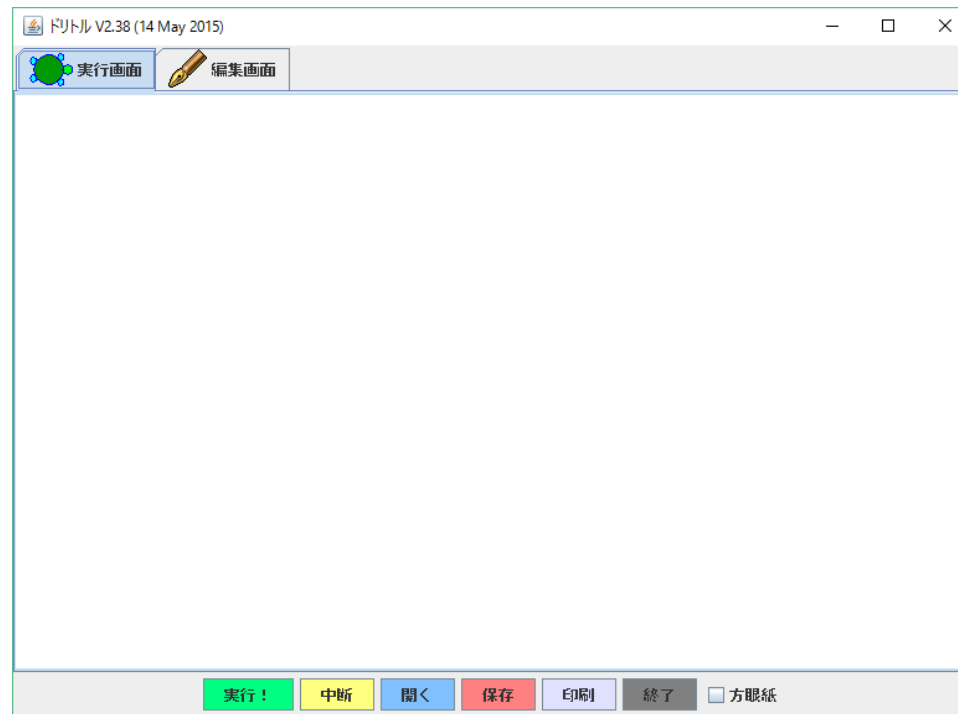


プログラミング基礎演習1

第1回 ガイダンス

ドリトルを起動する

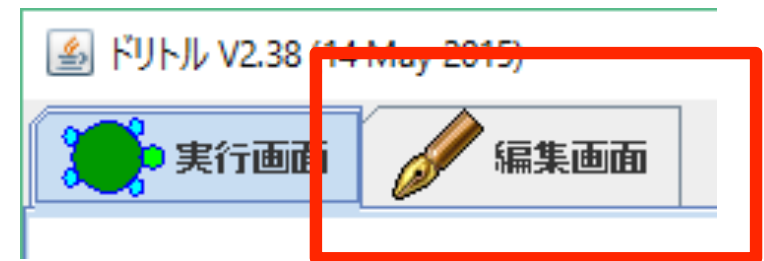
- スタート→すべてのプログラム→ドリトル
- dolittle.batをダブルクリック



※ 一緒に起動する黒い画面は閉じないこと！

プログラムの編集と実行

- プログラムを編集する
 - 上のタブを押すと
編集画面に切り替わり
プログラムを記述できる



- プログラムを実行する
 - 実行ボタンを押すと、
プログラムの実行結果を
「実行画面」に表示する

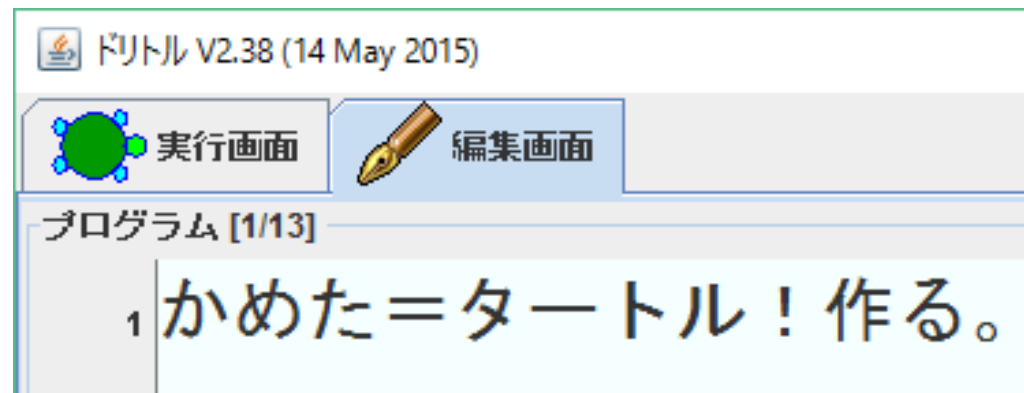


では、実際にプログラムを体験してみましょう
今日は「お花拾いゲーム」をつくります

主人公を作る

- 編集画面の1行目に次のプログラムを入力

かめた＝タートル！作る。

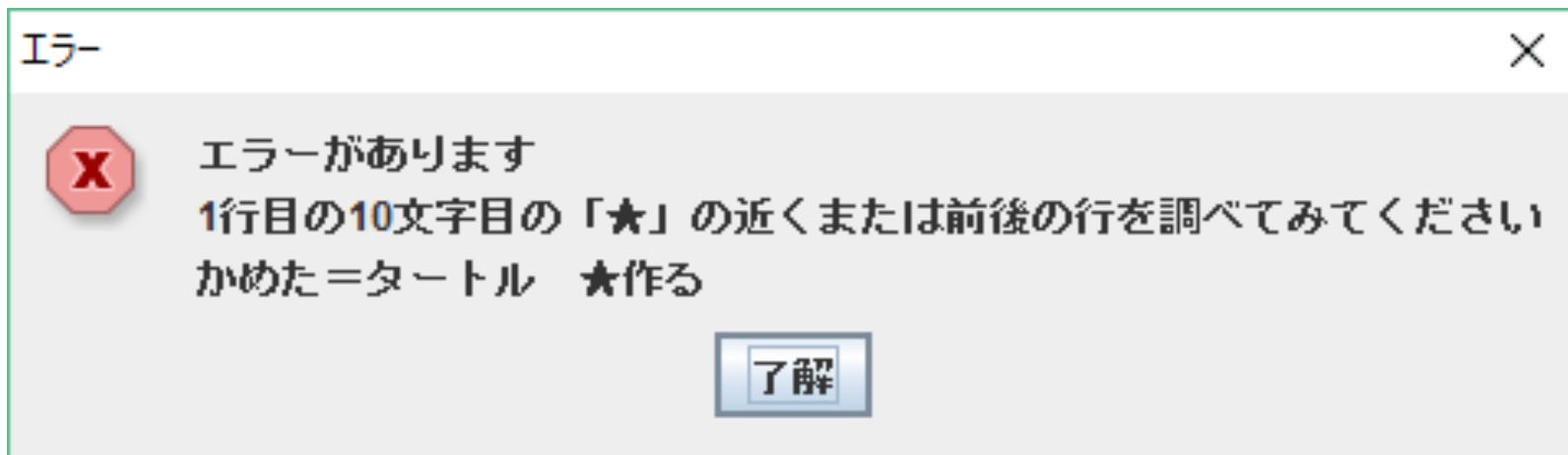


- 入力ができたら、「実行」ボタンをクリック



エラーが出た！

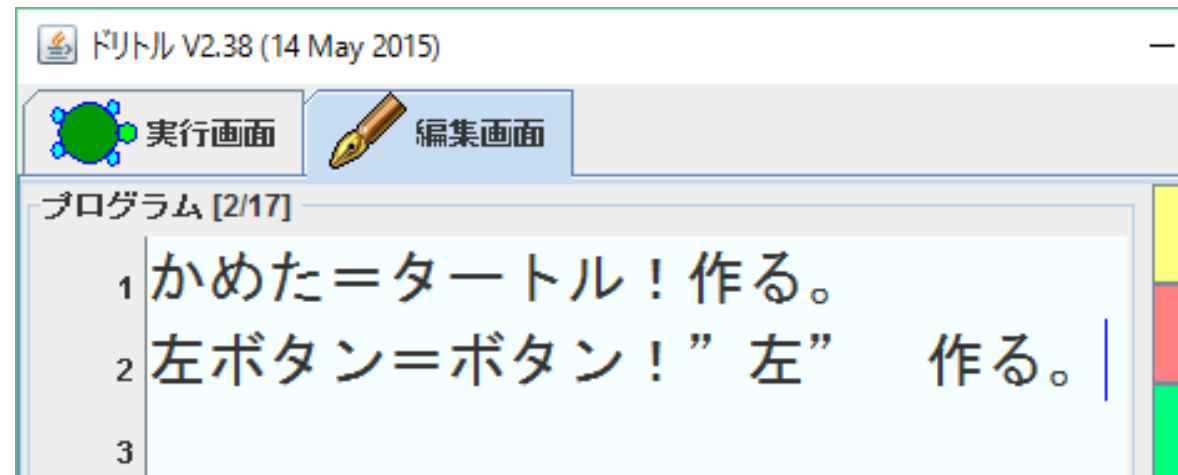
- もしも、エラーが出たら「★」マークの近く、または前後の行のプログラムを調べて、間違いを修正しましょう
- エラーを直すことを「デバッグ」と呼びます



カメが左に曲がるボタンを作ろう

- 2行目に次のプログラムを入力

左ボタン=ボタン！“左” 作る。



- 入力したら、「実行」ボタンをクリックし、作成したボタンの動作を確認してみよう



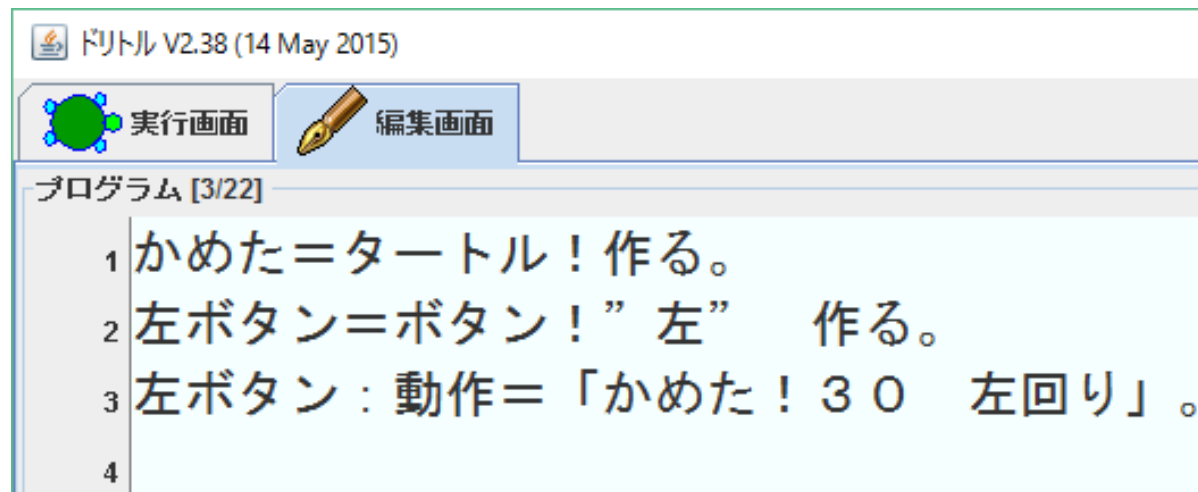
なぜ動かない？

- プログラムは書いたとおりにしか動かない
- 左ボタンは作ったが、
ボタンが押されたら何をするかという
プログラムは書いていない
- ボタンが押されたら、
カメが左に曲がるようプログラムを書こう

ボタンを押したときの動作をつけよう

- 3行目に次のプログラムを入力

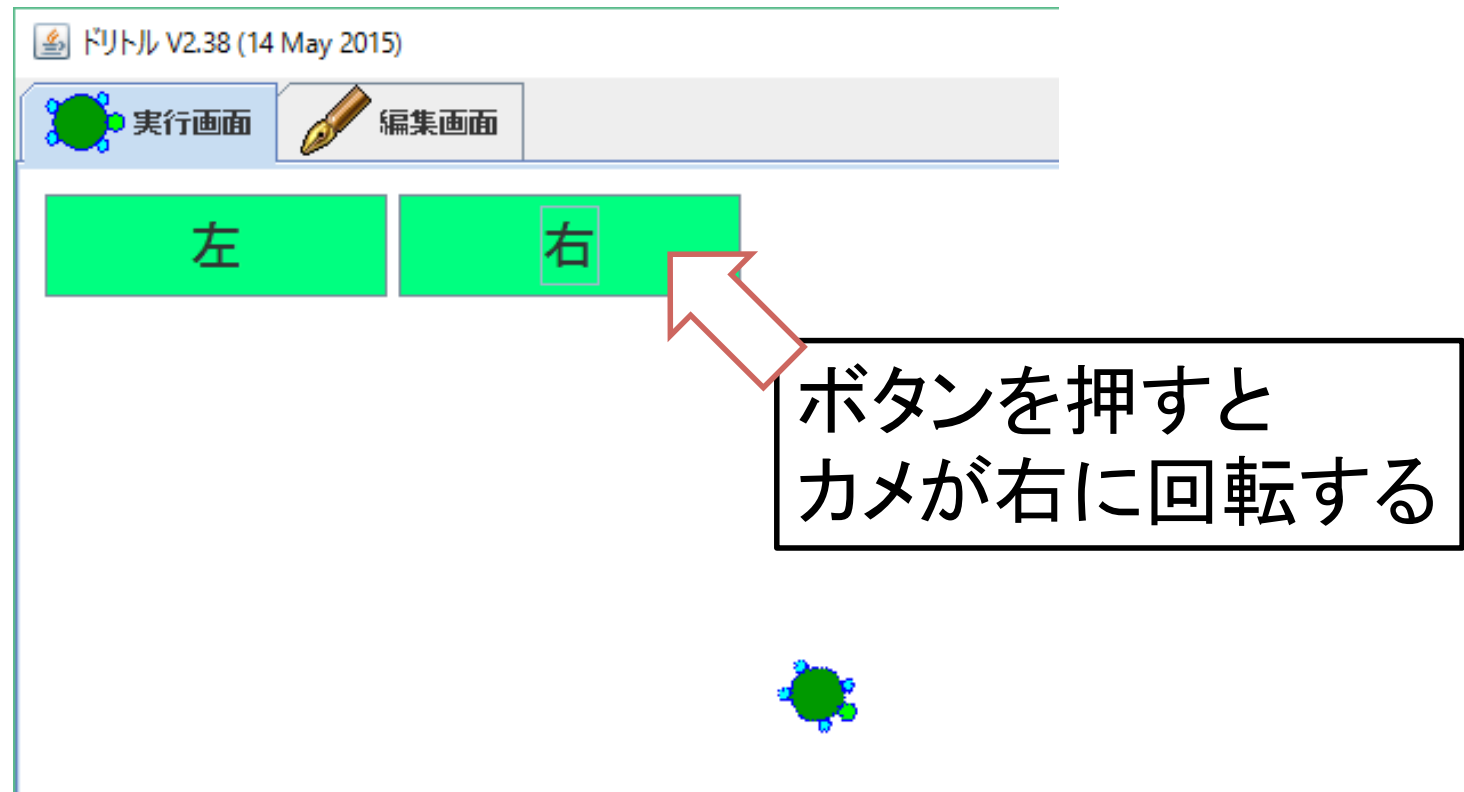
左ボタン：動作＝「かめた！30 左回り」。



- 入力したら「実行」ボタンをクリックし、動きを確認してみよう

カメラが右に曲がるボタンを作ろう

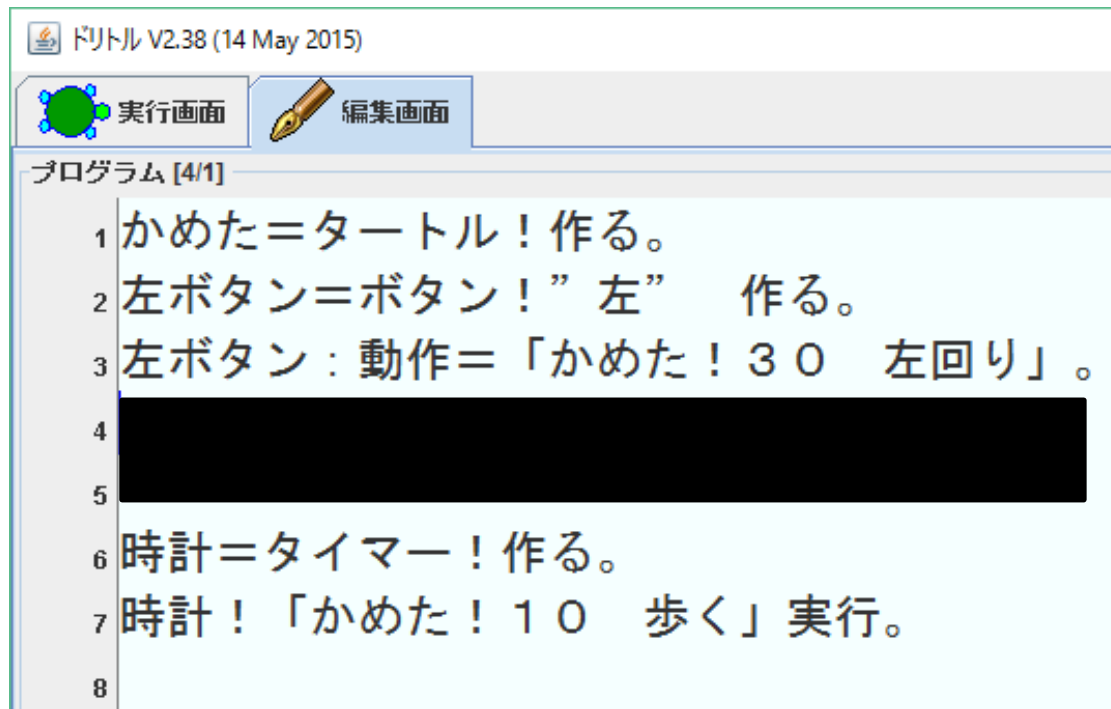
- 4行目と5行目にプログラムを追加し、ボタンを押すとカメラが右に曲がるようにしよう



カメを動かそう

- 6行目と7行目に次のプログラムを入力

時計＝タイマー！作る。
時計！「かめた！10 歩く」実行。



The screenshot shows the Drift V2.38 software interface. At the top, it says 'ドリトル V2.38 (14 May 2015)'. Below this are two tabs: '実行画面' (Execution Screen) and '編集画面' (Edit Screen). The '編集画面' tab is selected. The main area is titled 'プログラム [4/1]' and contains a list of program lines. Line 1 is 'かめた＝スタート！作る。', line 2 is '左ボタン＝ボタン！” 左” 作る。', line 3 is '左ボタン：動作＝「かめた！30 左回り」。', line 4 is a blacked-out line, line 5 is a blacked-out line, line 6 is '時計＝タイマー！作る。', line 7 is '時計！「かめた！10 歩く」実行。', and line 8 is empty.

```
1 かめた＝スタート！作る。
2 左ボタン＝ボタン！” 左” 作る。
3 左ボタン：動作＝「かめた！30 左回り」。
4 
5 
6 時計＝タイマー！作る。
7 時計！「かめた！10 歩く」実行。
8
```

入力したら実行し、
動きを確認しよう

カメが拾うお花を作ろう

- 8行目に次のプログラムを入力

花1=タートル！作る “tulip.png” 変
身する ペンなし 100 100 位置。

```
6 時計=タイマー！作る。  
7 時計！「かめた！10 歩く」実行。  
8 花1=タートル！作る “tulip.png” 変身する| ペンなし 100 100 位置。  
9
```

- 入力したら「実行」ボタンをクリックし、動きを確認してみよう

お花が消えるようにしよう

- カメとお花が重なったときに、お花が消えるようにします
- 9行目に次のプログラムを入力

「|」の入力は、「Shift」を押しながら「¥」を押す。

かめた:衝突=「| 相手 | 相手 ! 消える」。

```
7 時計! 「かめた! 10 歩く」 実行。  
8 花1=タートル! 作る “tulip.png” 変身す  
9 かめた:衝突=「| 相手 | 相手 ! 消える」。  
10
```

- 入力したら「実行」ボタンをクリックし、動きを確認してみよう

プログラムを改良しよう

- これでお花拾いゲームは完成
- 自分の好きなようにプログラムを改良しよう

例1: お花の数を増やす

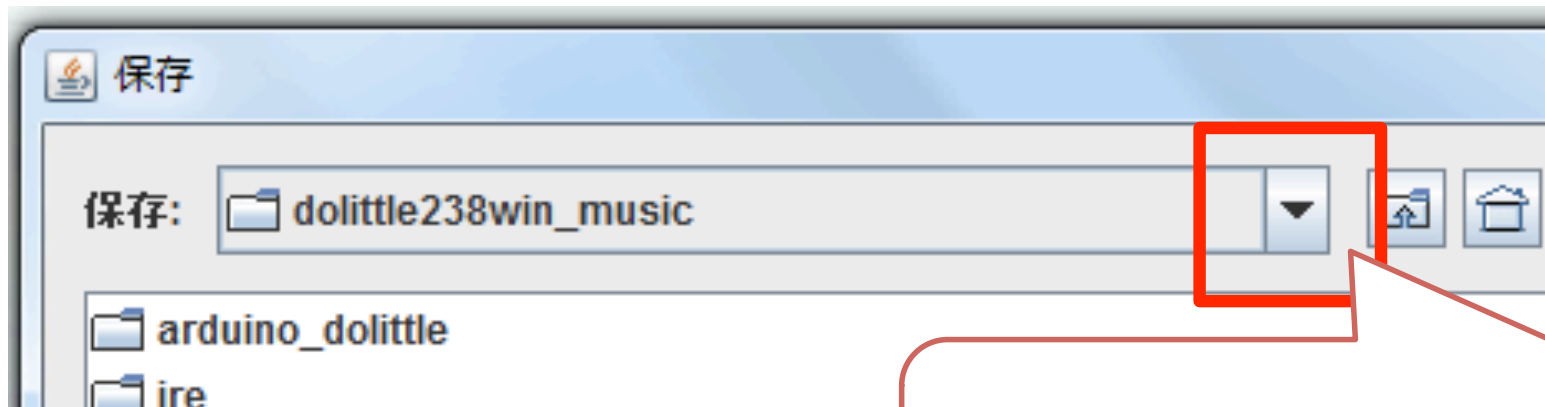
例2: お花を置く場所をランダムで変える

花1=タートル！作る “tulip.png” 変身する
ペンなし 乱数((600)-300) 乱数((600)-300) 位置。

プログラムが完成したら保存

1. 「保存」ボタンをクリック
2. 保存先のフォルダ名を作成したフォルダに変更

ドキュメント → プログラミング基礎演習1



「▼」を押すと、保存する
フォルダが選択できる