

I. プログラミング言語 Active Basic Ver2.6 を用いたプログラム開発の概要

① Basic とは、Beginner's All-purpose Symbolic Instruction Code の略で、1964年ダートマス大学で大型コンピュータの教育用言語として開発された。

コンピュータプログラミングの入門用としても、本格的なアプリケーション作成用としても使える。

現在、"Visual Basic", "F-Basic", "Yabasic", "Active Basic" などの Windows 上で動作するプログラムを作成するための様々な Basic がある。

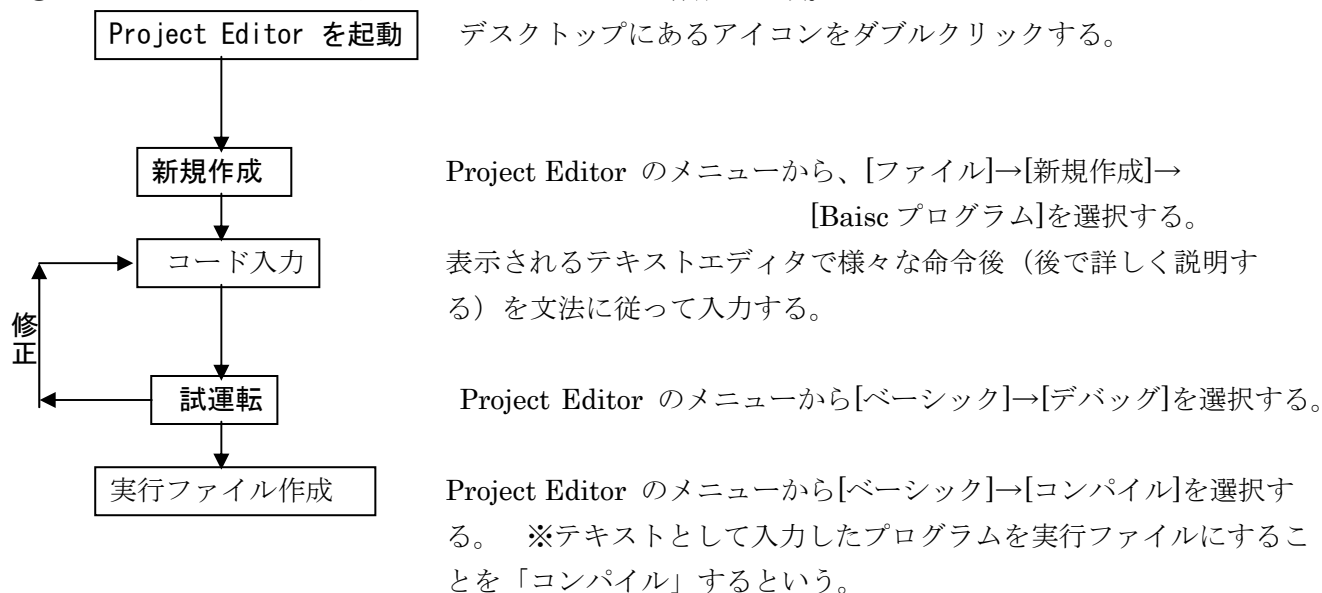
② 本授業で使用する "Active Basic" について。

静岡県在住の山本大祐氏(21 歳)開発の Windows 用プログラム開発言語。授業で使用する Active Basic Ver2.6 は山本氏が 16 歳 (!) のときに作られている。

(<http://www.activebasic.com/activebasic2.6/index.html>)

現在では、Windows アプリケーションの開発ツールとして進化した "Active BASIC 4.24" が、Ver2.6 とともにフリーソフト (無料, 著作権は山本氏) として使用できる。

③ Active Basic Ver2.6 を用いたアプリケーション作成の手順。



II. プログラミング実習 (■は半角スペース)

① ウィンドウに文字を表示させる。

例 1. 文字を表示させる。(下記の文字は全て半角、ただし " " の間は全角でも良い) ex01

```
print■"Hello!"
print■"My name is Potter."
```

例 2. 数値を表示させる。(下記の文字は全て半角) ex02

```
print■5
print■12+9
```

例 3. 複数のものを表示させる。(下記の文字は全て半角) [注]; (セミコロン) で区切る。 ex03

```
print■"12+9=";12+9
```

[注] 掛け算は * 割り算は / 足し算は + 引き算は - 累乗は ^

課題: 次の計算結果を上例 3 のように表示させなさい。

① $5 \times (6 + 3)$

② $(17 - 5)^2 \div 3$

②キーボードからの入力を受けさせる。

例 1. 数字入力を受けさせ、計算結果を表示させる。 ex04

```
input ■ a
print ■ a*a
```

[注]変数は半角アルファベットで始まる文字列なら何でも良い。

例. hankei
namae
p99 など

例 2. 数字入力を複数受けさせ、計算結果を表示させる。 [注] ,(カンマ)で区切る。 ex05

```
input ■ tate, yoko, takasa
print ■ tate*yoko*takasa
```

例 3. 文字変数の入力。 [注] 変数の最後尾に \$ をつけると文字変数とされる。 ex06

```
input ■ sei$
input ■ mei$
print ■ "-----"
print ■ sei$;mei$
```

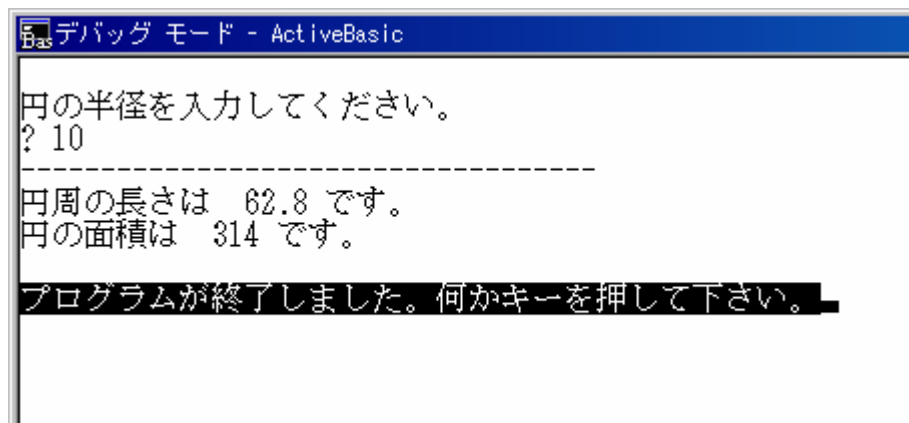
例 4. 文字変数の入力と連結 ex07

```
input ■ a$,b$
print ■ a$+b$
```

課題：これまでの学習内容を使用してつぎのプログラムをつくりなさい。

①半径の値を入力すると、円周の長さ、円の面積を表示する。(円周率は 3.14) ex08

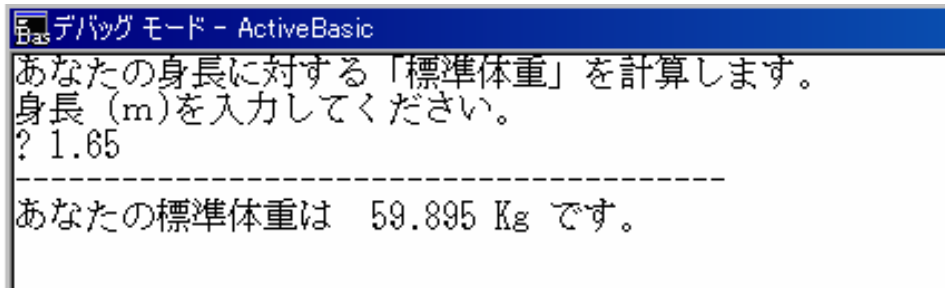
ただし、次のように表示させること。



プログラム例

```
print ■ "円の半径を入力してください。"
input ■ hankei
print ■ "-----"
print ■ "円周の長さは" ;2* hankei *3.14;"です。"
print ■ "円の面積は" ; hankei *hankei *3.14;"です。"
```

課題② 下図のような、身長(単位はメートル)を入力すると「標準体重」を計算するプログラムを作成
 下さい。ただし標準体重は、身長×身長×22 で計算する。 ex09



[3] 判断を伴う処理

例 1. 入力した値によって処理を変える。(1) ex10

```

print "あなたの誕生月を入力してください。"
input a
if a=1 then print "あなたの誕生石はガーネットです"
if a=2 then print "あなたの誕生石はアメシストです"
if a>2 then print "あなたの誕生石は知りません"
    
```

課題③ 下図を参考にして上のプログラムを全ての月に対応させて作り直して下さい。 ex11

1	2	3	4	5	6
ガーネット	アメシスト	アクアマリン	ダイヤモンド	エメラルド	パール

7	8	9	10	11	12
ルビー	ペリドット	サファイア	オパール	トパーズ	ターコイズ

例 2. 入力した値によって処理を変える。(2) ex12

```

print "あなたの誕生月を入力してください。"
input a

if a=1 then
    print "あなたの誕生石はガーネットです"
    print "あなたの誕生花はシンビジュームです"
endif

if a=2 then
    print "あなたの誕生石はアメシストです"
    print "あなたの誕生花はフリージアです"
endif

if a>2 then print "あなたのことは知りません"
    
```

[4] 繰り返し処理

(1)goto 文による繰り返し処理

例. 正解が入力されるまで何回も入力させるプログラム。 ex13

```
print ■ "鶴と亀があわせて 7 匹（羽）います。これらの足の数は合計 20 本でした。"
10 print ■ "-----"
print ■ "亀は何匹ですか。"
input ■ k
print ■ "鶴は何羽ですか。"
input ■ t
if ■ k<>3 ■ then
  ■ print ■ "間違っています。"
  ■ goto ■ 10
endif
if ■ t<>4 ■ then
  ■ print ■ "間違っています。"
  ■ goto ■ 10
endif
print ■ "正解です！！！”
```

課題

①既出のプログラム ex11 を修正し、誕生月としてふさわしくない月（1 より小さい値や 12 より大きい値）が入力されたとき ”間違っています。” と表示し再入力させるプログラムとしなさい。 ex14

②次のプログラムの実行結果を考えなさい。

```
print ■ " 繰り返す回数を入力してください。"
input ■ x
10 print ■ "お気に！" ; x ; "回目"
x=x-1
if ■ x>0 ■ then ■ goto ■ 10
end
```

(注) x=x-1 は、x から 1 引いた値を x に代入するという意味。

③ 上の②を作り、ex15 として保存し、実行結果を確認しなさい。

(2) for next 文による繰り返し処理

例①

ex15

```
print "繰り返す回数を入力してください。"
input n
for i=1 to n
print "お気楽に！" ; i ; "回目"
next i
```

例②

ex16

```
print "正の整数を入力してください。"
input x
goukei=0
for i=1 to x
goukei=goukei+i
next i
print "1 から" ; x ; "までの和は" ; goukei ; "です。"
```

【課題】上の例②を参考にして、 $X!$ を計算するプログラムを作成しなさい。 ex17

[5] 代入文

※通常ベーシックでは、等号（＝）は、左辺に右辺の値を代入するという意味で用いられる。ただし、「if 文」では等号（＝）は等しいという意味で使う。

例①

ex18

```
input x
x=x+1
print x
x=x+1
print x
```

例②下記のプログラムは何をするプログラムか？

ex19

```
input x, y
if x>y then
z=x
x=y
y=z
end if
print x, y
```

〔6〕グラフィック命令

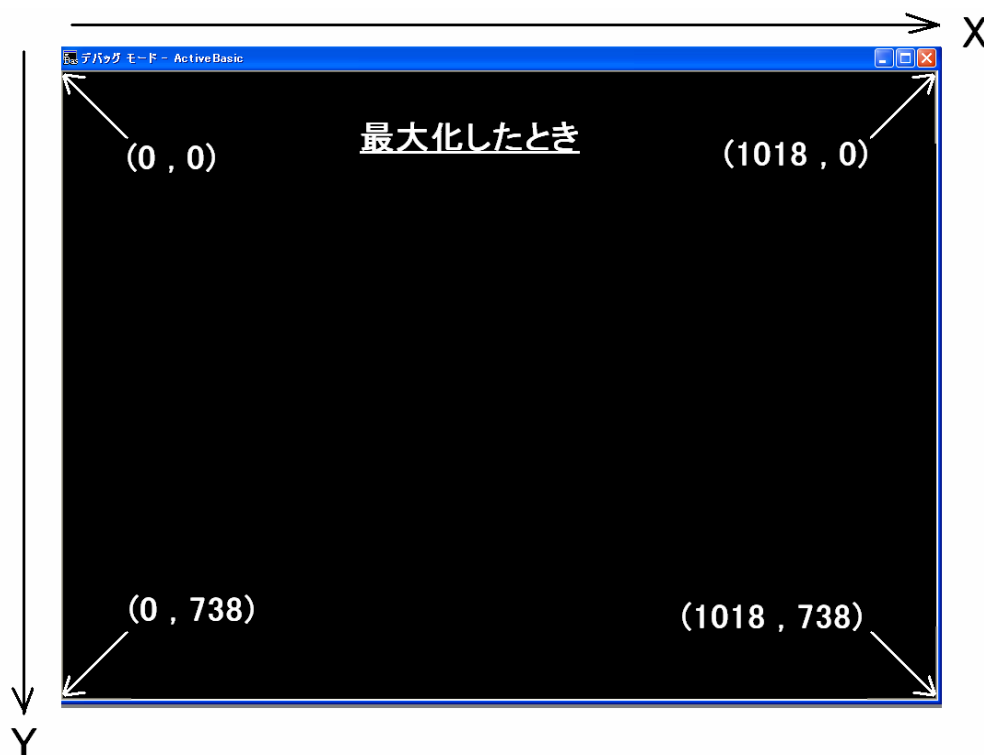
(1) 描かれたものを消去する

命令語	役割
cls■1	実行画面中に書かれた文字を消去する
cls■2	実行画面に描かれた図形を消去する
cls■3	実行画面に描かれた文字と図形を消去する

(2) 点を描く

命令語	役割
pset([x座標], [y座標]), [色]	指定した座標に指定した色の点を描く

※下図のように実行画面の座標を指定する



※色は次のような数字を使う

数値	0	1	2	3	4	5	6	7
色	黒	青	赤	マゼンタ	緑	シアン	黄	白

例① 点を描く

ex20

```

10■print■" x 座標を入力して下さい"
inputx
print■" y 座標を入力して下さい"
input■y
print■" 色を入力して下さい"
pset(x , y) , c
cls■1
goto■10

```

例② 乱数を使って沢山の点を描く

【注】関数 `rnd(1)` は 0 以上 1 未満の乱数 ex21

```
10 ■ x=1018*rnd(1)
   y=738*rnd(1)
   c=8*rnd(1)
   goto ■ 10
```

例③ 放物線を描く

ex22

```
for ■ x=0 ■ to ■ 600
  ■ y=(x-300) * (x-300) * 0.005+30
  ■ pset(x+50, y), 5
next ■ x
```

例③ 波を描く

ex23

```
for ■ x=0 ■ to ■ 800
  ■ y=200*sin(x/50)+200
  ■ pset(x, y), 5
next ■ x
```

例③ コンピュータアートで楽しむ（上の例③を少し改造）

ex24

```
for ■ a=0 ■ to ■ 100
for ■ x=0 ■ to ■ 800
  ■ y=200*sin(x/50)+200
  ■ pset(a+x, a+y), 5
next ■ x
next ■ a
```

(3)円を描く

ex25

命令語	役割 (x, y, r, c は数値または数値変数)
<code>circle(x, y), r, c</code>	点(x, y)に半径r、色cの円を描く
<code>circle(x, y), r, c, , , , f</code>	点(x, y)に半径r、色cの円を描き塗りつぶす

例①

ex26

```
circle (100, 100), 20, 7, , , , f
circle (160, 100), 20, 7, , , , f
circle (130, 150), 60, 7, , , , f
```

例②

ex27

```
circle (90, 100), 20, 7, , , , f
circle (170, 100), 20, 7, , , , f
circle (130, 150), 60, 7, , , , f
```

例③

ex28

```
circle (90, 100), 30, 7, , , , f
circle (170, 100), 30, 7, , , , f
circle (130, 150), 60, 7, , , , f
```

例④

```
circle(250, 150), 20, 4, , , , f
circle(450, 150), 20, 4, , , , f
circle(260, 150), 10, 0, , , , f
circle(450, 150), 20, 4, , , , f
circle(440, 150), 10, 0, , , , f
circle(350, 330), 40, 2, , , , f
circle(330, 250), 5, 3, , , , f
circle(370, 250), 5, 3, , , , f
circle(350, 230), 200, 6
```

例⑤乱数を使って沢山の円を描く

【注】関数 `rnd(1)` は0以上1未満の乱数

```
10■x=1018*rnd(1)
y=738*rnd(1)
c=8*rnd(1)
r=10
circle(x, y), r, c
goto■10
```

(4) その他のグラフィック命令文

①直線や四角形を描く

<code>line(x1, y1)-(x2, y2), c</code>	点(x1, y1)から点(x2, y2)に色 c の線分を描く。
<code>line(x1, y1)-(x2, y2), c, b</code>	点(x1, y1)から点(x2, y2)に色 c の長方形を描く。
<code>line(x1, y1)-(x2, y2), c, bf</code>	点(x1, y1)から点(x2, y2)に色 c の長方形を描き塗りつぶす。

②画面を指定した色で塗りつぶす。

`paint (x, y), c`

例①

```
Paint(0, 0), 7
Line(100, 100)-(200, 200), 4
Line(180, 180)-(250, 250), 6, b
Line(200, 200)-(300, 300), 1, bf
```

例②簡単なアニメーション

```
line(0, 0)-(200, 200), 4, bf
vx=0.5
vy=0.4
x=6
y=6
10 for i=1 to 20
circle(x, y), 5, 3, , , f
Next i
circle(x, y), 5, 4, , , f
If x>=196 then vx=-1*vx
If y>=196 then vy=-1*vy
If x<6 then vx=-1*vx
If y<6 then vy=-1*vy
x=x+vx
y=y+vy
goto 10
```