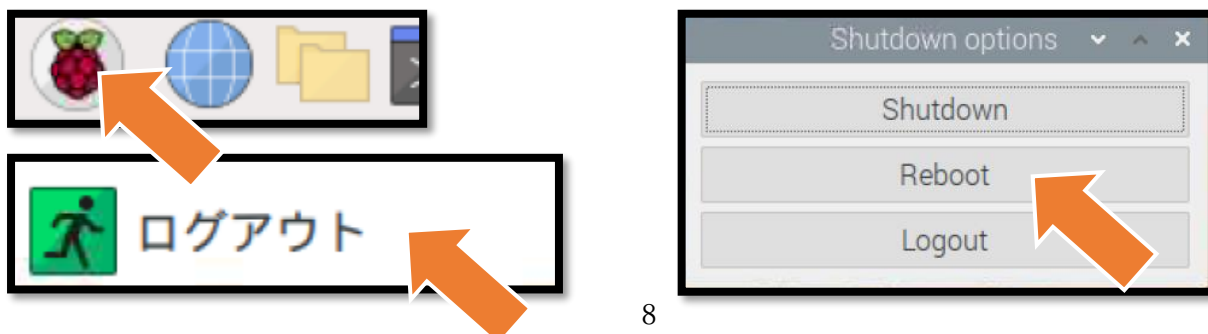
Java 実行用 JDK のインストールが始まるのでしばらく待機する。

コンソールに入力できるようになったら、[`java --version`]と入力しエンターキーを押す。

以下のようにバージョンと日付が表示されていればインストール完了となる。

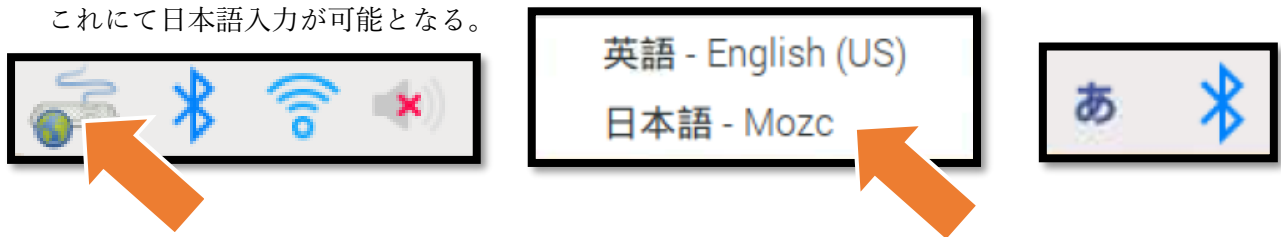


- ⑧ 上記でインストールしたソフトウェアを反映させる為、Raspberry Pi の再起動(Reboot)を行う。

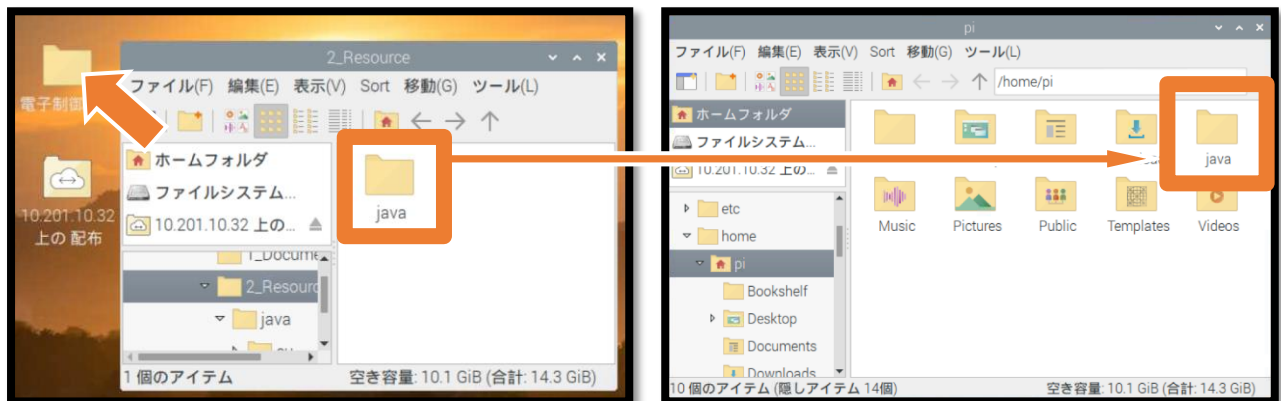


- ⑨ 画面右上部の入力モードアイコンから先程インストールした Mozc を有効にする。

これにて日本語入力が可能となる。



- ⑩ デスクトップにダウンロードした[電子制御演習]->[2_Resource]フォルダ内にある[java]フォルダを[/home/pi]階層に移動させる。この java フォルダがこの授業の作業領域になります。



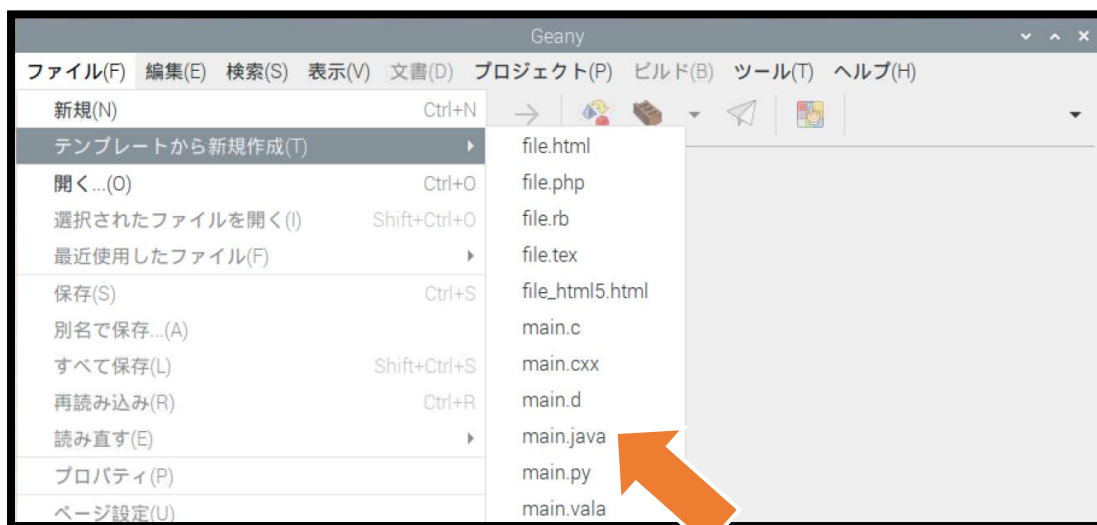
Java プログラム実行方法

この授業では Geany (ジーニー) という開発ツールを使って演習を行います。

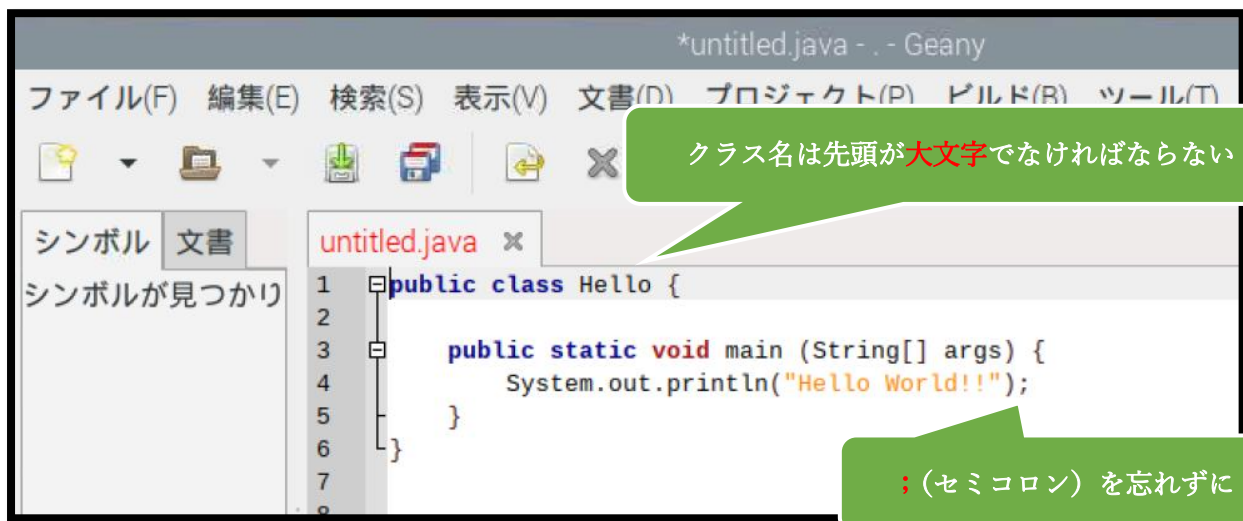
左上部からアイコンをクリックして実行しましょう。



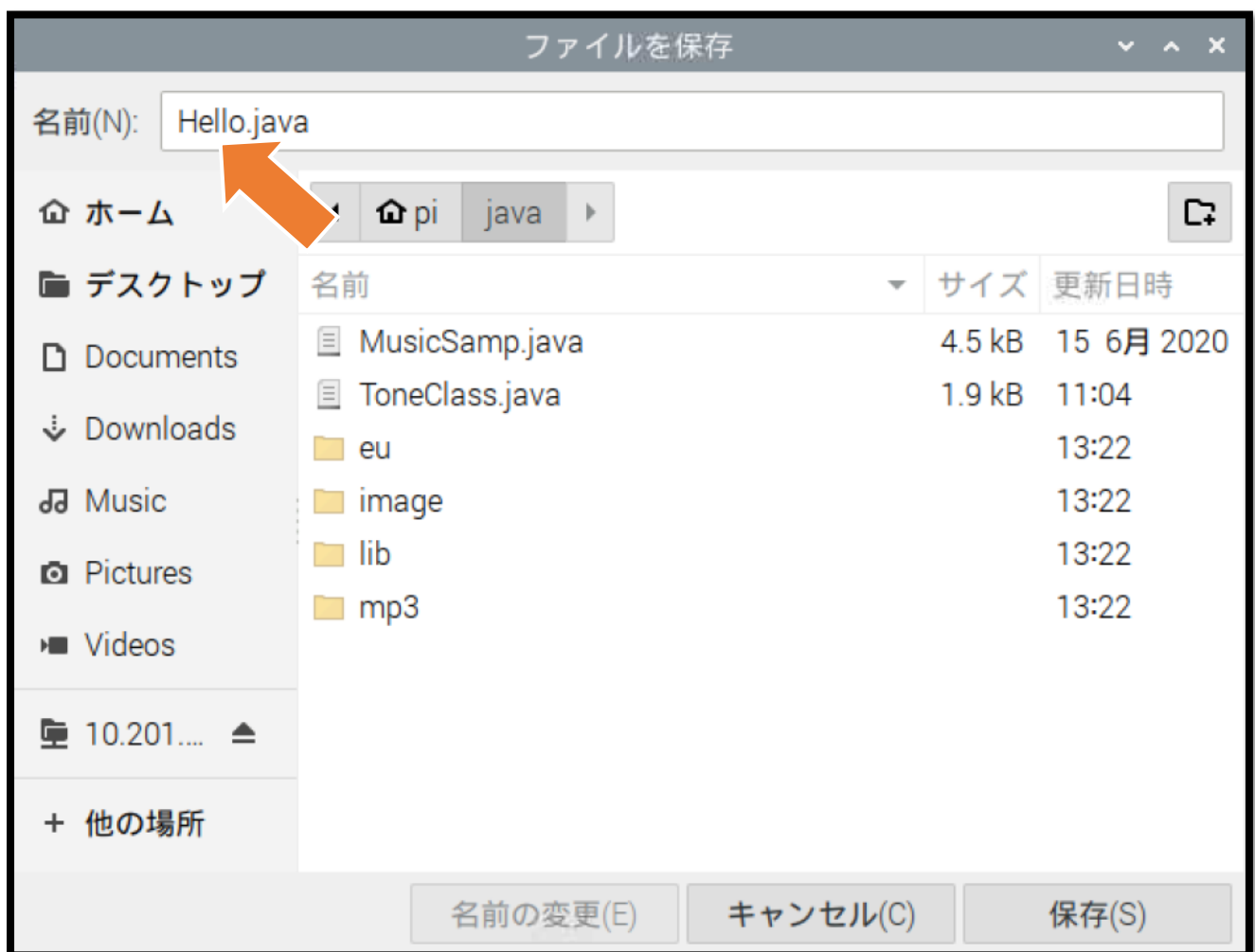
- ① 上部タブから[ファイル]->[テンプレートから新規作成]->[main.java]をクリックする。



- ② Untitled(タイトルなし)クラスのテンプレートが作成されるので、上部の赤いコメントを削除し、クラス名を[**Hello**]にして main メソッド内に[**System.out.println("Hello World!!");**]と入力する。

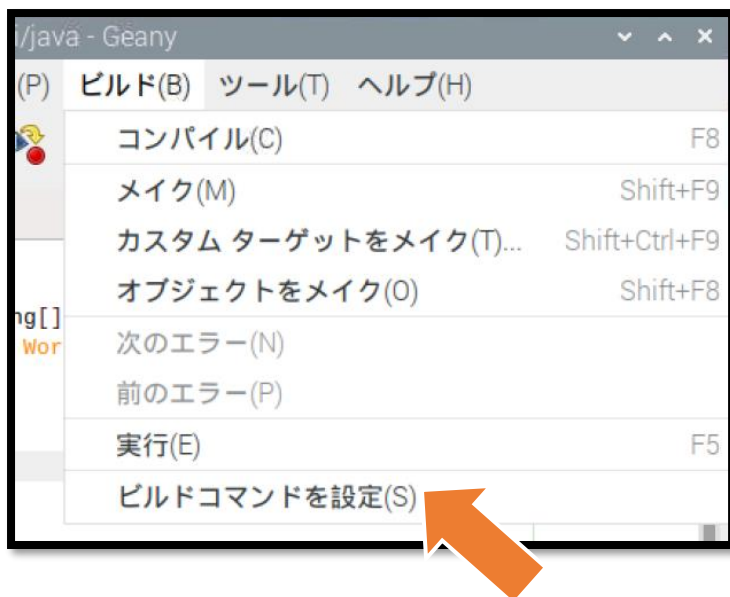


- ③ 上部のタブから[ファイル]->[別名で保存...]をクリックし、名前を[**Hello.java**] (クラス名.java) にして、[**/home/pi/java**]フォルダ (作業領域) 内に保存する。



- ④ 管理者権限で実行する為にビルドコマンド設定を行う。※以降の課題では行わなくて良い。

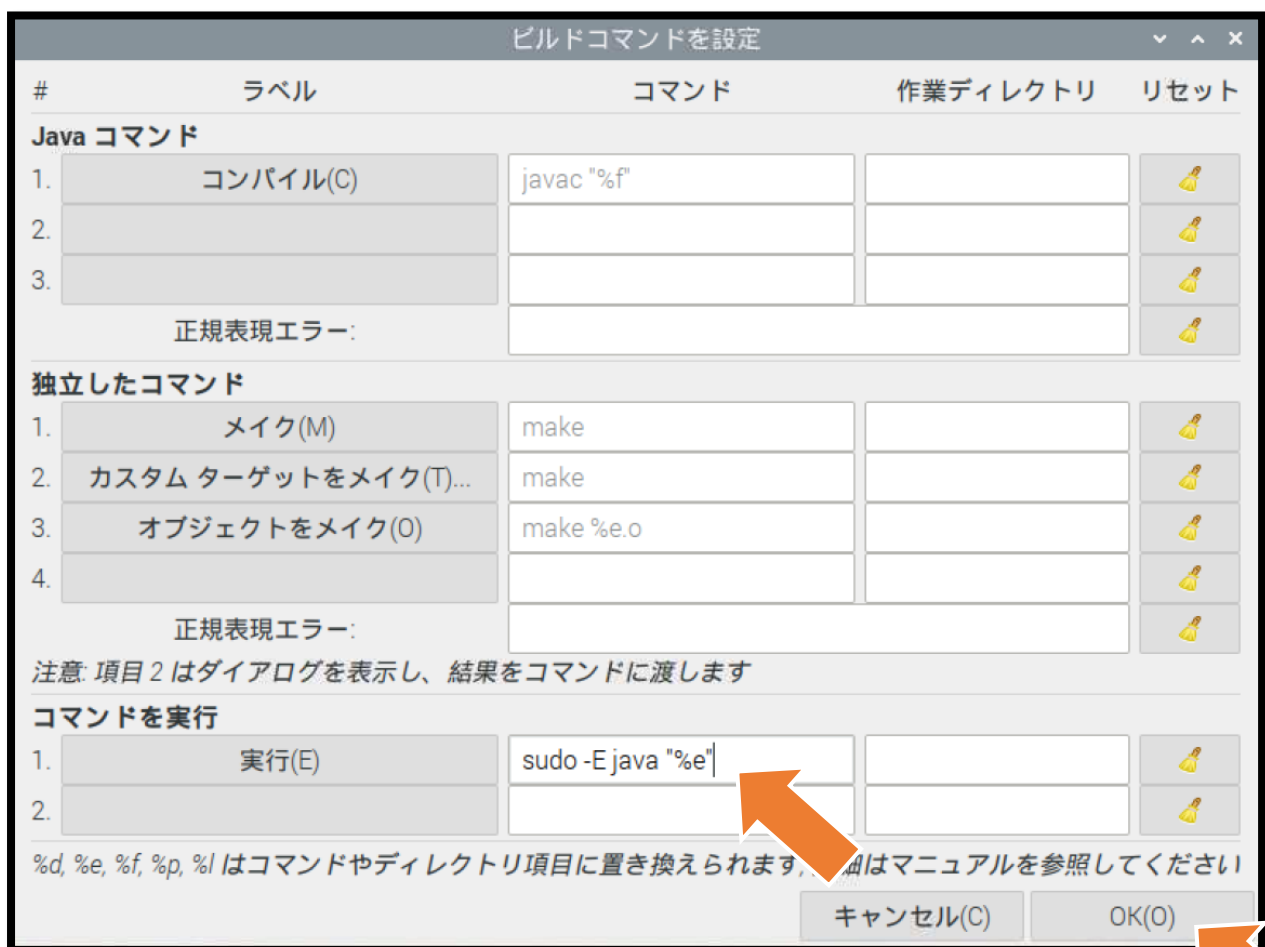
上部の[ビルド]->[ビルドコマンドを設定]をクリックする。



最下部の[コマンドを実行]->[実行]のコマンドを[java "%e"]から

[**sudo -E java "%e"**]に変更する。 ※全て半角で、-E の-はハイフンです

その後、[OK]をクリックして閉じる。



- ⑤ Java プログラムを実行する場合、まずはコンパイル（機械語に翻訳）を行う。

上部の **コンパイルアイコン** をクリックするとコンパイルが開始される。

※文法ミスや環境構築手順でミスをしているとこの時点でコンパイルエラーが発生する。

コンパイル後、下部のコンパイラが **[コンパイル完了]** と出力した場合は実行可能なので、上部の **実行アイコン** をクリックするとプログラムの実行が開始される。



実行が成功すると以下のようにコンソール画面に print 文で記述した内容が表示される。



以上が Java プログラムの実行方法の手順です。

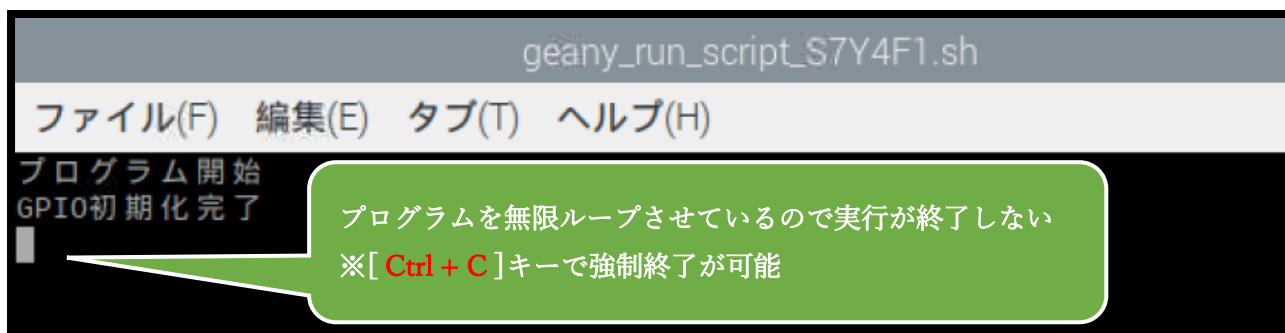
[Java ファイル作成] → [名前を付けて保存] → [コーディング(保存)] → [コンパイル] → [実行]
という流れを覚えて以降の課題に取り組みましょう。

課題 (Template.java)

下記のサンプルプログラムを入力して、今後の課題で電子制御を行う為に記述するプログラムのテンプレート（雛型）を作成しなさい。

電子制御はプログラムを終了させずに実行し続ける為、メイン処理は無限ループするのが基本となる。

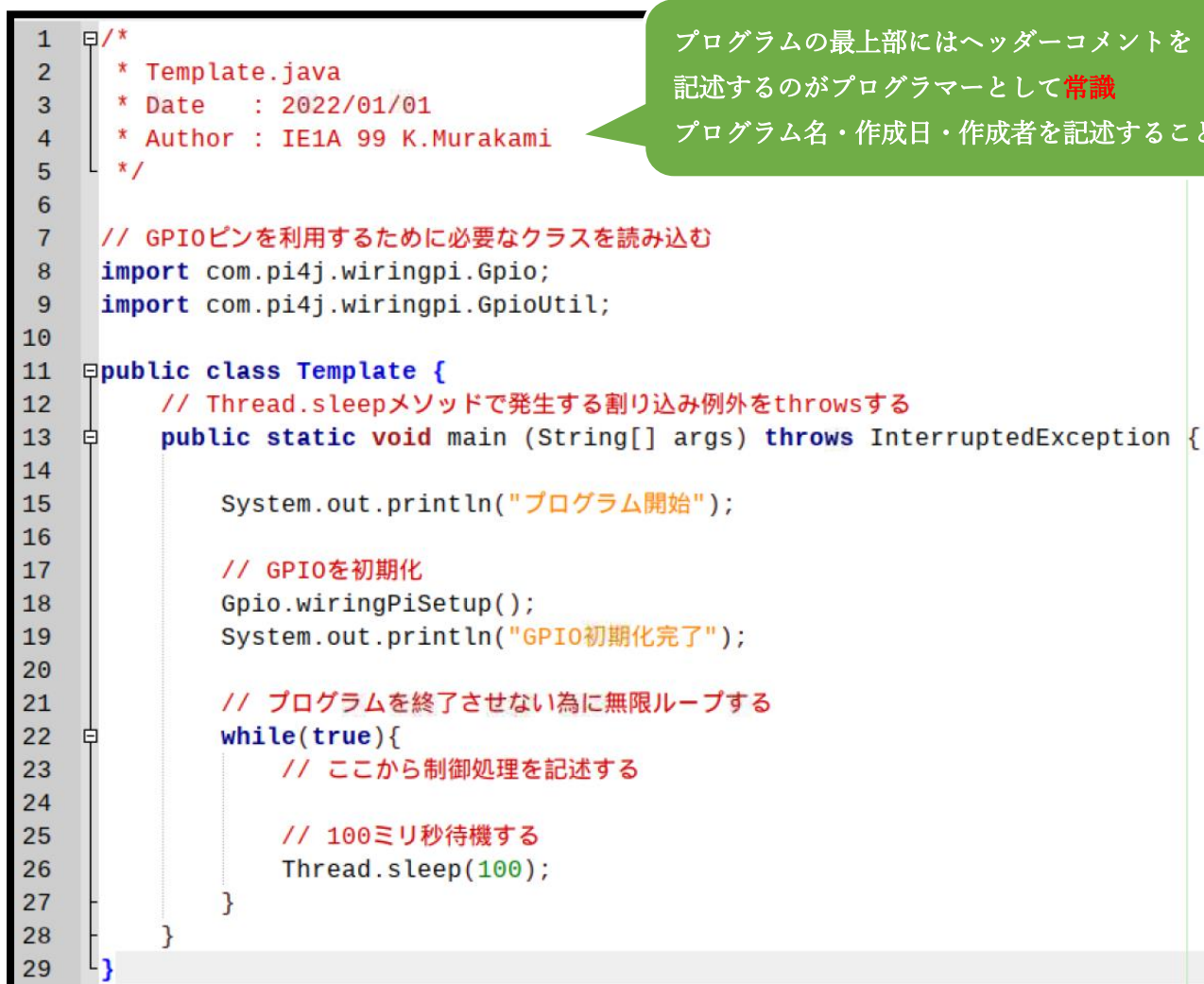
◆実行結果



```
geany_run_script_S7Y4F1.sh
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
プログラム開始
GPIO初期化完了
```

プログラムを無限ループさせているので実行が終了しない
※[Ctrl + C]キーで強制終了が可能

◆サンプルプログラム



```
1  /*
2   * Template.java
3   * Date   : 2022/01/01
4   * Author : IE1A 99 K.Murakami
5   */
6
7   // GPIOピンを利用するために必要なクラスを読み込む
8   import com.pi4j.wiringpi.Gpio;
9   import com.pi4j.wiringpi.GpioUtil;
10
11  public class Template {
12      // Thread.sleepメソッドで発生する割り込み例外をthrowsする
13      public static void main (String[] args) throws InterruptedException {
14
15          System.out.println("プログラム開始");
16
17          // GPIOを初期化
18          Gpio.wiringPiSetup();
19          System.out.println("GPIO初期化完了");
20
21          // プログラムを終了させない為に無限ループする
22          while(true){
23              // ここから制御処理を記述する
24
25              // 100ミリ秒待機する
26              Thread.sleep(100);
27          }
28      }
29  }
```

プログラムの最上部にはヘッダーコメントを記述するのがプログラマーとして常識
プログラム名・作成日・作成者を記述すること