

H17年度 前半(1~4回)講義のポイント

- 範囲 ~ 1. テキスト第1章(下記のただし書き以外については、きちんと確認して下さい)
ただし、複素数と p.31 の表 1.6 主な特殊数学関数および p.22 の表 1.4 の一部の関数 (flipud, fliplr, conj, rot90), p.24 行列の結合, 縮小, 切り出しについては、講義の範囲外とします。
2. for ~ end 文, 合計を求める定番のプログラム記述法(ホームページ資料参照のこと)
3. disp 文(テキスト pp.55~58 参照のこと)

変数名の付け方

- 変数名の最初の文字は必ず英字で始める。2文字目以降は英字, 数字, アンダーバーを使用する。それらを組み合わせて用いてもよい。例: total, x1, YS_11
- 大文字と小文字は区別される。例えば, 変数名 y と Y は異なる変数名と解釈される。
- 変数名を不必要に長くしない。最大 63 文字まで可であるが, 6 文字程度に抑える。
- 変数名として, 予約定数, 予約変数(テキスト pp.18, 表 1.2 参照)を用いない。

算術代入文の解釈

>>a=5 文(statement)の中に=が入っている文を”算術代入文”と呼ぶ。
 解釈は, 数学的な意味(左辺と右辺が等しい)とは異なり, =の右辺の数値または式の計算結果を左辺の変数名(記憶場所に付した名前)に代入しなさいの意味。
 (注)=の右辺に変数名が記述される場合には, 事前にその変数名に具体的な数値などが代入されている必要がある。

a=
5 変数名 a に数値 5 が代入された結果をエコープリントで表示

>>a=5; 文末に; が付くと, エコープリント非表示
>> 即ち, 文末の; の有無がエコープリント表示, 非表示に対応

文字列を変数に代入するには?

>>x='MATLAB' 文字列を変数に代入するときには, 文字列の前後をシングルのクォーテーション'で囲む。

x=
MATLAB

四則演算の記号

+ : 足し算, - : 引き算, * : 掛け算, / : 割り算, ^ : べき乗
計算の優先順序は数学的な解釈に同じ, 順序を変更するには()でくくる。

列ベクトルの発生

```
>>x=[4;5;2] または >>x=[4;      または >>x[4      とすると ,
                        5;                5
                        2]                2]

x=4
   5
   2
```

継続行の印

matlab の継続行の印は ,文の末尾に置かれた...である .通常青色で...が表示される .

```
例) >>x=[1 5 4 3 6 7 2 1 5 3 ... % 数値の末尾に 空白... を追加
      5 3 2 4 2 3 6 4 3 6];
```

この印は ,...のある文がその下の文に続いていることを意味する .

故に ,上の文を実行すると , x は 1 行 20 列の行ベクトルとなる .

通常 , 文が横長になりすぎて見づらくなるのを避けるため , 或いはプログラムリスト印刷時に見づらい所で改行されるのを避けるため , 継続行の印を用いる .

(注) ...が青色に変化しない場合には , 空白を入れ忘れて最初のピリオドが小数点と解釈されているので , 空白を追加 .

for ~end 文

例題 1 (1+2 + 3 + 4 + 5 + 6+7+8+9+10 の合計計算)

```
% test.m      %はコメント(注釈行)を意味し ,%が付くとプログラムの実行と無関係となり無視される .通常 ,プログラムの簡単な説明を記述(この例はファイル名)
s=0;          ループの外で合計を求める変数名を 0 と置く
for ii=1:10   ループは for と end の間を ii=1,2,3,...,10 と 1 から 10 まで 10 回まわる .
    s=s+ii;   ii=初期値[:増分値]:最終値 増分値が+1 のとき省略して記述可
end          [ ]は省略可能を意味する .
disp(sprintf('s=%d',s)); 結果の表示( disp 文については , 本資料末尾参照)
                文末 ; の有無に拘わらず , disp 文は , 表示命令なので表示
```

例題 2 x(1)+x(2) + x(3) + x(4) + x(5) + x(6)+x(7)+x(8)+x(9)+x(10)の合計計算

```
% total.m
x=[12 31 34 56 15 32 46 25 21 78]; 変数名 x(1)~x(10)に [ ]内の数値を順に代入する .
s=0;
for ii=1: length(x)    length(x)は ,行ベクトルx の長さ(今の例では 10)を返す .
    s=s+x(ii);        x の添え字 ii が 1 ~ 10 まで順に変化 . =の右辺にある変数
end                  には , 事前に数値が与えられていなければならない .
```

```
disp(sprintf('s=%d',s))
```

例題 3 演習問題 2 a=10,d=5,n=100 とした場合の $S = \sum_{i=1}^n (a + (i - 1)d)$ を求めなさい .

上式に i=1 を代入すると , $10+(1-1) \times 5 = 10$

i=2 を代入すると , $10+(2-1) \times 5 = 15$

:

i=100 を代入すると , $10+(100-1) \times 5 = 505$ だから

上式は , $S=10+15+20+25+\dots+505$ の合計を計算することと同じ .

% prob_2.m

```
a=10;
```

a,d の値は一度だけ代入すればよいので , for~end 文の中には入れない .

```
d=5;
```

```
n=100;
```

```
s=0;
```

この文を for~end 文の直前に入れると , 合計計算であることが分かり易い .

```
for ii=1:n
```

```
    s=s+a+(ii-1)*d;
```

下線部に着目 . この例では計算式が記されている . a,d の数値が事前に与

```
end
```

えられているので計算可能 . ii が順次増えると d=5 ずつ下線式の値が増大

```
disp(sprintf('s=%d',s))
```

disp 文

例 1 ii=___, a=___ と表示するには ,

```
s= sprintf('ii=%d, a=%d',ii,a);
```

 このとき , 変数名 s は文字列変数となります .

```
disp(s)
```

sprintf 文は , (' ', 変数名, 変数名) の様に , 表示したい文字列と変換指定子 (書式) を ' ' で囲み , 必要に応じてその後ろにカンマを付けた後 , 変数名を与える . 変数名が複数のときにはそれらをカンマで区切る .

表示したい文字列 (この例では ii= , a= の 3 つ)

変換指定子 (この例では %d が 2 つ) ~ 変換指定子は通常 , 後続の変数の数分必要で , 並び順に変数名と対応している .

変数名 ii,a ~ 事前に数値が保存されていることが前提

または , 1 行にまとめて ,

```
disp(sprintf('ii=%d, a=%d',ii,a))
```

%d の意味は 10 進表現 (符号付き) p.56 表 3.3 参照

例 2 s(1)=___ の様に () 内の数字が自動的に増え , 更に対応する値を表示するには ?

```
s(2)=___
```

```
s(3)=___
```

```
disp(sprintf('s(%d)=%d',ii,s(ii)))
```

変数 ii の値は for~end 文の制御変数に対応させる等して ,

自動的に増えることを前提とし、 s は 1 次元の配列変数で `disp` 文を実行する前に値が $s(1), s(2), s(3) \dots$ に保存されていることを前提としている。必要に応じて 2 つ目の `%d` は、他の変換指定子 (`%f, %e, %g, %s`) に変更する。

例 3 出力結果の行間を 1 行分あけるには、

`disp(sprintf(' '))` ' ' の中に空白を 1 つ入れる。

例 4 改行して出力するには、

`disp(sprintf('x¥ny'))` ¥n は、復帰改行の意味のエスケープ文字 (p.56 表 3.4 参照)

x

y