

Basicの基礎・変数と入出力

0. 目次

2. 変数と入出力

2. 1 変数とは

2. 2 数値型変数

2. 2. 1 数値の演算

2. 2. 2 数値の出力

2. 2. 3 数値の入力

2. 3 文字型変数

2. 3. 1 文字列の演算

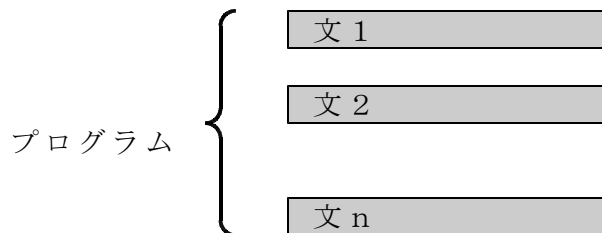
2. 3. 2 文字列の出力

2. 3. 3 文字列の入力

2. 3. 4 文字列の関数

2. 変数

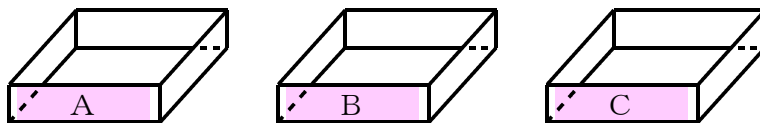
プログラムは、コンピュータに実行させる処理を文書にしたものである。文書は、厳密に決められた指示で書かれている。この指示を「文」という。すなわち、プログラムは、文の集まりで、記述された順に先頭から実行されていく。



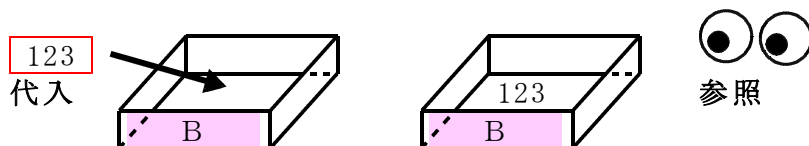
種々の処理を行うために、多くの文が用意されている。

2. 1 変数とは

- 変数とは名前(変数名という)の付いた箱のようなもので、コンピュータ内の記憶場所にある。プログラムが実行されていく過程で、変化するデータを保存する。



- 箱にデータを入れることを「代入」といい、箱の中のデータを見ることを「参照」という。



箱にデータを入れると、前に入っていたデータは消去され、復元できない。

- 箱に入るデータの種類は決まっており、数値だけが入る数値型変数、文字だけが入る文字型変数とがある。文字型変数名には、末尾に「\$」をつける。

- 変数名は、英字から始まる英数字で構成される。大文字、小文字は区別されない。

たとえば、ABC78, xy99, PQ\$, pq3\$。

テキストでは、変数名は大文字に統一している。

2. 2 数値型変数

2. 2. 1 数値の演算

加減乗除等の演算ができる。

文	機能
代入文	変数=式 右辺の式が計算され結果が左辺の変数に代入される。 右辺に変数が現れると、変数の値が参照され計算に使われる。
Print	数値や文字を画面に表示する。 Print 変数 変数の値を表示後、改行する。 Print"文字列";変数 「文字列」に引き続き変数の値を表示した後改行する。 「;」をつけると改行が行われない。
End	プログラムの実行を終了する。

●プログラム (K221a. bas)

```

1 ' << K221a. bas >>
2 '
3 A=2
4 B=3
5 Print"A=";A;" B=";B
6 Print"A+B=";A+B: ' 加算。
7 Print"A-B=";A-B: ' 減算。
8 Print"A*B=";A*B: ' 乗算。
9 Print"A/B=";A/B: ' 除算。
10 Print"A^B=";A^B: ' べき乗。
11 End

```

(注意 1) 「'」以降の文字列は、注釈として解釈され実行されない。

実行結果

```

1 A= 2 B= 3
2 A+B= 5
3 A-B=-1
4 A*B= 6
5 A/B= 0.66666666666666667
6 A^B= 8
7 OK

```

余り、整数化、絶対値等の計算もできる。

関数は与えられたデータに従ってある処理を行い、その結果を関数値として返す。すでに定義されていて、すぐに使える関数を組み込み関数（以後、関数という）という。

関数	機能
Abs	Abs(X) 変数Xの絶対値を返す。 Abs(3)は3を返す。 Abs(-3)は3を返す。
Int	Int(X) 変数Xの値を超えない整数を返す。 Int(2.6)は2を返す。 Int(-3.7)は-4を返す。
Mod	A Mod B AをBで割った余りを返す。 Aは整数、Bは正整数。 15 Mod 3 は0を返す。 16 Mod 3 は1を返す。 17 Mod 3 は2を返す。 -15 Mod 3 は0を返す。 -16 Mod 3 は-1を返す。 -17 Mod 3 は-2を返す。

●プログラム (K221b. bas)

```

1  ' << K221b. bas >>
2  '
3  A=123: B=45
4  Print"A=";A;" B=";B
5  '
6  Print"A Mod B=";A Mod B: ' AをBで割った余り。
7  Print"Int(A/B)=";Int(A/B): ' 関数Int(X) : 変数Xの整数部。
8  Print"Abs(-3)=";Abs(-3): ' 関数Abs(X) : 変数Xの絶対値。
9  End

```

実行結果

```

1  A= 123 B= 45
2  A Mod B= 33
3  Int(A/B)= 2
4  Abs(-3)= 3
5  OK

```

2. 2. 2 数値の出力

出力の書式を考察する。

文	機能
Print Using	数値や文字列を指定した書式で表示する。 Print Using"#####";A 変数Aの値を5桁右寄せで表示する。 Print Using"#####.##";A 変数Aの値を小数点付きで表示する。

●プログラム (K222a. bas)

```

1  ' << K222a. bas >>
2  ' 整数。
3  A=123
4  Print A
5  Print Using"##";A
6  Print Using"###";A
7  Print Using"####";A
8  End

```

実行結果

```

1  123
2  |%123|      数値の桁数が表示幅より大きい場合
3  |123|      数値の桁数が表示幅と等しい場合
4  | 123|     数値の桁数が表示幅より小さい場合、右寄せ。
5  OK

```

●プログラム (K222b. bas)

```

1  ' << K222b. bas >>
2  ' 実数。
3  B=12.35
4  Print B
5  Print Using"##.#";B
6  Print Using"##.##";B
7  Print Using"##.###";B
8  End

```

実行結果

```

1  12.35
2  |12.4|      数値の精度が表示幅より大きい場合、四捨五入される。
3  |12.35|     数値の桁数が表示幅と等しい場合
4  |12.350|    数値の桁数が表示幅より小さい場合、0が埋められる。
5  OK

```

2. 2. 3 数値の入力

Read文を使って、Data文からデータを読み込む。

文	機能
Data	・データ(数値、文字列)をカンマで区切って設定する。
Read	・Data文に記述されたデータ(数値、文字列)を変数に読み込む。 ・対応するデータがない場合や型が違う時はエラーになる。

●プログラム(K223. bas)

```

1  ' << K223. bas >>
2  ' 数値の入力。
3  Read A, B
4  '
5  Print "A="; A
6  Print "B="; B
7  End
8  '
9  ' データ。
10 Data 123, 45.678

```

実行結果

```

1  A= 123
2  B= 45.678
3  OK

```

2. 3 文字型変数

2. 3. 1 文字列の演算

文字列の連結演算ができる。「+」または「&」記号で文字列の連結を行う。

●プログラム (K231. bas)

```

1  ' << K231. bas >>
2  '
3  A$="abc"
4  B$="12345"
5  Print"A$=";A$;" B$=";B$
6  '
7  Print"A$+B$=";A$+B$
8  '
9  C$=B$&A$
10 Print"C$=";C$
11 End

```

実行結果

```

1  A$=abc B$=12345
2  A$+B$=abc12345
3  C$=12345abc
4  OK

```

2. 3. 2 文字列の出力

文字列の書式を考察する。

文	機能
Print Using	文字列を指定された書式で表示する。 &<n個の空白>& 先頭からn+2文字分を出力する。 長い場合は無視され、短い場合は左詰で出力され、 空白で埋められる。

●プログラム (K232. bas)

```

1  ' << K232. bas >>
2  '
3  A$="abcd"
4  Print Using"& &|";A$
5  Print Using"& &|";A$
6  Print Using"& &|";A$
7  End

```

実行結果

```

1  |abc|   表示幅が文字列の長さより小さい場合、左から表示幅分表示。
2  |abcd|  表示幅が文字列の長さと等しい場合
3  |abcd | 表示幅が文字列の長さより大きい場合、空白で埋められる。
4  OK

```

2. 3. 3 文字列の入力

Read文、Data文を使って、文字列を入力する。

文字列は、「"」で囲む。

●プログラム (K233. bas)

```
1 ' << K233. bas >>
2 '
3 Read A$, B$, C$
4 '
5 Print "A$="; A$
6 Print "B$="; B$
7 Print "C$="; C$
8 End
9 '
10 ' データ。
11 Data "ABCDE", "12345", "漢字"
```

実行結果

```
1 A$=ABCDE
2 B$=12345
3 C$=漢字
4 OK
```


2. 3. 4 文字列の関数

●文字列の長さ

関数	機能
Len	<p>文字列の長さを求める。</p> <p>Len(S\$) 変数S\$の文字数を返す。 半角文字は1文字分とされ、全角文字(漢字を含む)も1文字分とされる。</p> <p>○例 A\$=abcde Len(A\$)は、「5」となる。 B\$=茨城太郎 Len(B\$)は、「4」となる。</p>

●文字列の抽出

関数	機能
Mid\$	<p>文字列を抽出する。</p> <p>Mid\$(S\$, m, n) 変数S\$のm文字目からn文字分を返す。</p> <p>○例 A\$=abcdefg Mid\$(A\$, 2, 3)は、「bcd」となる。 B\$=茨城県水戸市 Mid\$(B\$, 2, 3)は、「城県水」となる。</p>
Left\$	<p>文字列を抽出する。</p> <p>Left\$(S\$, n) 変数S\$の左端からn文字分を返す。</p> <p>○例 A\$=abcdefg Left\$(A\$, 2)は、「ab」となる。 B\$=茨城県水戸市 Left\$(B\$, 2)は、「茨城」となる。</p>
Right\$	<p>文字列を抽出する。</p> <p>Right\$(S\$, n) 変数S\$の右端からn文字分を返す。</p> <p>○例 A\$=abcdefg Right\$(A\$, 3)は、「efg」となる。 B\$=茨城県水戸市 Right\$(B\$, 3)は、「水戸市」となる。</p>

●文字列の検索

関数	機能
Instr	<p>文字列を検索する。</p> <p>Instr(n, A\$, B\$) 文字列A\$のn番目の文字から文字列B\$を探し、見つければ、開始位置からの文字数で返す。見つからない場合0を返す。</p> <p>○例 A\$=abcdefg B\$=def Instr(1, A\$, B\$)は、「4」となる。 Instr(3, A\$, B\$)は、「2」となる。</p>

●文字列の置換

関数	機能
Replace\$	<p>文字列を置換する。</p> <p>Replace\$(A\$, B\$, C\$) A\$中に含まれるB\$をC\$で置き換えた結果を返す。</p> <p>○例 A\$=abcdefg B\$=def C\$=33 Replace\$(A\$, B\$, C\$)は、「abc33g」となる。</p>

●文字列の分解

関数	機能
Split\$	<p>文字列を分解する。</p> <p>Split\$(A\$, B\$) A\$を区切り文字B\$で分解し、最初の区切りまでの文字列を返す。 文字列変数A\$から最初の区切り文字までが削除される。</p> <p>○例 A\$=abc@123 B\$=@ Split\$(A\$, B\$)は、「abc」となる。 A\$は、「123」となる。</p>