

Basicの基礎・グラフィックス

0. 目次

6. グラフィックス

6. 1 グラフィック画面の設定

6. 2 直線の描画

6. 3 円の描画

6. 4 点の描画

6. グラフィックス

Basicの画面には、テキスト画面とグラフィック画面がある。テキスト画面には文字が表示でき、グラフィック画面には点や線が表示できる。

6. 1 グラフィック画面の設定

グラフィック画面の利用手順を示す。

①背景色（グラフィック画面の色）の設定

BackColor="White"（デフォルト値）

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

Navy:濃紺色 Teal:青緑色 Maroon:えび茶色 Fuchsia:赤紫色
(フクシャと読む)

②前景色（グラフィック画面に描く点や線の色）の設定

ForeColor="Black"（デフォルト値）

③グラフィック画面のサイズ（横幅、縦幅）を指定

GScreen(w, h) グラフィック画面のサイズを横幅w、縦幅h
ピクセル単位で設定する。たとえば、w=200, h=100

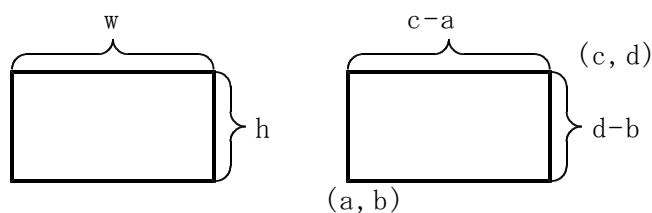
④座標の設定

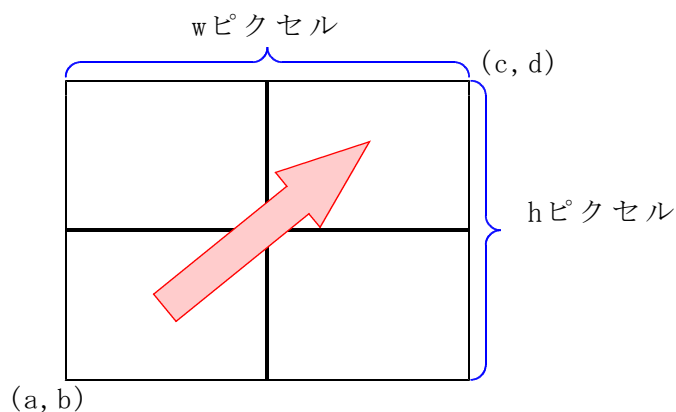
●数学的座標の場合

Window(a, b)-(c, d) グラフィック画面の左下隅を(a, b)、
右上隅を(c, d)に設定する。 $a < b$, $c < d$

MathGraph On グラフィック画面を数学的座標（左下から右上へ）に
指定

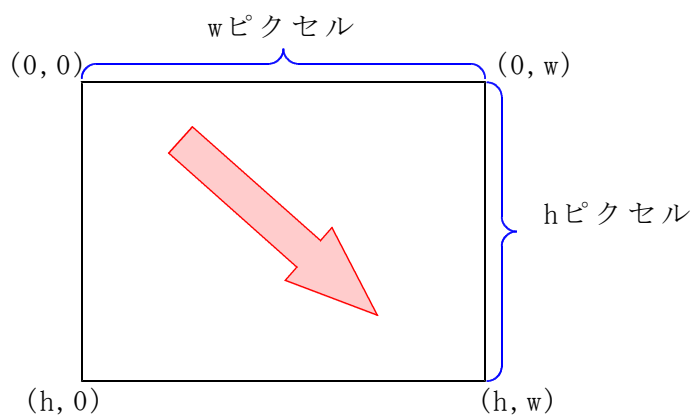
(注意) Windowの縦横比とGScreenの縦横比は同じがよい。
 $w : h = (c-a) : (d-b)$





● コンピュータ座標

MathGraph Off グラフィック画面を通常のコンピュータでの方法
 (左上から右下へ) に指定。
 デフォルトはMathGraph Off。



グラフィック画面の設定を行う。

文	機能
GScreen	GScreen (x, y) 横 x ピクセル、縦 y ピクセルの大きさの画面を開く。
Window	グラフィック画面の座標を設定する。 Window (x1, y1)-(x2, y2) 左下隅 $(x1, y1)$ から右上隅 $(x2, y2)$ まで。 グラフィック画面の座標を指定する。
MathGraph	MathGraph On グラフィック座標を数学の方法 (左下から右上へ) に指定する。 MathGraph Off グラフィック座標をコンピュータの方法 (左上から右下へ) に指定する。こちらがデフォルト値。

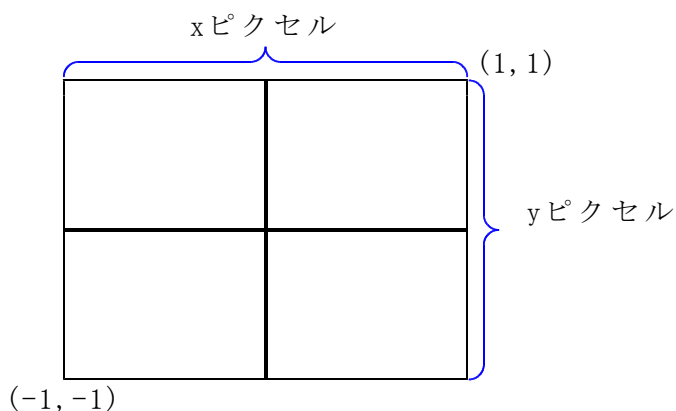
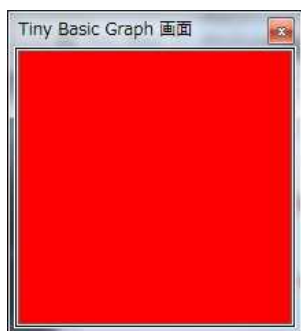
●プログラム (K611. bas)

```

1  ' << K611. bas >>
2  '
3  '   グラフィック画面の背景色を赤に設定。
4  BackColor="Red"
5  '
6  '   グラフィック画面のサイズを設定。
7  X=200: Y=200
8  GScreen(X, Y)
9  '
10 '   座標の設定。
11 Window(-1, -1)-(1, 1)
12 MathGraph On
13 End

```

実行結果



1 6種類の色を表示する。

文	機能
GLocate	GLocate(x, y) グラフィック画面での位置(x, y)を指定する。
GPrint	GPrint 式 グラフィック画面に式の値を表示する。

●プログラム (K612. bas)

```

1  ' << K612. bas >>
2  '   グラフィック画面の設定。
3  '
4  '   グラフィック画面のサイズを設定。
5  GScreen(200, 300)
6  '
7  '   座標の設定。
8  Window(0, 0)-(1, 1)
9  MathGraph On
10 '
11 '   16種類の色を文字型配列C$(15)に設定。

```

```

12 ' 配列の添え字は、色番号に対応する。
13 '
14 Dim C$(15)
15 '
16 C$(0)="Black": C$(1)="Navy": C$(2)="Green": C$(3)="Teal"
17 C$(4)="Maroon": C$(5)="Purple": C$(6)="Olive": C$(7)="Silver"
18 C$(8)="Gray": C$(9)="Blue": C$(10)="Lime": C$(11)="Cyan"
19 C$(12)="Red": C$(13)="Fuchsia": C$(14)="Yellow": C$(15)="White"
20 '
21 ' 16種類の色を表示。
22 For I=0 To 15
23 ' 前景色を黒に設定。
24 ForeColor="Black"
25 ' 描画位置を決める。
26 GLocate(0.1, I/17+0.1)
27 ' 色番号を描画する。
28 GPrint "("+Str$(I)+") "
29 ' 前景色をC$(I)に設定。
30 ForeColor=C$(I)
31 ' 描画位置を決める。
32 GLocate(0.3, I/17+0.1)
33 ' ●を描画する。
34 GPrint C$(I)+" ● "
35 Next I
36 End

```

実行結果



6. 2 直線の描画

直線を描画する。

文	機能
Line	<p>グラフィック画面に直線を描く</p> <p>Line(x1, y1)-(x2, y2), c (x1, y1)から(x2, y2)まで色番号cで直線を引く。</p> <p>ForeColor="Red" Line(x1, y1)-(x2, y2) (x1, y1)から(x2, y2)まで赤色で直線を引く。</p> <p>Line(x1, y1)-(x2, y2), c, b (x1, y1)と(x2, y2)を対角頂点とする色番号cの長方形を描く。</p> <p>ForeColor="Red" Line(x1, y1)-(x2, y2), , b (x1, y1)と(x2, y2)を対角頂点とする赤色の長方形を描く。</p> <p>Line(x1, y1)-(x2, y2), c, bf (x1, y1)と(x2, y2)を対角頂点とする色番号cの長方形を描く。内部も同じ色で塗る。</p> <p>ForeColor="Red" Line(x1, y1)-(x2, y2), , bf (x1, y1)と(x2, y2)を対角頂点とする赤色の長方形を描く。内部も同じ色で塗る。</p> <p>DrawWidth=数値 線の太さはDrawWidthで指定する。 初期値は1。</p>

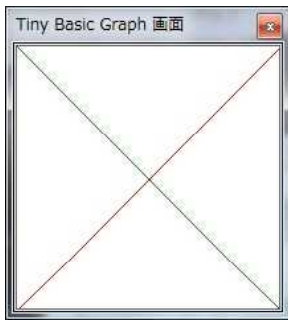
●プログラム (K621. bas)

```

1  ' << K621. bas >>
2  ' 直線の描画。
3  '
4  ' グラフィック画面のサイズを設定。
5  GScreen(200, 200)
6  '
7  ' 座標の設定。
8  Window(-1, -1)-(1, 1)
9  MathGraph On
10 '
11 ' 左下隅から右上隅への色番号12 (赤色) の対角線を描く。
12 Line(-1, -1)-(1, 1), 12
13 '
14 ' 左上隅から右下隅への緑の対角線を描く。
15 ForeColor="Green"
16 Line(-1, 1)-(1, -1)
17 End

```

実行結果



線の太さを指定する。

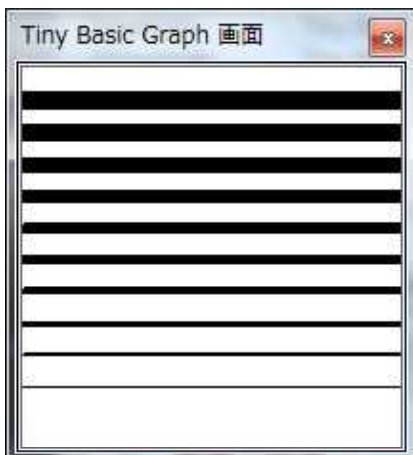
●プログラム (K622. bas)

```

1  ' << K622. bas >>
2  ' 線の太さ比較。
3  '
4  ' グラフィック画面のサイズを設定。
5  GScreen(200,200)
6  '
7  ' 座標の設定。
8  Window(0,0)-(1,1)
9  MathGraph On
10 '
11 ' 線の太さが1から10までの直線を描く。
13 For I=1 To 10
14   ' 線の太さをIに設定。
15   DrawWidth=I
16   ' (0,Y)から(1,Y)まで直線を引く。
17   Y=I/11
18   Line(0,Y)-(1,Y)
19 Next I
20 End

```

実行結果



長方形を描画する。

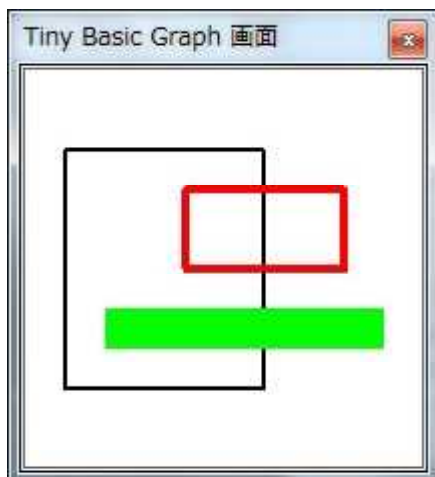
●プログラム (K623. bas)

```

1  ' << K623. bas >>
2  ' 長方形の描画。
3  '
4  ' グラフィック画面のサイズを設定。
5  GScreen(200,200)
6  '
7  ' 座標の設定。
8  Window(0,0)-(1,1)
9  MathGraph On
10 '
11 ' 黒の長方形を描く。
12 ForeColor="Black"
13 DrawWidth=2
14 Line(0.1,0.2)-(0.6,0.8),,B
15 '
16 ' 赤の長方形を描く。
17 C=12
18 DrawWidth=4
19 Line(0.4,0.5)-(0.8,0.7),C,B
20 '
21 ' 緑の長方形を描く。
22 ForeColor="Lime"
23 DrawWidth=1
24 Line(0.2,0.3)-(0.9,0.4),,BF
25 End

```

実行結果



6. 3 円の描画

円を描く。

文	機能
Circle	グラフィック画面に円を描く。 Circle(x, y), r, c (x, y)を中心とする半径rの円を色番号cで描く。 ForeColor="Red" Circle(x, y), r, c (x, y)を中心とする半径rの円を赤色で描く。

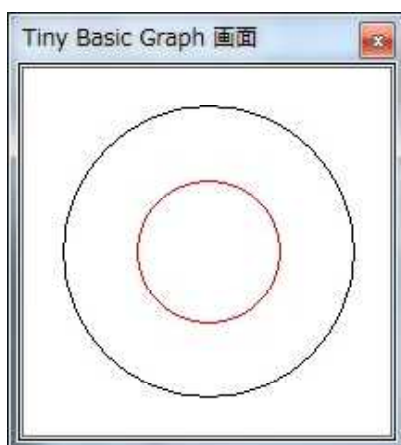
●プログラム (K631. bas)

```

1  ' << K631. bas >>
2  '  円の描画。
3  '
4  '  グラフィック画面のサイズを設定。
5  GScreen(200, 200)
6  '
7  '  座標の設定。
8  Window(-1, -1)-(1, 1)
9  MathGraph On
10 '
11 '  (0, 0)を中心とする半径0.8の円を黒色で描く。
12 Circle(0, 0), 0.8, 0
13 '
14 '  (0, 0)を中心とする半径0.4の円を前景色（赤色）で描く。
15 ForeColor="Red"
16 Circle(0, 0), 0.4
17 End

```

実行結果



塗りつぶした円を描く。

文	機能
Paint	領域を指定した色で塗る。 Paint(x, y) 点(x, y)を含む境界色で囲まれた領域を前景色で塗りつぶす。

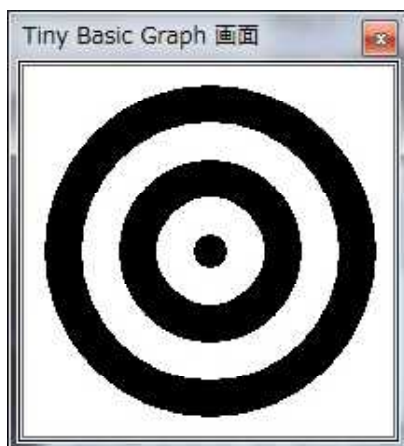
●プログラム (K632. bas)

```

1  ' << K632. bas >>
2  ' 円の描画 (塗りつぶし)。
3  '
4  ' グラフィック画面のサイズを設定。
5  GScreen(200, 200)
6  '
7  ' 座標の設定。
8  Window(-1, -1)-(1, 1)
9  MathGraph On
10 '
11 ' 円。
12 C$="Black"
13 For R=0.9 To 0.1 Step -0.2
14   ' 前景色をC$に設定。
15   ForeColor=C$
16   ' (0, 0)を中心とする半径Rの円を前景色C$で描く。
17   Circle(0, 0), R
18   ' 点(0, 0)を含む境界色で囲まれた領域を前景色C$で塗りつぶす。
19   Paint(0, 0)
20   ' 前景色を切り替える。
21   If C$ = "Black" Then C$="White" Else C$="Black"
22 Next R
23 End

```

実行結果



点滅する円を描く。

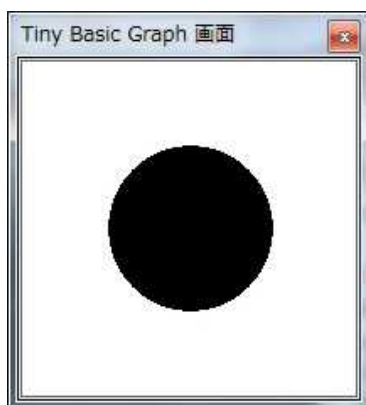
●プログラム (K633. bas)

```

1  ' << K633. bas >>
2  ' 点滅する円の描画。
3  '
4  ' グラフィック画面のサイズを設定。
5  GScreen(200,200)
6  '
7  ' 座標の設定。
8  Window(-1,-1)-(1,1)
9  MathGraph On
10 '
11 ' 円。
12 C$="Black"
13 For I=1 To 100
14   ' 前景色をC$に設定。
15   ForeColor=C$
16   ' (0,0)を中心とする半径0.5の円を前景色C$で描く。
17   Circle(0,0),0.5
18   ' 点(0,0)を含む境界色で囲まれた領域を前景色C$で塗りつぶす。
19   Paint(0,0)
20   ' 前景色を切り替える。
21   If C$ = "Black" Then C$="White" Else C$="Black"
22   ' 1000ミリ秒待つ。
23   Sleep(1000):
24 Next I
25 End

```

実行結果



6. 4 点の描画

点を描く。

文	機能
PSet	グラフィック画面に点を描く。 PSet (x, y) (x, y)に点を前景色で描画する。 PSet (x, y), c (x, y)に点を色番号cで描画する。

関数	機能
Rnd	[0, 1)の乱数を返す。

●プログラム (K641. bas)

```

1  ' << K641. bas >>
2  ' 点の描画。
3  '
4  ' グラフィック画面のサイズを設定。
5  GScreen(200, 200)
6  '
7  ' カラーの模様を描く。
8  For X=0 To 99
9    For Y=0 To 149
10     C=Int(15*Rnd)
11     ' (X, Y)に点を色番号Cで描画する。
12     PSet(X, Y), C
13   Next Y
14 Next X
15 End

```

実行結果

