

# 放送大学「コンピュータとソフトウェア('18)」

放送大学「コンピュータとソフトウェア('18)」でのドリトルの利用について説明します。放送では、第8回と第10回でドリトルのプログラミングを扱っています。

番組のシラバス(2018年度) : <http://www.ouj.ac.jp/hp/kamoku/H30/kyouyou/C/joho/1570285.html>

ドリトルは「オンライン版」「BitArrow版」「インストール版」の3通りの方法で使えます。オンライン版はいちばん手軽で、ログインやインストールなしに使えます。

## 準備

オンライン版はWindowsのEdge、MacのSafariをはじめ、ChromeやFirefoxなど代表的なブラウザで使えます。Windowsの古いブラウザであるIE（インターネットエクスプローラー）では動作しないかもしれませんが、タブレットやスマートフォンでも文字の入力は大変かもしれませんが動くと思います。 [オンライン版](#)をクリックして起動してください。

## 実習

画面が起動すると、左側に「かめた = タートル！作る（青）線の色 ゆっくり動く。」という文字が表示されています。これはプログラムのサンプルで、操作するためのカメのキャラクターを画面に表示します。学習時には次の行からプログラムを書いていきます。



画面の左上にある「実行」ボタンを押すと、画面の右側にカメ（タートル）が表示されると思います。これがこのプログラムが作ったかめたです。



テキストでは簡単のために次のように書かれています。この場合、カメは黒い線を引きながら高速に動きます。

```
かめた = タートル！作る。  
かめた！100 歩く。
```



次のように書いた場合には、カメは青い線を引きながらゆっくり動くようになります。テキストと少し違いますが、1行目はこのまま使ったほうがわかりやすいと思います。

```
かめた = タートル！作る（青）線の色 ゆっくり動く。  
かめた！100 歩く。
```



これらの例のように、テキストや講義のプログラムを入力して学習を進めてください。

ドリトルの説明は、[ドリトルのサイト](#)を参照してください。詳しい機能は[マニュアル](#)で説明されています。

## 保存と終了

学習が終わったら、必要であればプログラムを画面からコピーして他のテキストエディタなどに貼り付けて保存してください。ブラウザのドリトルは、そのままウィンドウを閉じてても問題ありません。

## 第8回の内容

### 「歩いて左を向く」プログラム（6分頃）

ドリトルのプログラムは「[オンライン版](#)」から実行できます。左側にプログラムを書いて、「実行」を押すと、右側に結果が表示されます。

```
かめた = タートル！作る（青）線の色 ゆっくり動く。  
かめた！ 1 0 0 歩く 9 0 左回り。
```

### 「四角形を描く」プログラム（7分頃）

```
かめた = タートル！作る（青）線の色 ゆっくり動く。  
かめた！ 1 0 0 歩く 9 0 左回り。  
かめた！ 1 0 0 歩く 9 0 左回り。  
かめた！ 1 0 0 歩く 9 0 左回り。  
かめた！ 1 0 0 歩く 9 0 左回り。
```

### 繰り返して「四角形を描く」プログラム（8分頃）

```
かめた = タートル！作る（青）線の色 ゆっくり動く。  
□  
  かめた！ 1 0 0 歩く 9 0 左回り。  
」！ 4 回 繰り返す。
```

### 繰り返して「五角形を描く」プログラム（10分頃）

```
かめた = タートル！作る（青）線の色 ゆっくり動く。  
□  
  かめた！ 1 0 0 歩く 7 2 左回り。  
」！ 5 回 繰り返す。
```

### 繰り返して「いびつな星を描く」プログラム（11分頃）

```
かめた = タートル！作る（青）線の色 ゆっくり動く。  
□
```

```
かめた！ 1 0 0 歩く 1 4 0 右回り。  
」！ 5 回 繰り返す。
```

## 繰り返して「星を描く」プログラム（12分頃）

```
かめた = タートル！作る（青）線の色 ゆっくり動く。  
□  
かめた！ 1 0 0 歩く 1 4 4 右回り。  
」！ 5 回 繰り返す。
```

## 繰り返して「不思議な図形を描く」プログラム（13分頃）

```
かめた = タートル！作る（青）線の色 ゆっくり動く。  
□  
かめた！ 1 0 0 歩く 1 3 9 右回り。  
」！ 4 0 回 繰り返す。
```

## 繰り返して「不思議な図形を描く」プログラム（13分頃）

```
かめた = タートル！作る（青）線の色 ゆっくり動く。  
□  
かめた！ 1 0 0 歩く 1 4 5 右回り。  
」！ 4 0 回 繰り返す。
```

## 「対話的な」プログラム（20分頃）

```
かめた = タートル！作る（青）線の色。  
左ボタン = ボタン！"左" 作る。  
左ボタン：動作 = 「かめた！ 3 0 左回り」。  
右ボタン = ボタン！"右" 作る。  
右ボタン：動作 = 「かめた！ 3 0 右回り」。  
時計 = タイマー！作る。  
時計！「かめた！ 1 0 歩く」実行。  
タートル！作る "tulip.png" 変身する ペンなし 1 0 0 1 0 0 位置。  
かめた：衝突 = 「 | 相手 | 相手！消える」。
```

## 「値に名前を付けて覚えておく」プログラム（26分頃）

□の値を 5 だけでなく、3 や 6 などに変えて実行してみましょう。

```
かめた = タートル！作る（青）線の色 ゆっくり動く。  
□□□□  
□  
かめた！ 2 0 0 歩く □□□□ ÷ □□□□ 左回り。  
□□□□□□ 繰り返す。
```

## 「繰り返しの回数などを数える」プログラム（28分頃）

```
かめた = タートル！作る（青）線の色 ゆっくり動く。  
□□□□  
    かめた□□□□×□□□歩く  9 0  左回り。  
」！1 0  繰り返す。
```

## 「2つの処理を切り替えて実行する」プログラム（29分頃）

```
かめた = タートル！作る（青）線の色 ゆっくり動く。  
□□□□  
    かめた！1 0 0  歩く。  
□□□□□□□□□□なら「  
    かめた！9 0  右回り。  
」そうでなければ「  
    かめた！9 0  左回り。  
」実行。  
」！4回 繰り返す。
```

## 「新しい命令を定義して使う」プログラム（30分頃）

```
かめた = タートル！作る（青）線の色 ゆっくり動く。  
かめた：星を描く = 「  
□□□  
    かめた！1 0 0  歩く  1 4 4  右回り。  
」！5回 繰り返す。  
□□  
□  
    かめた！星を描く  1 2 0  左回り。  
」！3回 繰り返す。
```

## 「配列で複数の値を使う」プログラム（32分頃）

```
データ = 配列！1 0  4 0  5 0  2 0  3 0  作る。  
表示 = テキストエリア！作る。  
表示！（データ）書く  改行。  
表示！（データ！要素数？）書く  改行。  
  
合計 = 0。  
データ！「 | 要素 | 合計 = 合計 + 要素」それぞれ実行。  
表示！（合計）書く。
```

## 作りかけの「びよこびよこ」プログラム（38分頃）

このプログラムはC言語のため、ドリトルではなく[Bit Arrow](https://dolittle.eplang.jp/)から実行してください。クラスIDは「guest□□ユーザ名は「tanaka1234□など他の人と重ならない名前でログインしてください。ログイン後は「新規プロジェクト」から□ouj□など任意のプロジェクトを、言語は「C□を指定して作成してください。プログラムの作成画面が開いたら、上部の「ファイル」メニューから「新規」を選び、□pyoko□などお好きな名前でプログラムを作成して実行してください。

```
#include <stdio.h>
main(){
    for (int i=1; i<=4; i++) {
        printf("かえる ");
        printf("ぴょこぴょこ ");
    }
}
```

## 完成した「ぴょこぴょこ」プログラム (39分頃)

```
#include <stdio.h>
main(){
    for (int i=1; i<=4; i++) {
        if (i==1) { printf("かえる ");}
        if (i==2) { printf("み ");}
        if (i==3) { printf("あわせて ");}
        if (i==4) { printf("む ");}
        printf("ぴょこぴょこ ");
    }
}
```

From:

<https://dolittle.eplang.jp/> - プログラミング言語「ドリトル」

Permanent link:

<https://dolittle.eplang.jp/ouj18?rev=1527160034>

Last update: **2018/05/24 20:07**

