

# 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)」でのドリトルとサクセス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、第11回でサクセスとSQL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！ 100 歩く 90 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！ 100 歩く 90 左回り。  
かめた！ 100 歩く 90 左回り。  
かめた！ 100 歩く 90 左回り。  
かめた！ 100 歩く 90 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！ 100 歩く 90 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！ 100 歩く 72 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! 100 歩く 140 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! 100 歩く 144 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! 100 歩く 139 右回り。  
」! 40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! 100 歩く 145 右回り。  
」! 40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! 30 左回り」。

右ボタン = ボタン! "右" 作る。

右ボタン: 動作 = 「かめた! 30 右回り」。

時計 = タイマー! 作る。

時計! 「かめた! 10 歩く」実行。

タートル! 作る "tulip.png" □ 変身する ペンなし 100 100 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! 200 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 90 左回り。  
」！10 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！100 歩く。  
□□□□□□□□□□なら「  
    かめた！90 右回り。  
」そうでなければ「  
    かめた！90 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！100 歩く 144 右回り。  
」！5回 繰り返す。  
□□  
□  
    かめた！星を描く 120 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！10 40 50 20 30 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](#) 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ご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("びよこびよこ ");
    }
}
```

## 完成した「びよこびよこ」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("びよこびよこ ");
    }
}
```

## 第9回の内容

### 「ユークリッドの互除法」プログラム (16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。

```
x=125□
y=35□
「全部!(x!=y)(x>0)(y>0)本当!」の間「
    □x>y□!なら□x=x-y□実行。
    □x<y□!なら□y=y-x□実行。
」実行。
ラベル!(x)作る。
```

### 「選択ソート」プログラム (21分頃)

```
c=配列!64371825作る。
ラベル!(c)作る 次の行。
n=c!要素数?。
□|i|
    e=n-i+1□
    □|s|
        □(c!(s)読む) > (c!(e)読む)!なら
```

```

    t=(c!(s)読む)。
    c!(s)(c!(e)読む)上書き。
    c!(e)(t)上書き
  」実行。
  !!(e)繰り返す。
  !(n-1)繰り返す。
  ラベル!(c)作る 次の行。

```

## 「二分探索」プログラム (28分頃)

```

Hanoi=
  |n from to work|
  !n>0!なら「
    !(n-1)(from)(work)(to) Hanoi
    ラベル!("円盤"+n+"を"+from+"から"
            +to+"に移動させる。")作る 次の行。
    !(n-1)(work)(to)(from) Hanoi
  」実行。
  !3 "A" "B" "C" Hanoi

```

## 「ハノイの塔」プログラム (36分頃)

```

c=配列!1 3 4 7 8 10 12 13 15 作る。
n=c!要素数?。
q=7
start=1
end=n
!start<=end!の間「
  m=floor((start+end)/2)
  !(c!(m)読む)==q!なら「結果=q!実行。
  !q<=(c!(m)読む)!なら!end=m-1!実行。
  !(c!(m)読む)<=q!なら!start=m+1!実行。
」実行。
「結果==q!なら「ラベル!(結果)作る」
そうでなければ「ラベル!『見つからない』作る」
実行。

```

## 「クイックソート」プログラム (40分頃)

```

クイック=「|対象;結果 n 閾値 小さい方
大きい方 v |
  結果=対象。
  n=対象!要素数?。
  !n>1!なら「
    閾値=対象!(n) 読む。
    小さい方=配列!作る。
    大きい方=配列!作る。
    !i|

```

```

        v=対象!(i)読む。
        □v<閾値」!なら「小さい方!(v)書く」
        そうでなければ「大きい方!(v)書く」
    実行。
    □!(n-1)繰り返す。
結果=配列!作る
    (!!(小さい方)クイック)
    (閾値)
    (!!(大きい方)クイック)
    連結。
」実行。
結果。
□□

```

```

入力=配列!5 3 7 2 8 6 1 4 作る。
ラベル!(入力)作る 次の行。
ラベル!(!(入力)クイック)作る 次の行。

```

## 第10回の内容

### 「出会うまでの時間」プログラム（14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。

```

xa=0□va=1.2□ラベル!0 作る 0-10 位置。
xb=200□vb=0.9□ラベル!200 作る 200-10 位置。
かめA□タートル!作る (xa□0□位置 (青) 線の色。
かめB□タートル!作る ペンなし□xb□0□位置 180 右回り "ayumiAka.gif"□変身する ペンあり
(赤) 線の色。
かめA□位置表示□□□f==undef□□なら「ラベル!(((かめA□横の位置?)+(かめB□横の位置?))/2)作る
95 40 位置□:f=1□実行」。
かめA:衝突 = 「時計!中断。!位置表示。」。
時計 = タイマー!作る。
時計!「
    かめA□□va□歩く。
    かめB□□vb□歩く。
」実行。

```

### 「つるかめ算」プログラム（20分頃）

```

合計 = 1 0 0。
足 = 3 4 2。

「 | つる |
    かめ = 1 0 0 - つる。
    「 4 * かめ + 2 * つる = 足 」 ! なら 「 つる数 = つる。 かめ数 = かめ。 」 実行。
」 ! 1 0 0 回 繰り返す。

```

ラベル! ("つるは"+つる数+"羽、かめは"+かめ数+"匹") 作る 24 文字サイズ。

## 「四角形を描く」プログラム (25分頃)

```
vy=0 dy=-0.7
かめた=タートル! 作る。
かめた! 500 歩く 500 戻る 図形を作る。
かめた! 90 左回り 200 歩く (青) 線の色。
かめた:衝突:vy=-0.9*vy
タイマー! 作る 200 回数「
  vy=vy+dy
  かめたvy移動する。
」実行。
```

## 第11回の内容

### サクセスの実習 (9分頃)

サクセスは「[sAccess: データベース実習支援ツール](#)」から実行できます。

最初の画面で、左上の「クラスID」に「ouj (半角の英字小文字)」を入れて「クラス変更」をクリックしてください。次に、「氏名 ニックネーム (任意)」に名前を入れてください。ニックネームや架空の名前でも構いません。次に、「プリセットDB」に「コンビニ\*」が表示されていることを確認して「プリセットDB選択」をクリックしてください。

実習画面が表示されたら、左側の「操作コマンドを追加しよう」という部分にコマンドを書いて「追加」を押すと、「操作コマンド列」にコマンドが追加され、右側に実行した結果の表が表示されます。

```
表示 売上データ
選択 時間帯 夕方
結合 商品データ
射影 商品名、時間帯、年齢層
```

```
表示 売上データ
数える 時間帯
```

### SQLの実習 (16分頃)

SQLは「[SQLエディタ](#)」から実行できます。

最初の画面で、左上の「クラスID」に「ouj (半角の英字小文字)」を入れて「クラス変更」をクリックしてください。次に、「ユーザID」に名前を入れてください。ニックネームや架空の名前でも構いません。次に、「ログイン」をクリックしてください。

実習画面が表示されたら「SQL実行」という部分にSQLの命令を書いて「送信」を押すと、「履歴表示」に命令が追加され、画面の下に実行した結果の表が表示されます。

```
use コンビニ
```



```
select * from 商品データ
```

```
use コンビニ  
select * from 商品データ where 価格=100
```

```
use コンビニ  
select * from 商品データ where 価格>=150
```

```
use コンビニ  
select * from 商品データ where 商品名 like 'ミルクティー'
```

```
use コンビニ  
select * from 商品データ where 商品名 like '%ミルクティー'
```

```
use コンビニ  
select * from 商品データ where 商品名 like '%茶%'
```

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=1528271471>

Last update: **2018/06/06 16:51**

