

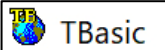
# 海洋情報技術 [プログラミング]

## 指導書

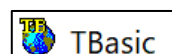
### ソフトウェアの準備

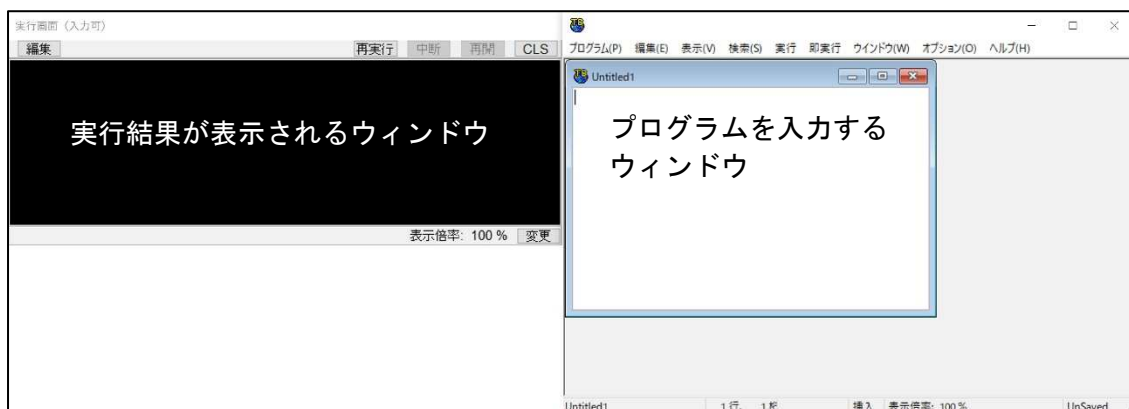
TinyBasic ソフトウェアのダウンロード先

<https://tbasic.org/>

ファイルをダウンロードし、展開したら、実行ファイル  を生徒の実習用フォルダやUSBメモリにコピーして使用する。

### ソフトウェアの起動

 をダブルクリックすると、ソフトウェアが起動する。



### プログラムの入力と実行

プログラム入力ウィンドウにプログラムを入力して、  
[実行]メニューから[プログラムの実行]を選ぶ、または[F9]キーを押すことで、  
プログラムが実行される。





### フォントのインストール

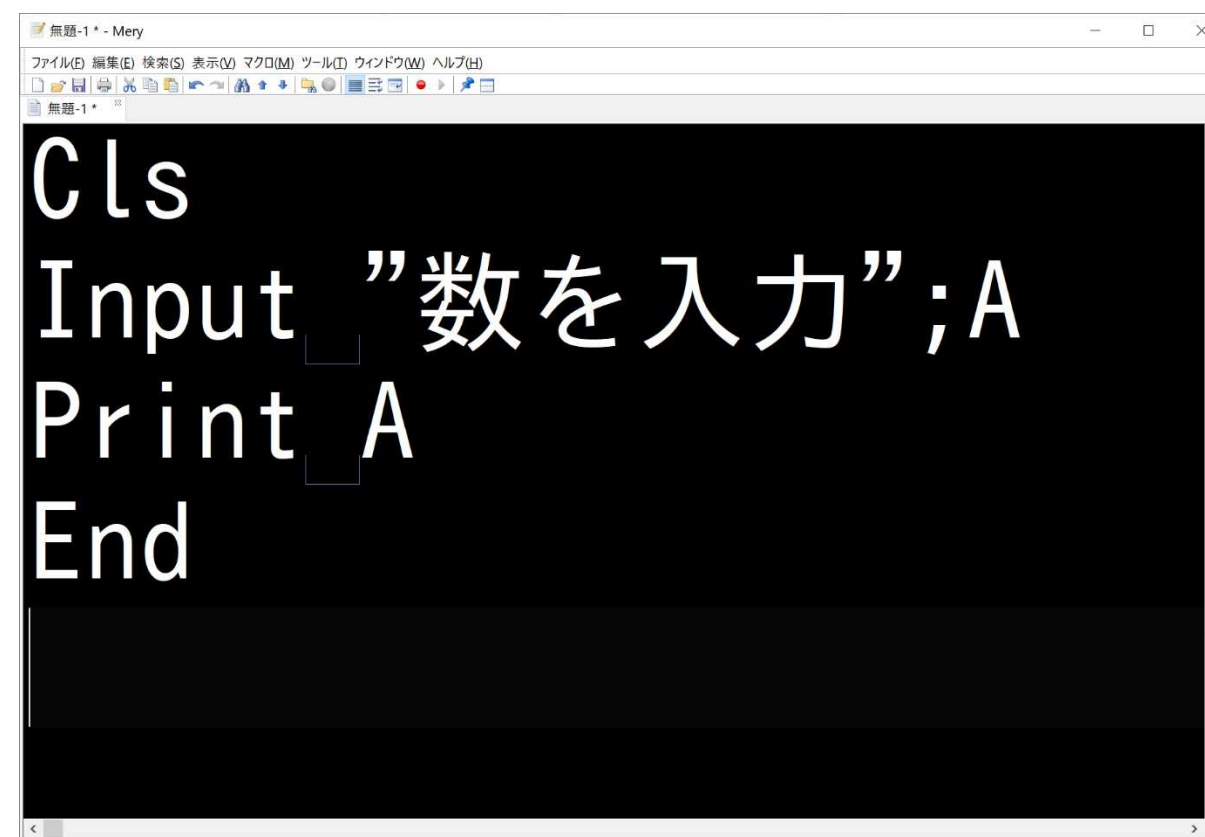
プログラムは通常フォントでは、区別が付きにくい文字があるため、プログラミング用のフォントをインストールして使用すると便利です。


Century 0 0 1 1 1	MS ゴシック 0 0 1 1 1	Takao ゴシック 0 0 1 1 1	Ricty(リクティ) 0 0 1 1 1
----------------------	----------------------	-------------------------	--------------------------

### プログラムの投影

Windows のメモ帳  やエディター  を使用して、フォントサイズを

大きくし、プロジェクターに投影すると便利です。



添付のエディターは  [F11]キーを押すと全画面表示に切り替わります。

# 水産海洋技術〔プログラミング〕

## 1 時間目

### Step01 文字を表示するプログラム

```
Cls
Print "HELLO"
```

※プログラムは、上から1つずつ実行されていきます。  
 1行目の Cls (Clear Screen) は、画面を消去する命令、  
 2行目の Print は、文字や数値を表示する命令です。

### Step02 変数を使用して、値を表示するプログラム

```
Cls
A=35
Print A
```

※変数とは、数値や文字を記憶することができる入れ物です。  
 ※変数名はアルファベットを先頭とした数字との組み合わせなら何でも構いません。  
 2行目の A=35 で、数値 35 を変数 A に代入しています。  
 3行目の Print A で、変数 A の値を表示しています。  
 ※Print "A" とすると、A という文字そのものが表示され、  
 Print A とすると、変数 A に代入されている値が表示されます。

### Step03 計算をして、結果を表示するプログラム

[時間調整] 時間がない場合は行わなくてもよい。

```
Cls
A=15
B=10
Print A+B
```

※プログラムで四則演算を行う場合、+ - × ÷ が、それぞれ + - \* / となります。

### Step04 プログラム実行後に値を入力させるプログラム

```
Cls
Input "値を入力してください", A
Print A
```

※2行目の Input が、入力させる命令です。  
 Input "表示させる文字", 変数

プログラムの解説と実行確認 (約30分)

### 練習問題

問題1 底辺と高さを Input 命令で入力して、三角形の面積を表示してみよう。

```
Cls
Input "底辺を入力してください", A      ... 底辺
Input "高さを入力してください", B      ... 高さ
Print A*B/2                             ... 面積の計算
```

※できるだけ、いきなり答えを言わないで、ヒントで解答に導くとよいです。  
 ※Step03 を行っていない場合は、計算式の説明をしてください。

練習問題 (約5分)

### 評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
1	BASIC プログラムの入力と実行 ①文字の表示 ②変数の使用 ③計算 ④値の入力	①プログラムの入力と実行ができる。 ②変数と計算式が理解できる。 ③Input 命令を理解し使用できる。	①目的のプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

### 評価シート (例)

番号	氏名	知識・技術									思考・判断・表現			主体的に学習に取り組む態度		
		①プログラムの入力と実行ができる。			②変数と計算式が理解できる。			③Input 命令を理解し使用できる。			①目的のプログラムを自ら考え表現(作成)することができる。			①プログラミングに主体的に取り組もうとしている。		
		A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
1	愛知 太郎		A			A			A			A			A	
2	石川 花子		A			B			B			C			B	
3	岩手 次郎		A			A			B			B			A	

A:よくできる B:できる C:あまりできない

# 水産海洋技術 [プログラミング]

## 2 時間目

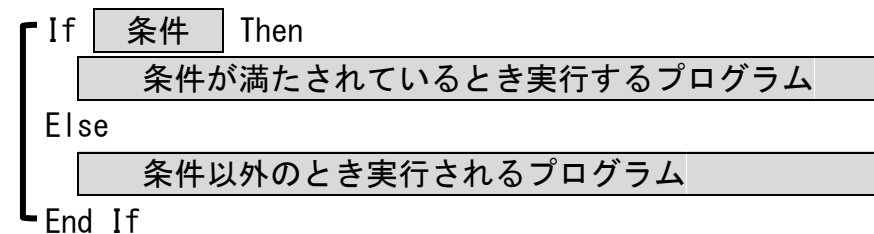
### Step01 前回の確認

```
Cls
Input "値を入力してください", A
Print A
```

### Step02 条件によって判断させるプログラム

```
Cls
Input "点数を入力してください", A
If A<30 Then
    Print "赤点"
Else
    Print "合格"
End if
```

※点数が 30 未満のとき "赤点" と表示し、それ以外のとき "合格" と表示する。



※If の終わりは、必ず End If で締めます。

※Else の処理がない場合は省略できます。

### 練習問題

問題 1 数値を入力して、50 以上なら "A 班"、それ以外なら "B 班" と表示してみよう。

```
Cls
Input "数値を入力してください", A
If A>=50 Then
    Print "A 班"
Else
    Print "B 班"
End If
```

プログラムの解説と実行確認 (約 20 分)

練習問題

問題 2 数値を入力して、100 以下のときだけ "範囲内" と表示してみよう。

```
Cls
Input "数値を入力してください", A
If A<=100 Then
    Print "範囲内"
End If
```

問題 3 2つの値、A と B を Input 命令で入力し、大きいほうの値を表示してみよう。

```
Cls
Input "値 A を入力してください", A
Input "値 B を入力してください", B
If A>B Then
    Print A
Else
    Print B
End If
```

※生徒間で進行に差が出やすいため、全員ができる到達目標を定めて、それ以上はよくできる生徒の応用問題とするとよいです。

応用 1 カレンダーの月を考えたとき、月の数を入力して、1 より小さければ "小さいです"、12 より大きければ "大きいです" と表示してみよう。

```
Cls
Input "月を入力してください", Tuki
If Tuki<1 Then
    Print "小さいです"
End If
If Tuki>12 Then
    Print "大きいです"
End If
```

応用 2 三つの数値を入力して、一番大きい数値を表示してみよう。

```
Cls
Input "数値 1 を入力してください", A
Input "数値 2 を入力してください", B
Input "数値 3 を入力してください", C
If A>B Then
    If A>C Then
        Print A
    Else
        Print C
    End If
Else
    If B>C Then
        Print B
    Else
        Print C
    End If
End If
```

練習問題 (約 25 分)

### 評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
2	BASIC プログラムの入力と実行 ①判断命令 If	①プログラムの入力と実行ができる。 ②If 命令を理解し使用できる。	①目的のプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

# 水産海洋技術〔プログラミング〕

## 3 時間目

### Step01 前回の確認

```
Cls
Input "年齢を入力してください", Nen
If Nen >= 18 Then
    Print "成人"
Else
    Print "子供"
End if
```

### Step02 繰り返しプログラム

```
Cls
For I=1 To 10 Step 1
    Print "ABC"
Next
```

※最初に I の値を 1 とし（初期値）、10 になるまで繰り返す（終了値）

1 回繰り返すたびに 1 ずつ I の値を増やす（増分値）

※ "ABC" が 10 回表示されます。

### Step03 値を表示する

```
Cls
For I=1 To 10 Step 1
    Print I
Next
```

※ I の値を表示している。I の値の変化が見て取れる。

### Step04 While 命令

```
Cls
I=1
While I <= 10
    Print I
    I=I+1
Wend
```

※Step03 と同じプログラムを While で記述

プログラムの解説と実行確認（約20分）

### 練習問題

問題1 For ~ Next を使って、“ABC” を 50 回表示しなさい。

```
Cls
For A=1 To 50 Step 1
    Print "ABC"
Next
```

問題2 1 ~ 30 までの数値を表示しなさい。

```
Cls
For A=1 To 30 Step 1
    Print A
Next
```

```
Cls
A=1
While A <= 30
    Print A
    A=A+1
Wend
```

問題3 1 ~ 100 までの奇数を表示しなさい。

```
Cls
For A=1 To 100 Step 2
    Print A
Next
```

```
Cls
A=1
While A <= 100
    Print A
    A=A+2
Wend
```

問題4 1 ~ 100 までの偶数を表示しなさい。

```
Cls
For A=2 To 100 Step 2
    Print A
Next
```

```
Cls
A=2
While A <= 100
    Print A
    A=A+2
Wend
```

応用1 数値を入力して、その数値の段の九九を表示しなさい。

応用2 1 ~ 100 までの合計を計算して表示しなさい。

```
Cls
Input "段は?"; Dan
For A=1 To 9 Step 1
    Print Dan*A
Next
```

```
Cls
Kei=0
For A=1 To 100 Step 1
    Kei=Kei+A
Next
Print Kei
```

練習問題（約25分）

### 評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
3	BASICプログラムの入力と実行 ①繰り返し命令 For ②繰り返し命令 While	①For 命令を理解し使用できる。 ②While 命令を理解し使用できる。	①For と While の違いを理解し使い分けられる。 ②目的のプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

# 水産海洋技術〔プログラミング〕

## 4 時間目

### Step01 前回の確認

```
Cls
For I=1 To 10 Step 1
  Print I
Next
```

```
Cls
I=1
While I<=10
  Print I
  I=I+1
Wend
```

### Step02 ランダム値

```
Cls
Print Rnd
```

※0以上1未満の適当な数値（ランダム値）が表示されます。

※ゲーム等の作成には必要な値で、Rnd を計算によって目的の値に変化させます。

例	計算式	得られる値
元の式	R=Rnd	0 ~ 0.9999999
10倍する	R=Rnd*10	0 ~ 9.999999
少数以下を切り捨てる	R=Int(Rnd*10)	0 ~ 9
+1する	R=Int(Rnd*10)+1	1 ~ 10

※ 式  $R = \text{Int}(\text{Rnd} * 10) + 1$  により 1~10までのランダムが得られる。

※ 式中の \*10 を変えることで、1~目的の値までのランダムに変えられる。

※ このままでは、毎回ランダムな数値が同じ順番で出るので、プログラムの最初にランダムの初期化 Randomize を入れると良い。

### Step03 サイコロを10回出力

```
Randomize
Cls
For I=1 To 10 Step 1
  Print Int(Rnd*6)+1
Next
```

プログラムの解説と実行確認 (約20分)

### 練習問題

問題1 1~3のランダムな整数が表示されるプログラムを作成しなさい。

```
Randomize
Cls
Print Int(Rnd*3)+1
```

問題2 1~3までのランダムな整数を発生させ、値が1であれば”ゲー”、2であれば”パー”、3であれば”チョコキ”と表示しなさい。

```
Randomize
Cls
R=Int(Rnd*3)+1
If R=1 Then
  Print "ゲー"
End If
If R=2 Then
  Print "パー"
End If
If R=3 Then
  Print "チョコキ"
End If
```

応用1 サイコロを1000回振って、1が出た回数を数えてみよう。

```
Randomize
Cls
Kai=0
For A=1 To 1000 Step 1
  R=Int(Rnd*6)+1
  If R=1 Then
    Kai=Kai+1
  End If
Next
Print "1の回数は";Kai;"回です。"
```

練習問題 (約25分)

### 評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
4	BASICプログラムの入力と実行 ①ランダム Rnd	①Rnd 命令を理解し使用できる。	①計算によって目的の値を導き出せる。 ②目的のプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

# 水産海洋技術〔グラフィックプログラミング〕

## 5 時間目

### Step01 前回の確認

```
Randomize
Cls
For I=1 To 10 Step 1
  Print Int(Rnd*6)+1
Next
```

### Step02 グラフィック画面の設定

```
GScreen(600,400)
Cls 2
```

※新しいグラフィックを表示するためのウィンドウが表示される。  
 ※ウィンドウサイズは 横 600 ドット×縦 400 ドット  
 ※Cls 2 でグラフィック画面を消去

### Step03 直線と四角形を描く

```
GScreen(600,400)
Cls 2
Line(0,0)-(300,200),0
```

※座標(0,0)から(300,200)まで、0番の色(黒)で線を引いている。

```
GScreen(600,400)
Cls 2
Line(0,0)-(300,200),0,B
```

※最後に ,B を付けると四角形となる。(Box)

```
GScreen(600,400)
Cls 2
Line(0,0)-(300,200),0,BF
```

※最後に ,BF を付けると塗りつぶし四角形となる。(Box Fill)

色番号

色番号	名前	色番号	名前	色番号	名前	色番号	名前
0	Black	1	Navy	2	Green	3	Teal
4	Maroon	5	Purple	6	Olive	7	Silver
8	Gray	9	Blue	10	Lime	11	Cyan
12	Red	13	Fuchsia	14	Yellow	15	White

プログラムの解説と実行確認

### Step04 円を描く

```
GScreen(600,400)
Cls 2
Circle(300,200),100,0
```

※座標(300,200)を中心点とする半径 100 の円を 色番号 0 (黒)で描く。

```
GScreen(600,400)
Cls 2
Circle(300,200),100,0
```

※座標(300,200)を中心点とする半径 100 の円を 色番号 0 (黒)で描く。

※Circle 命令では開始角度、終了角度を指定して円弧、比率を変えることで楕円を描くこともできる。

Circle(横座標, 縦座標), 半径, 色番号, 開始角度, 終了角度, 比率, [F]

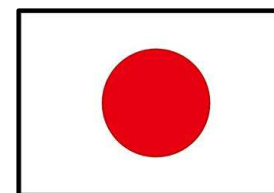
```
GScreen(600,400)
Cls 2
Circle(300,200),100,0,,,F
```

※色番号 0 (黒)の後に , を 4 つ付けて F とすると塗りつぶしもできる。

#### 練習問題

問題 1 日本の国旗を描いてみよう。

```
GScreen(600,400)
Cls 2
Line(0,0)-(600,400),15,BF
Circle(300,200),100,12,,,F
```



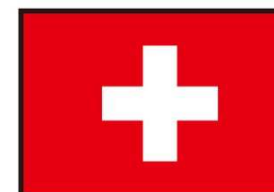
問題 2 フランス国旗を描いてみよう。

```
GScreen(600,400)
Cls 2
Line(0,0)-(200,400),9,BF
Line(201,0)-(400,400),15,BF
Line(401,0)-(600,400),12,BF
```



応用 1 スイス国旗を描いてみよう。

```
GScreen(600,400)
Cls 2
Line(0,0)-(600,400),12,BF
Line(250,50)-(350,350),15,BF
Line(150,150)-(450,250),15,BF
```



プログラムの解説と実行確認 (約 20 分)

練習問題 (約 25 分)

#### 評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
5	BASIC プログラムの入力と実行 ①グラフィック命令 直線・四角形・円の描画	①各命令を理解し使用できる。 ②座標を理解している。	①目的の図形を描くプログラムを自ら考え表現することができる。	①プログラミングに主体的に取り組もうとしている。

# 水産海洋技術〔グラフィックプログラミング〕

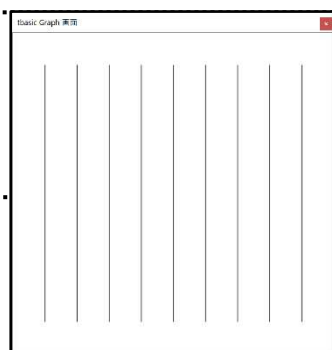
## 6 時間目

### Step01 前回の確認

```
GScreen(600, 400)
Cls 2
Line(0, 0)-(300, 200), 12
Line(300, 200)-(350, 250), 9, B
Line(350, 250)-(400, 300), 2, BF
Circle(300, 200), 150, 14, , , , F
```

### Step02 繰り返し命令 For を使って直線を描く

```
GScreen(500, 500)
Cls 2
For X=50 To 450 Step 50
  Line(X, 50)-(X, 450), 0
Next
```

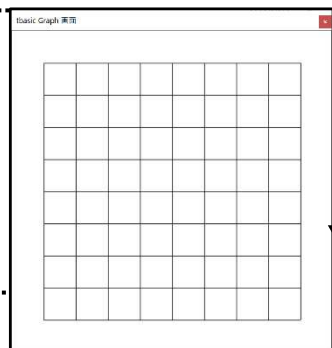


※横軸を 50 から 450 まで 50 間隔でずらして描いている。  
 ※Step の間隔を変えることで、線の幅が変わります。

#### 練習問題

問題 1 続けて横線を描いてマス目にしてみよう。

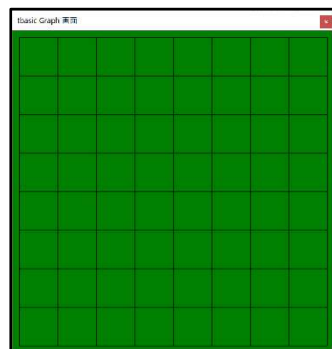
```
GScreen(500, 500)
Cls 2
For X=50 To 450 Step 50
  Line(X, 50)-(X, 450), 0
Next
For Y=50 To 450 Step 50
  Line(50, Y)-(450, Y), 0
Next
```



#### 練習問題

問題 1 背景を色番号 2 (緑) にし、黒色の線 (色番号 0) で 8 x 8 のオセロ盤を描いてみよう。ただし、四隅の余白は 10 ドットとする。

```
GScreen(500, 500)
Cls 2
Line(0, 0)-(500, 500), 2, bf
For X=10 To 490 Step 60
  Line(X, 10)-(X, 490), 0
Next
For Y=10 To 490 Step 60
  Line(10, Y)-(490, Y), 0
Next
```

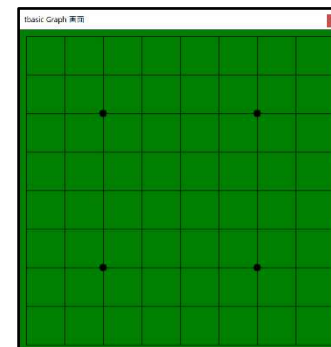


プログラムの解説と実行確認 (約 20 分)

練習問題

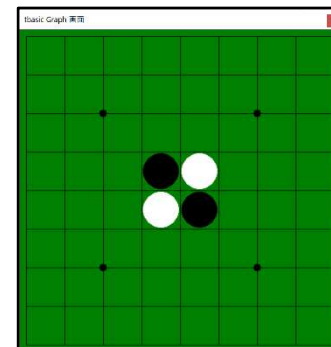
問題 2 オセロ盤の中央 4 x 4 マスの四隅に、黒い丸を描いてみよう。

```
GScreen(500, 500)
Cls 2
Line(0, 0)-(500, 500), 2, bf
For X=10 To 490 Step 60
  Line(X, 10)-(X, 490), 0
Next
For Y=10 To 490 Step 60
  Line(10, Y)-(490, Y), 0
Next
Circle(130, 130), 6, 0, , , , F
Circle(370, 130), 6, 0, , , , F
Circle(130, 370), 6, 0, , , , F
Circle(370, 370), 6, 0, , , , F
```



問題 3 オセロ盤に最初の駒を描いてみよう。

```
GScreen(500, 500)
Cls 2
Line(0, 0)-(500, 500), 2, bf
For X=10 To 490 Step 60
  Line(X, 10)-(X, 490), 0
Next
For Y=10 To 490 Step 60
  Line(10, Y)-(490, Y), 0
Next
Circle(130, 130), 6, 0, , , , F
Circle(370, 130), 6, 0, , , , F
Circle(130, 370), 6, 0, , , , F
Circle(370, 370), 6, 0, , , , F
Circle(220, 220), 28, 0, , , , F
Circle(280, 220), 28, 15, , , , F
Circle(220, 280), 28, 15, , , , F
Circle(280, 280), 28, 0, , , , F
```



応用 1 オセロゲームを進めてみよう。

練習問題 (約 25 分)

### 評価のポイント

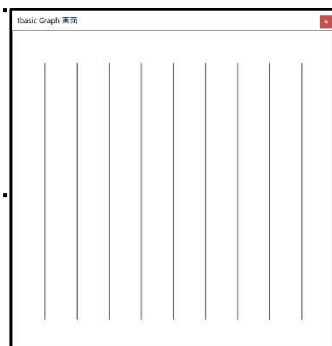
時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
6	BASIC プログラムの入力と実行 ①グラフィック命令 直線・四角形・円を使って オセロ盤の描画	①各命令を理解し使用できる。 ②座標を理解している。	①オセロ盤を描くために、 各命令を組み合わせてプログラムを自ら考え表現することができる。	①プログラミングに主体的に取り組もうとしている。

# 水産海洋技術〔グラフィックプログラミング〕

7 時間目

## Step01 前回の確認

```
GScreen(500, 500)
Cls 2
For X=50 To 450 Step 50
  Line(X, 50)-(X, 450), 0
Next
```



プログラムの解説と実行確認(約10分)

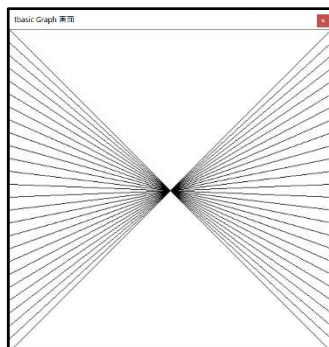
## 練習問題

問題1 次のような線を、繰り返し命令 For を使って描きなさい。  
ただし、線の間隔は 20 ドットとする。



```
GScreen(500, 500)
Cls 2
For Y=0 To 500 Step 20
  Line(0, Y)-(500, Y), 0
Next
```

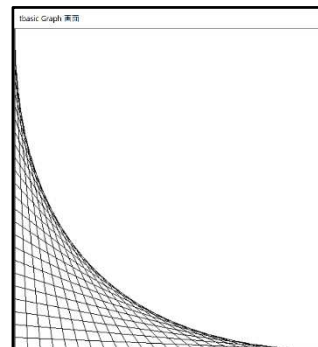
問題2 次のような線を、繰り返し命令 For を使って描きなさい。



```
GScreen(500, 500)
Cls 2
For Y=0 To 500 Step 20
  Line(0, Y)-(500, 500-Y), 0
Next
```

※始点と終点の座標がどのように動いていくかを説明して生徒に考えさせるとよいです。

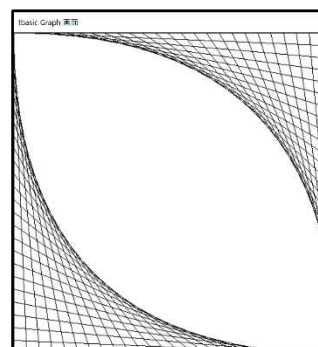
問題3 次のような線を、繰り返し命令 For を使って描きなさい。



```
GScreen(500, 500)
Cls 2
For Y=0 To 500 Step 20
  Line(0, Y)-(Y, 500), 0
Next
```

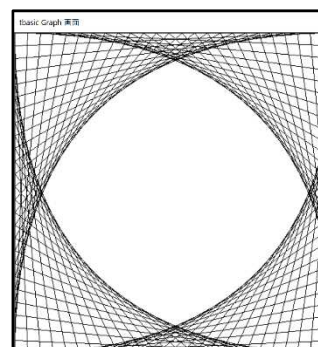
※始点と終点の座標がどのように動いていくかを説明して生徒に考えさせるとよいです。

応用1 次のような線を、繰り返し命令 For を使って描きなさい。



```
GScreen(500, 500)
Cls 2
For Y=0 To 500 Step 20
  Line(0, Y)-(Y, 500), 0
Next
For Y=0 To 500 Step 20
  Line(Y, 0)-(500, Y), 0
Next
```

応用2 次のような線を、繰り返し命令 For を使って描きなさい。



```
GScreen(500, 500)
Cls 2
For Y=0 To 500 Step 20
  Line(0, Y)-(Y, 500), 0
Next
For Y=0 To 500 Step 20
  Line(Y, 0)-(500, Y), 0
Next
For Y=0 To 500 Step 20
  Line(0, Y)-(500-Y, 0), 0
Next
For Y=0 To 500 Step 20
  Line(Y, 500)-(500, 500-Y), 0
Next
```

練習問題(約35分)

## 評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
7	BASIC プログラムの入力と実行 ①グラフィック命令 繰り返し命令を使って、様々な直線を描く	①繰り返し命令を理解し使用できる。 ②座標を理解している。	①図形を描くために、プログラムを自ら考え表現することができる。	①プログラミングに主体的に取り組もうとしている。

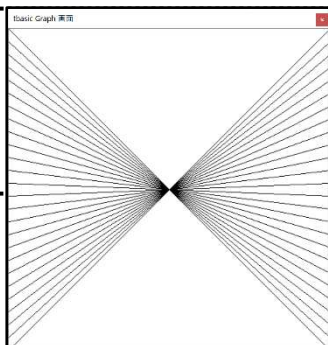


# 水産海洋技術 [プログラミング]

## 8 時間目

### Step01 前回の確認

```
GScreen(500,500)
Cls
For Y=0 To 500 Step 20
  Line(0,Y)-(500,500-Y),0
Next
```



### Step02 Read と Data

```
Cls
Read A
Print A
Data 100
```

※Read 命令は、Data の先頭から順番に値を読み込む命令である。

```
Cls
Read A,B,C
Print A;B;C
Data 100,200,300
```

```
Cls
For I=1 To 8 Step 1
  Read A
  Print A
Next
Data 50,30,75,90,27,10,100,60
```

※繰り返し命令 For を使って、連続して読み込むこともできる。

```
Cls
For I=1 To 8 Step 1
  Read A
  Print "点数は";A;"点"
Next
Data 50,30,75,90,27,10,100,60
```

※Print 命令に ; を付けると、表示を連続させることができる。

プログラムの解説と実行確認

```
Cls
MX=0
For I=1 To 8 Step 1
  Read A
  Print "点数は";A;"点"
  If MX<=A Then
    MX=A
  End If
Next
Print "最高点は";MX;"点"
Data 50,30,75,90,27,10,100,60
```

※最初に、変数 MX に最も小さい値を入れておく。  
Read 命令で値を読むごとに、MX と比較して、読み込んだ値の方が大きければ、その値を MX に代入する。これを繰り返すことによって、最大値を MX に記憶させることができる。

### 練習問題

問題 1 15 人分のテストの点数を Data に入力し、Read 命令で表示しなさい。

```
Cls
MX=0
For I=1 To 15 Step 1
  Read A
  Print "点数は";A;"点"
Next
Data 50,30,75,90,27,10,100,60,49,84,61,18,50,97,77
```

問題 2 最低点を表示しなさい。

```
Cls
MN=999
For I=1 To 15 Step 1
  Read A
  Print "点数は";A;"点"
  If MN>=A Then
    MN=A
  End If
Next
Print "最低点は";MN;"点"
Data 50,30,75,90,27,10,100,60,49,84,61,18,50,97,77
```

プログラムの解説と実行確認 (約 3 分)

練習問題 (約 15 分)

### 評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
8	BASIC プログラムの入力と実行 ①Read・Data 命令 Data から値を読み込み処理を行う	①Read 命令を理解し使用することができる。 ②繰り返し命令 For、判断命令 If と組み合わせて使用することができる。	①目的のプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

# 水産海洋技術〔プログラミング〕

## 9 時間目

### Step01 前回の確認

```
Cls
For I=1 To 5 Step 1
  Read A
  Print A
Next
Data 50, 30, 75, 90, 27
```

### Step02 一次元配列

```
Cls
Dim A(5)
A(1)=50
A(2)=30
A(3)=75
A(4)=90
A(5)=27
Print A(3)
```

※Dim を使用して配列を使用することを宣言する。  
変数 A の 1 番目 A(1) に 50 を代入、A(2) に 30 を代入、A(3) に 75 を代入、A(4) に 90 を代入、A(5) に 27 を代入して、最後に A(3) を表示している。  
このように、変数名に添字を付けて使用する変数を配列という。

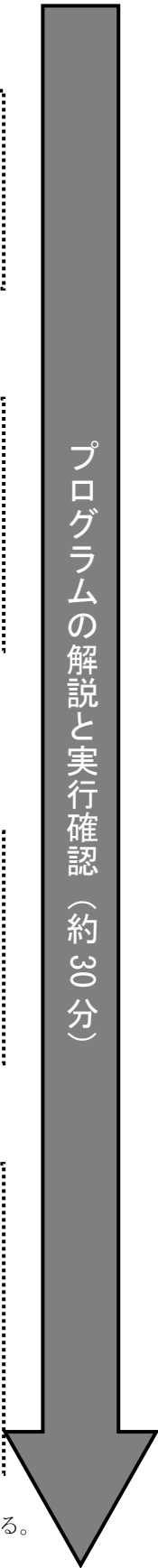
```
Cls
Dim A(5)
For I=1 To 5 Step 1
  Read A(I)
Next
Data 50, 30, 75, 90, 27
Print A(3)
```

※繰り返し命令 For と組み合わせることによって、配列に一気に値を入れることができる。

```
Cls
Dim A$(5)
For I=1 To 5
  Read A$(I)
Next
Data "A", "B", "C", "D", "E"

For I=5 To 1 Step -1
  Print A$(I)
Next
```

※文字を配列 A\$ に記憶し、それを逆順で表示している。文字を記憶するときの変数名には \$ を付ける。



### 練習問題

問題 1 8 個の数値を配列 A に読み込みなさい。

```
Cls
Dim A(8)
For I=1 To 8 Step 1
  Read A(I)
Next
Data 50, 30, 75, 90, 27, 100, 40, 68
```

問題 2 続けて、読み込んだ数値をすべて表示しなさい。

```
Cls
Dim A(8)
For I=1 To 8 Step 1
  Read A(I)
Next
Data 50, 30, 75, 90, 27, 100, 40, 68

For I=1 To 8 Step 1
  Print A(I)
Next
```

応用 1 読み込んだ数値の基数番目を表示しなさい。また偶数番目も表示しなさい。

```
Cls
Dim A(8)
For I=1 To 8 Step 1
  Read A(I)
Next
Data 50, 30, 75, 90, 27, 100, 40, 68

For I=1 To 8 Step 2
  Print A(I)
Next

Cls
Dim A(8)
For I=1 To 8 Step 1
  Read A(I)
Next
Data 50, 30, 75, 90, 27, 100, 40, 68

For I=2 To 8 Step 2
  Print A(I)
Next
```

応用 2 読み込んだ数値を逆順にすべて表示しなさい。

```
Cls
Dim A(8)
For I=1 To 8 Step 1
  Read A(I)
Next
Data 50, 30, 75, 90, 27, 100, 40, 68

For I=8 To 1 Step -1
  Print A(I)
Next
```



### 評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
9	BASIC プログラムの入力と実行 ①一次元配列 配列を使った処理を行う	①Read 命令を理解し配列に値を読み込むことができる。 ②繰り返し命令を組み合わせ使用することができる。	①目的のプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

# 水産海洋技術 [プログラミング]

10 時間目

## Step01 Read・Dataの確認

```
Cls
For I=1 To 5 Step 1
  Read A
  Print A
Next
Data 50, 30, 75, 90, 27
```

## Step02 二重ループ

```
Cls
For Y=1 To 4 Step 1
  For X=1 To 4 Step 1
    Read A
    Print A;
  Next
  Print
Next
Data 1, 2, 3, 4
Data 5, 6, 7, 8
Data 9, 10, 11, 12
Data 13, 14, 15, 16
```

※二重ループの動作を確認します。  
 ※内ループのPrintは ; を付けて、横につなげていますが、内ループを抜けた段階で、  
 ; を付けずに改行しています。

## Step03 掛け算九九

```
Cls
For Y=1 To 9 Step 1
  For X=1 To 9 Step 1
    Print X*Y;
  Next
  Print
Next
```

プログラムの解説と実行確認

## Step04 キャラクターの作成

```
Cls
For Y=1 To 4 Step 1
  For X=1 To 4 Step 1
    Read C
    If C=1 Then
      Print "■";
    Else
      Print " ";
    End If
  Next
  Print
Next
Data 1, 1, 1, 1
Data 1, 1, 0, 1
Data 1, 0, 1, 1
Data 1, 1, 1, 1
```



※Dataの値によって、表示する文字を変えている。  
 ※Printの"■"と" "は全角文字と空白です。

### 練習問題

問題1 8×8のキャラクターにしてみよう。

```
Cls
For Y=1 To 8 Step 1
  For X=1 To 8 Step 1
    Read C
    If C=1 Then
      Print "■";
    Else
      Print " ";
    End If
  Next
  Print
Next
Data 0, 1, 0, 0, 0, 0, 1, 0
Data 0, 1, 0, 0, 0, 0, 1, 0
Data 0, 1, 1, 0, 0, 1, 1, 0
Data 1, 1, 1, 1, 1, 1, 1, 1
Data 1, 0, 0, 1, 1, 0, 0, 1
Data 1, 0, 1, 1, 1, 1, 0, 1
Data 1, 1, 1, 1, 1, 1, 1, 1
Data 0, 1, 1, 1, 1, 1, 1, 0
```

問題2 キャラクターを自由に変えてみよう。

### 評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
10	BASICプログラムの入力と実行 ①二重ループ 二重ループを使った処理を行う	①二重ループを理解し、プログラミングすることができる。	①二重ループの動作を理解し、目的の結果を得るためのプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

プログラムの解説と実行確認 (約3分)

練習問題 (約5分)

# 水産海洋技術 [グラフィックプログラミング]

## 11 時間目

### Step01 キャラクター表示をグラフィックス化する

```
GScreen(500,500)
Cls 2
For Y=1 To 8 Step 1
  For X=1 To 8 Step 1
    Read C
    If C=1 Then
      Line((X-1)*20, (Y-1)*20)-(X*20, Y*20), 0, BF
    End If
  Next
Next
Data 0,1,0,0,0,0,1,0
Data 0,1,0,0,0,0,1,0
Data 0,1,1,0,0,1,1,0
Data 1,1,1,1,1,1,1,1
Data 1,0,0,1,1,0,0,1
Data 1,0,1,1,1,1,0,1
Data 1,1,1,1,1,1,1,1
Data 0,1,1,1,1,1,1,0
```



※Read C で読み込んだData の値が 1 ならば、Print "■"ではなく、Line 命令によって、縦横 20 ドットの塗りつぶし四角形を描いている。

※色 の部分を黒色の 0 ではなく、C とすることで、Data 内の値を色番号とすることができる。

### Step02 色付きのキャラクターを表示する

```
GScreen(500,500)
Cls 2
For Y=1 To 8 Step 1
  For X=1 To 8 Step 1
    Read C
    Line((X-1)*20, (Y-1)*20)-(X*20, Y*20), C, BF
  Next
Next
Data 15,15,12,12,12,12,12,12
Data 15,9,6,9,6,14,14,15
Data 6,15,6,6,6,6,6,15
Data 8,15,6,14,6,6,15,6
Data 8,12,12,12,12,12,12,8
Data 15,15,12,12,12,12,15,15
Data 15,15,8,8,8,8,15,15
Data 15,8,8,15,15,8,8,15
```

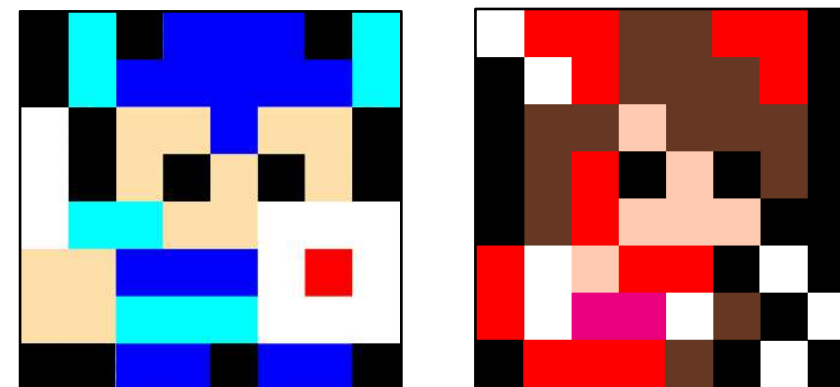
プログラムの解説と実行確認 (約 30 分)

色番号

色番号	名前	色番号	名前	色番号	名前	色番号	名前
0	Black	1	Navy	2	Green	3	Teal
4	Maroon	5	Purple	6	Olive	7	Silver
8	Gray	9	Blue	10	Lime	11	Cyan
12	Red	13	Fuchsia	14	Yellow	15	White

### 練習問題

問題 1 好きなキャラクターを描いてみよう。



練習問題 (約 15 分)

### 評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
11	BASIC プログラムの入力と実行 ①二重ループ 二重ループを使ってキャラクターの描画	①二重ループを理解し、プログラミングすることができる。	①二重ループの動作を理解し、キャラクターを描画するプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

# 水産海洋技術〔グラフィックプログラミング〕

## 12 時間目

### Step01 データを工夫して処理しやすくする

```
GScreen(500,500)
Cls 2
For Y=1 To 8 Step 1
  Read C$ ← 1行分の文字列として読み込む
  For X=1 To 8 Step 1
    C=Val("&H"+Mid$(C$,X,1))
    Line((X-1)*20,(Y-1)*20)-(X*20,Y*20),C,BF
  Next
Next
Data "FFCCCCC"
Data "F9696EEF"
Data "6F66666F"
Data "8F6E66F6"
Data "8CCCCC8"
Data "FFCCCCFF"
Data "FF8888FF"
Data "F88FF88F"
```

色データを16進数1桁にする。  
データが、数値から文字になるため" "で囲う

→ C = Val("&H" + Mid\$(C\$, X, 1))  
文字列 C\$の X 番目から1文字抜き出している。  
Val("&H" は16進を10進にするため。

※少し難易度が高いため、「このようにすれば、16進1桁を番号として、キャラクターが描ける。」程度の説明が良い。

### 練習問題

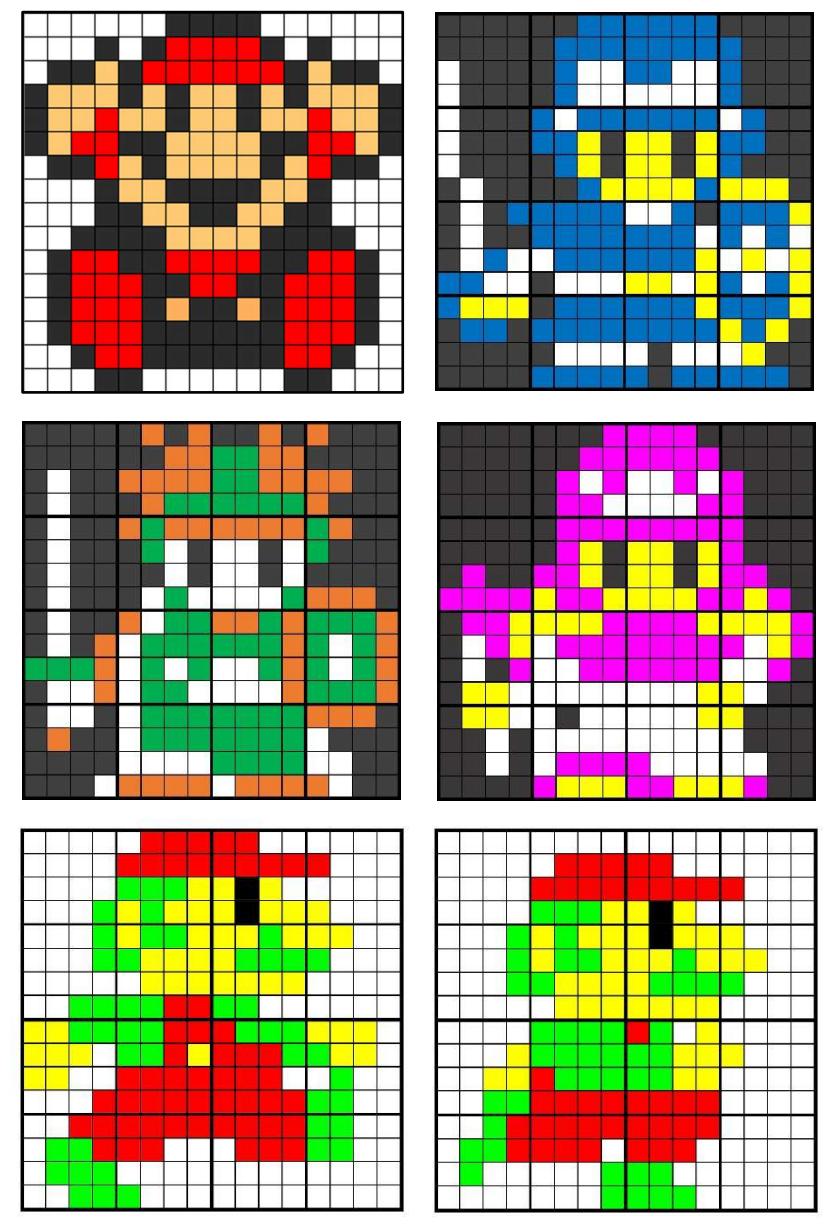
問題1 キャラクターサイズを16×16にしてみよう。

```
GScreen(500,500)
Cls 2
For Y=1 To 16 Step 1
  Read C$
  For X=1 To 16 Step 1
    C=Val("&H"+Mid$(C$,X,1))
    Line((X-1)*20,(Y-1)*20)-(X*20,Y*20),C,BF
  Next
Next
Data "FFFFFFFFFFFFFFFF"
Data "F00000000000000F"
Data "F00000000000000F"
Data "F00000000000000F"
Data "F00000000000000F"
Data "F00000000000000F"
Data "F00000000000000F"
Data "F00000000000000F"
Data "F00000000000000F"
Data "F00000000000000F"
Data "F00000000000000F"
Data "F00000000000000F"
Data "F00000000000000F"
Data "F00000000000000F"
Data "F00000000000000F"
Data "FFFFFFFFFFFFFFFF"
```

プログラムの解説と実行確認 (約20分)

プログラムの解説と実行確認

問題2 好きなキャラクターを描いてみよう。



練習問題 (約25分)

### 評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
12	BASICプログラムの入力と実行 ①二重ループ 二重ループを使ってキャラクターの描画	①二重ループを理解し、プログラミングすることができる。	①二重ループの動作を理解し、キャラクターを描画するプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

# 水産海洋技術〔グラフィックプログラミング〕

## 13 時間目

### Step01 ランダムな位置に円を描く

```
GScreen(500,500)
Cls 2
Randomize

X=Int(Rnd*500)
Y=Int(Rnd*500)
Circle(X,Y),30,0
```

ランダムな位置に円を描いている。

### Step02 サブルーチン

```
GScreen(500,500)
Cls 2
Randomize
Call MARU() ← Call 命令で、サブルーチン呼び出す
End

Sub MARU()
X=Int(Rnd*500)
Y=Int(Rnd*500)
Circle(X,Y),30,0
End Sub
```

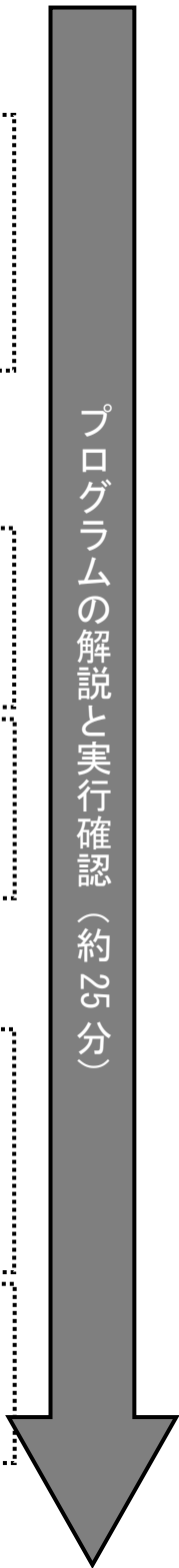
ランダムな位置に円を描くサブルーチン

※独立したプログラムのかたまりを、サブルーチンという。  
 ※Call 命令で、サブルーチン呼び出すことができる。

```
GScreen(500,500)
Cls 2
Randomize
For I=1 To 10 Step 1
    Call MARU()
Next
End

Sub MARU()
X=Int(Rnd*500)
Y=Int(Rnd*500)
Circle(X,Y),30,0
End Sub
```

※サブルーチンは何度でも呼び出して実行することができる。



### 練習問題

問題1 ランダムな位置に、ランダムな大きさの四角形を描くプログラムを作成しなさい。

```
GScreen(500,500)
Cls 2
Randomize

X1=Int(Rnd*500)
Y1=Int(Rnd*500)
X2=Int(Rnd*500)
Y2=Int(Rnd*500)
Line(X1,Y1)-(X2,Y2),0,B
```

問題2 ランダムな大きさの四角形を描くプログラムをサブルーチン化しなさい。

```
GScreen(500,500)
Cls 2
Randomize
Call SIKAKU()
End

Sub SIKAKU()
X1=Int(Rnd*500)
Y1=Int(Rnd*500)
X2=Int(Rnd*500)
Y2=Int(Rnd*500)
Line(X1,Y1)-(X2,Y2),0,B
End Sub
```

問題3 ランダムな大きさの四角形を描くプログラムをサブルーチンを10回実行しなさい。



### 評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
13	BASIC プログラムの入力と実行 ①サブルーチン 何度も行ふ処理をサブルーチン化する	①サブルーチンを理解し、プログラミングすることができる。	①サブルーチンの動作を理解し、プログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

# 水産海洋技術〔グラフィックプログラミング〕

## 14 時間目

### Step01 前回の確認

```
GScreen(500,500)
Cls 2
Randomize
Call MARU() ← Call 命令で、サブルーチン呼び出す
End
```

```
Sub MARU()
  X=Int(Rnd*500)
  Y=Int(Rnd*500)
  Circle(X,Y),30,0
End Sub
```

ランダムな位置に円を描くサブルーチン

### Step02 円の色を指定するために、値を渡す

```
GScreen(500,500)
Cls 2
Randomize
Call MARU(12) ← カッコ内に渡す値を記述する
End
```

```
Sub MARU(C)
  X=Int(Rnd*500)
  Y=Int(Rnd*500)
  Circle(X,Y),30,C
End Sub
```

カッコ内を書いた数値を、変数 C で受け取る。

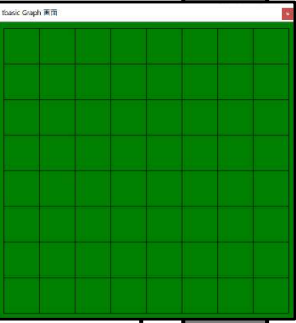
### Step03 オセロ盤を描いて、コマを表示するサブルーチンを作成する

```
GScreen(500,500)
Cls 2
Line(0,0)-(500,500),2,bf
For A=10 To 490 Step 60
  Line(A,10)-(A,490),0
  Line(10,A)-(490,A),0
Next
Call KOMA(4,4,0)
Call KOMA(5,5,0)
Call KOMA(5,4,15)
Call KOMA(4,5,15)
End
```

オセロ盤を描いている  
黒コマを置いている  
白コマを置いている

```
Sub KOMA(X,Y,C)
  Circle(X*60-20,Y*60-20),28,C,,,F
End Sub
```

コマを表示するサブルーチン  
X:横 Y:縦 C:コマの色



プログラムの解説と実行確認 (約35分)

### 練習問題

問題1 オセロを進捗させてみよう。

```
GScreen(500,500)
Cls 2
Line(0,0)-(500,500),2,bf
For A=10 To 490 Step 60
  Line(A,10)-(A,490),0
  Line(10,A)-(490,A),0
Next
Call KOMA(4,4,0)
Call KOMA(5,5,0)
Call KOMA(5,4,15)
Call KOMA(4,5,15)

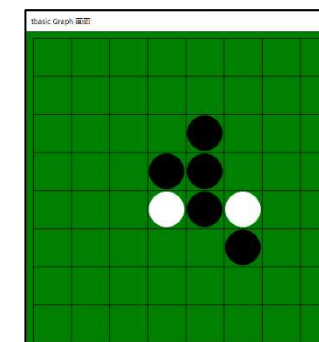
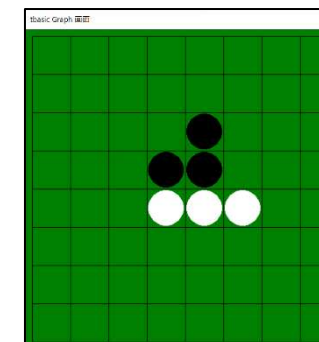
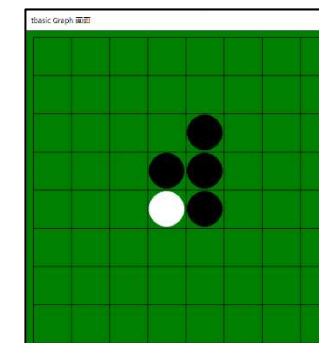
Call KOMA(5,3,0)
Call KOMA(5,4,0)

Call KOMA(6,5,15)
Call KOMA(5,5,15)

Call KOMA(6,6,0)
Call KOMA(5,5,0)

End

Sub KOMA(X,Y,C)
  Circle(X*60-20,Y*60-20),28,C,,,F
End Sub
```



練習問題 (約5分)

### 評価のポイント

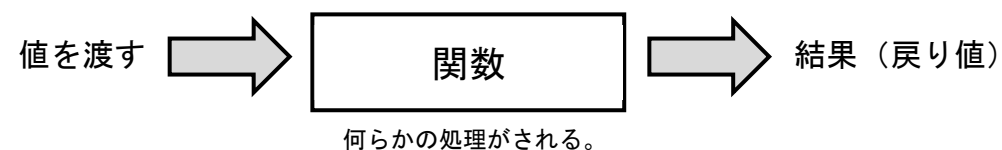
時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
14	BASICプログラムの入力と実行 ①サブルーチン 値の受け渡しをするサブルーチン	①値を受け渡すサブルーチンを理解しプログラミングすることができる。	①サブルーチンの動作を理解し、オセロプログラムに例えて値の受け渡しを行うプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。

# 水産海洋技術 [プログラミング]

15 時間目

## ■ 関数の作成

関数とは、値を渡すと、結果（戻り値）が得られるサブルーチンです。



### Step01 関数の作成

テストの点数を渡すと、不合格(30点未満)ならば0、合格(それ以外)ならば1の値を返す関数を作成する。

```
Cls
A=TESUTO(29)
Print A
End

Function TESUTO(TEN)
If TEN<30 Then
K=0
Else
K=1
End If
TESUTO=K
End Function
```

※テストの点数は 29 点

※テストの点数を変数 TEN で受け取る。

※返す値を設定する

帰ってきた値をもとに、合格・不合格を文字で表示する。

```
Cls
A=TESUTO(29)
If A=0 Then
Print "不合格"
Else
Print "合格"
End If
End
```

```
Function TESUTO(TEN)
If TEN<30 Then
K=0
Else
K=1
End If
TESUTO=K
End Function
```

プログラムの解説と実行確認 (約 25分)

## 練習問題

問題 1 テストの点数を渡すと、優秀(80点以上)なら1、それ以外なら2の値を返す関数を作成しなさい。

```
Cls
A=TESUTO(80)
Print A
End

Function TESUTO(TEN)
If TEN>=80 Then
K=1
Else
K=2
End If
TESUTO=K
End Function
```

問題 2 テストの点数を渡すと、優秀(80点以上)なら1、再考査(30点未満)なら0、それ以外なら2の値を返す関数を作成しなさい。

```
Cls
A=TESUTO(80)
Print A
End

Function TESUTO(TEN)
If TEN>=80 Then
K=1
Else
If TEN<30 Then
K=0
Else
K=2
End If
End If
TESUTO=K
End Function
```

練習問題 (約 25分)

## 評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
15	BASIC プログラムの入力と実行 ①関数 関数の作成と利用	①値を受け渡し、処理した結果を返す関数を理解しプログラミングすることができる。	①関数の動作を理解し、簡単な関数を使ったプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。



# 水産海洋技術 [プログラミング]

## 16 時間目

### ■ 二次元配列

表のような構造を持つ配列です。  
一次元配列同様、Dim 命令で使用を宣言します。

(例 1) Dim A(5, 3) ※横 5 マス、縦 3 マスの箱を用意する。

変数 A

(1, 1)	(2, 1)	(3, 1)	(4, 1)	(5, 1)
(1, 2)	(2, 2)	(3, 2)	(4, 2)	(5, 2)
(1, 3)	(2, 3)	(3, 3)	(4, 3)	(5, 3)

値を代入する際には、A(1, 1)=10 : A(2, 1)=20 : A(3, 1)=30 の様にする。

(例 2) 3 人分の、5 教科のテストの点数を記憶する。

変数 TEN

	国語	社会	数学	理科	英語
1 人目	65	81	34	42	16
2 人目	80	95	77	70	75
3 人目	100	100	98	95	92

```
Cls
Dim TEN(5, 3)
TEN(1, 1)=65 : TEN(2, 1)=81 : TEN(3, 1)=34 : TEN(4, 1)=42 : TEN(5, 1)=16
TEN(1, 2)=80 : TEN(2, 2)=95 : TEN(3, 2)=77 : TEN(4, 2)=70 : TEN(5, 2)=75
TEN(1, 3)=100 : TEN(2, 3)=100 : TEN(3, 3)=98 : TEN(4, 3)=95 : TEN(5, 3)=92
```

としてもよいが、  
For で繰り返し、Read、Data、で値を読み込んで代入させると便利である。

```
Cls
Dim TEN(5, 3)
For I=1 To 3
  For J=1 To 5
    Read TEN(J, I)
  Next
Next
Data 65, 81, 34, 42, 16
Data 80, 95, 77, 70, 75
Data 100, 100, 98, 95, 92
```

プログラムの解説と実行確認

このままでは、何も表示されず、結果がわかりにくいいため、中身を表示するプログラムを追加する。

```
Cls
Dim TEN(5, 3)
For I=1 To 3
  For J=1 To 5
    Read TEN(J, I)
  Next
Next

For I=1 To 3
  For J=1 To 5
    Print TEN(J, I);
  Next
Print
Next

Data 65, 81, 34, 42, 16
Data 80, 95, 77, 70, 75
Data 100, 100, 98, 95, 92
```

表示するプログラム



プログラムの解説と実行確認 (約 30 分)

### 練習問題

問題 1 横 4 マス、縦 2 マスの二次元配列を作り、以下の様な値を代入して、表示させてみよう。

50	65	25	80
100	40	75	95

応用 1 この中から一番大きい値を表示してみよう。

```
Cls
Dim TEN(5, 3)
For I=1 To 3
  For J=1 To 5
    Read TEN(J, I)
  Next
Next
MX=0
For I=1 To 3
  For J=1 To 5
    If TEN(J, I)>MX Then
      MX=TEN(J, I)
    End If
  Next
Next
Print "最大値: ";MX
Data 65, 81, 34, 42, 16
Data 80, 95, 77, 70, 75
Data 100, 100, 98, 95, 92
```

練習問題 (約 25 分)

### 評価のポイント

時間	内容	知識・技術	思考・判断・表現	主体的に学習に取り組む態度
16	BASIC プログラムの入力と実行 ①二次元配列 二次元配列の作成と利用	①二次元配列について理解しプログラミングすることができる。	①二次元配列を理解し、簡単なプログラムを自ら考え表現(作成)することができる。	①プログラミングに主体的に取り組もうとしている。