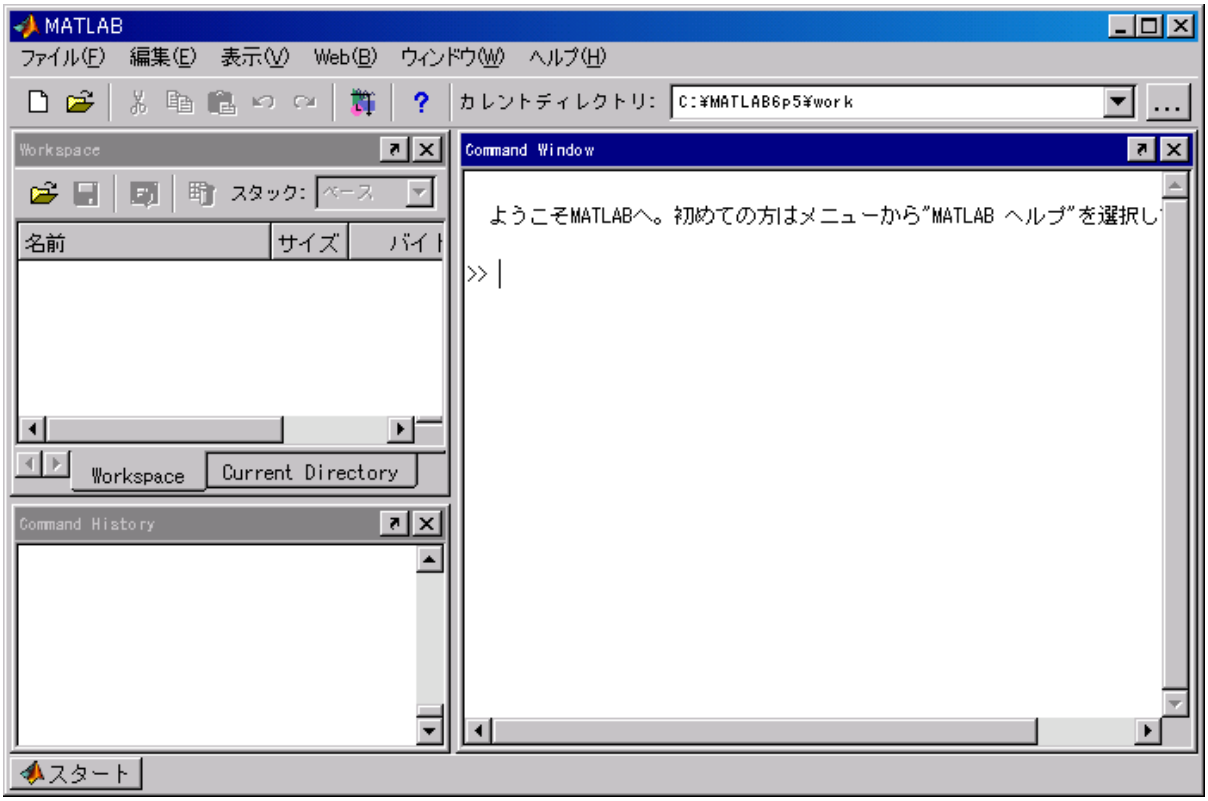
MATLAB(1)

MATLAB とは？

MATLAB とは、数値計算、データビジュアライゼーション、プログラミングなどが可能な工学技術者のためのソフトウェアである。海外では、工学的な計算などにおいて MATLAB は標準的なソフトである。

MATLAB の起動

MATLAB を起動するとまず以下のような MATLAB コマンドウィンドウが表示される.



MATLAB での操作は、このコマンドウィンドウが基本となり、ここにコマンドをタイプすることで様々な計算処理が実行できる．MATLAB コマンドウィンドウを終了する場合には、quit または、exit とタイプすればよい.

単純な数値演算

MATLAB では､起動した時点で簡単なポケコンのような感覚で簡単に数値計算ができる. 例えば、“A 君が文房具に行き、100 円の消しゴムを 3 個、150 円のノートを 3 さつ買いました．1000 円出したらお釣がいくら戻ってくるのでしょう？”というような問題があった場合､MATLAB では、以下のようにタイプすればよい．なお、”>>”は、MATLAB のプロンプトである.

```
>>1000-(100*3+150*3)
ans　=
250
```

と出てくる．消費税（__%）を考慮にしていないのはおかしい．これを直してみよう.
直すには、まず_____を 1 回押す．すると前回打った式が出てくる．このようにして出た式は、ラインエディタのように編集することができる．右左の矢印キーを使いカーソルを移動し以下のように書き換える．これは、前回打ったコマンドを記憶してくれる history 機能と呼ばれ知っていると便利である.

```
》　1000-(100*3+150*3)*1.05
ans =
   212.5000
```

今度は小数点以下の数は払えないというような文句が出た場合にはどうすればよいのか？MATLAB には、help　コマンドというのがある．そこで整数をキーワードにしてみると、

```
》 help integer
integer.m not found.
```

このように出てきてしまい、わからない場合がある．このような場合に便利なコマンドが_____である．_____コマンドは、ヘルプファイルの中身を検索し関連のある記述のあるコマンドを見つけることができる．上の例でのキーワード”整数”を検索してみると．

```
》 lookfor '整数'
BITMAX　 最大の浮動小数点整数
ROUND　 最も近い整数への丸め
BASE2DEC　 B で設定された基底で表わされた数字を 10 進数整数に変換
BIN2DEC　 2 進数文字列を 10 進数整数に変換
```

というのが出てきた．要するに、round　というコマンドが使えそうだということが分かった．もし、更に内容を詳しく調べたい場合には help　round とタイプすればよい．さて、A 君の行った文房具屋さんは、消費税四捨五入の店だったと仮定すると、以下のように計算できる．

```
>> 1000- round((100*3+150*3)*1.05)
ans =
  212
```

となる．MATLAB のコマンドは、全て小文字である．help などコマンドを調べる場合には、大文字で(例えば ROUND)記述されているが実際に使用する場合には**関数名は__文字で書く必要**がある．

変数による入力

MATLAB で変数を使った計算例を見てみよう．

```
>> kesigomu=100
kesigomu =
    100
>> note=150;
>>kekka=kesigomu*3+note*3
kekka =
    750
>>kekka
    750
```

ここでは、３つの MATLAB 変数 kesigomu,　note,　kekka が定義されている．kesigomu の値を代入してリターンを押すと MATLAB はその結果を変数名とその値を再表示している．　しかし、次の note では、その結果は表示されない．MATLAB では、コマンドの最後に**セミコロン（；）を付けると結果の表示を行わない**．また、MATLAB での**変数の定義は、最大 63 文字 MATLAB6.1 までは 31 まで）であり、変数名の大文字、小文字は区別**する．また変数として漢字は使えない．

MATLAB の基本算術演算子

基本的な加減乗除演算子は，C 言語と同様に定義されている．ここで注意すべき点は、割り算の定義である．/（スラッシュ）は、通常の割り算と同じであるが、それ以外にも¥日本のフォントでは円マークであるが、オリジナルでは＼(バックスラッシュ)相当するものが定義されている．(／)スラッシュが A÷B を意味するとすると＼(バックスラッシュ)は_____を意味する．これは、

行列の演算では、掛け算が可換では無いことから来ている.
また，行列以外にもベクトルを扱いやすくする演算子，通常の演算子の前に.（ドット）を付けたドット演算子もある.（具体的な例は，後述する.）

MATLAB ワークスペース

MATLAB は BASIC 言語と同じようにインタプリタ型の言語であり、対話的に使うことができる.
MATLAB は、入力して代入された変数名やその変数の値を MATLAB ワークスペース内に保存している．例えば、前に代入した kekka をチェックしてみると

```
>>kekka
 kekka =
    750
```

となる．ではもし、変数の名前を忘れてしまった場合にはどうすればよいか？
MATLAB には今まで入力した変数名を表示するコマンド_____がある.

```
》 whos
  Name              Size            Bytes  Class
  ans               1x1                 8  double array
  kekka             1x1                 8  double array
  keshigomu         1x1                 8  double array
  note              1x1                 8  double array
Grand total is 4 elements using 32 bytes
》
```

MATLAB 変数の注意事項

<u>関数名と同じ変数名を定義した場合には、変数名が優先</u>される.
例えば、round という変数を定義した場合、_____関数は使用できなくなってしまう．このような場合には、変数を消す_____という命令がある.

```
》 round=3.1;                    ← Round 変数が定義してあるので使えない
》 round(2.1)
Warning: サブスクリプトインデックスは整数値である必要があります.
???  インデックスが行列の次元を超えました
》clear round
》 round(2.1)                    ← Clear コマンドで復活
ans =
      2
》
```

変数を全て消去したい場合には、
単に_____とタイプすればよい．注意すべき点としては、複数の変数を個別に消去したい場合には

_____ round kekka

というように行なう．round と kekka の間は、スペースである．<u>**カンマ(，)やセミコロン（；）を入れるとその変数は消去されない**</u>ので注意すること．（MATLAB 文法での意味が異なる）

コメント文

MATLAB でのコメント文は、%を用いる.

```
>>kekka        %result  of  purchase
```

といった感じである．コメント文は、後述するスクリプトファイルの記述でよく使う.

MATLAB スクリプトファイル

長いコマンドを打ち込む手間を省くために MATLAB では、MATLAB スクリプトファイル（通称 M ファイル）を用意している．MATLAB スクリプトファイルは、拡張子.m が付いたファイルであり、ファイル名がそのまま実行できるコマンド名となる．
MS-DOS などに親しい人には、バッチファイルをイメージしてもらうと分かり易い．MATLAB スクリプトは、テキスト形式で記述する．スクリプトには、通常、MATLAB コマンドウィンドウでタイプするコマンド以外にもプログラミング上必要な制御構造（後述）などを書くことが可能である．コマンドウィンドウ上でファイル名（拡張子.m を除く）をタイプすると、MATLAB は、あたかも MATLAB スクリプトファイル内のコマンドをタイプしたのと同じように実行する．この機能を活用することでユーザが簡単に新しいコマンドを定義できる．この機能は便利な反面、コマンド名（ファイル名）、関数名、また変数名と同じ表記になるため、ときおり混乱を起こす．そのため MATLAB では、以下に示すファイル管理用の関数が用意されている．

MATLAB ファイル管理（関数管理）

MATLAB では、いくつかのファイル管理用のコマンド、例えばファイルのリストを見るコマンドや、ファイルの中身を見るコマンド、消すコマンド、カレントディレクトリやフォルダを見つけるコマンドディレクトリを移動するコマンドなどを用意している．

cd	現在の MATLAB のカレントディレクトリの位置表示する．
cd path	MATLAB のカレントディレクトリの位置を変更する．
delete test	test.m を消去する．
dir	MATLAB のカレントディレクトリ内にあるファイルを表示する．
ls	dir と同じ
path	MATLAB がサーチする関数などの順番を示す．
pwd	cd　と同じ意味、カレントディレクトリの位置表示
type test	test.m　のファイルの内容を表示する．
which test	どのディレクトリにある test を使っているのかを表示する．

MATLAB では関数名や、変数名は表面上同じ表記になっており、ときおり混乱するので注意すること，（もし振る舞いがおかしい場合には，help コマンドでチェックしたり，clear 命令を実行すると良い.）それでは簡単なスクリプトファイルを作って見よう。
まず、ファイルを作る前に、作業用ディレクトリに MATLAB コマンドプロンプトを移動する．MATLAB コマンドプロンプトのカレントディレクトリィの位置は、cd または、pwd で確認することができる．後は、MS－DOS や UNIX でのディレクトリの移動の方法とほぼ同様のコマンドでカレントディレクトリを移動する．下の例では、z:ドライブの作業用ディレクトリ hogehoge に移動した例である．移動した最後に pwd コマンドでカレントディレクトリの位置の確認をしている．

```
》　cd z:¥
》　cd hogehoge
》　pwd
ans =
Z:¥hogehoge
```

ディレクトリ移動の際、ドライブを移動する時、通常 DOS コマンドでは、z：で移動可能であるが MATLAB では、必ず cd z:とする必要があるので注意すること．
スクリプトファイルを作るには、専用のエディタ、または、自分の使い慣れたエディタで編集する．まず、以下のようにタイプし、

```
% test.m Example of Simple Script file
clear
kesigomu=100;
note=150;
kekka= 1000- round((kesigomu*3+note*3)*1.05)
```

このスクリプトを M ファイル　test.m としてディレクトリ z :¥hogehoge に保存する.
MATLAB 上でこの test.m を実行するためには、単純に test とタイプするだけで良い.（動作しない場合には、pwd とタイプし MATLAB コマンドウィンドウのカレントディレクトリが z:¥hogehoge であることを確認すること.）結果は以下のようになる.

```
>> test
kekka =
   212
```

_____関数を使ったスクリプトファイルは以下のようになる.

```
% test2.m simple script file for input function test
clear
katudon=700;
ikutu=input('Enter number of katudon order >  ');
nedan = ikutu*katudon
```

MATLAB ファイル名の注意点

MATLAB でスクリプトファイルを保存する時、Windows の通常のファイルの感覚で保存してしまうとスクリプトファイルとして実行できないため注意が必要である。
実行不可のファイル名の例をいくつか挙げると、

```
1sttest.m  ×　ファイル名の最初が数字から始まっている。
kadai.2.3.m ×　ドットが複数あるファイル名
kadai2-3.m ×　ファイル名に四則演算子が使われている。
kadai(2).m　×　ファイル名にカッコがある。関数と間違えてしまう。
kadai2 3.m　×　ファイル名にスペースがある。
plot.m　　×　すでに関数が存在する。
```

これ以外にも、ファイル名に漢字などを使うとスクリプトファイルとして実行できないので注意する必要がある。
ファイル名として使ってよい文字としては、アンダースコア（シフトキーを押したまま「ろ」を押すと出る文字）がある。

```
kadai2_3.m
```

MATLAB の基本ベクトル演算子

MATLAB では、スカラー量や行列以外にベクトル量が扱える．このベクトル量は、行列量とは若干異なる振る舞いをするが、非常に便利な機能である.
今、サイン関数の半周期の値の計算を考えてみる．y=sin(x)で$0 \le x \le \pi$の範囲での波形である．例えば、0.1π間隔で考えると、x=0,0.1π,...,1.0πとなる．このようなことを MATLAB で x の範囲を記述するには

```
>>x=［0 0.1*pi 0.2*pi 0.3*pi 0.4*pi 0.5*pi 0.6*pi 0.7*pi 0.8*pi 0.9*pi 1.0*pi];
>>y=sin(x)
y =
   Columns 1 through 7
         0    0.3090    0.5878    0.8090    0.9511    1.0000    0.9511
```

```
  Columns 8 through 11
    0.8090  0.5878       0.3090       0.0000
```

となる．**MATLAB のよいところは、ベクトルを作ってしまうと、sin であろうが cos であろうが一気に関数処理ができる点である.**

ベクトルの中の__番目の要素を知りたいと言った場合には、

```
>>y(3)
ans    =
                  0.5878
```

とすればよい．さらに例えば__番目から__番目までを一度に見たい場合には、____演算子を用い、

```
>>y(1:3)
ans =
            0       0.3090       0.5878
```

また、例えば、__から 2 飛びで__までの値を見たい場合には

```
>>y(2:2:7)
ans =
     0.3090       0.8090       1.0000
```

また、コロン演算子を使うとベクトルからの要素を取り出すのと同様に、ベクトルを作ることも可能である．例えば、0 から 1 まで 0.1 ステップのベクトルを作るには_____と書くだけでよい．これを前の問題に応用すると

```
>>x=(0:0.1:1)*pi;
```

とコンパクトに書き換えることが出来る．

ベクトル操作

以上の例では、常にベクトルは 1 行複数列の行列であった．このようなことからこれを横ベクトルと呼ぶが、もちろん、縦ベクトルも扱う事が出来る．縦ベクトルの作り方の簡単な例を挙げてみると、

```
>> c =[1;2;3;4;5]
 c    =
           1
           2
           3
           4
           5
```

とすればよい、行ベクトルを作るときとの違いは、要素の間にスペースでなく、セミコロン（;）を入れている．さらに MATLAB では転置演算子（’）というのが定義されている．

```
>>c=(1:5)’
 c =
           1
           2
           3
           4
           5
```

ベクトルとセミコロンを使って簡単に行列形式に書き直すことが出来る．

例えば、

```
>>g =［1 2 3 4;2 3 4 5］
g =
    1 2 3 4
    2 3 4 5
```

ここで g は、２行と 4 列からなっている．つまり、２×４の行列となる．

スカラーとベクトル/行列の算術演算

始めのベクトルの例で、ベクトル x にスカラー量の pi を乗算した例を挙げた．他の例は、簡単な算術演算とほぼ同様である．さらに、減算、掛け算、割り算はベクトル/行列の全ての要素に対して作用する．

例えば、

```
>> 3*g
ans =
     3     6     9    12
     6     9    12    15
```

ベクトル/行列とベクトル/行列の算術演算

ベクトル同士の算術演算は、スカラーとベクトル/行列の演算ほど単純ではない．異なったサイズのベクトル/行列同士の演算を定義するのは、直感的にわかりづらいからである．MATLAB では、特にベクトル/行列のサイズが同じ場合のみ加算、減算、乗算、割り算がおのおのの要素ごとに対して出来る．例えば、

```
>>h=[3 4 5 6;4 5 6 7];
>>g+h
ans =
     4     6     8    10
     6     8    10    12
>> g-h
    -2    -2    -2    -2
    -2    -2    -2    -2
```

ここで注意しなければいけないのが乗算と割り算である．おのおののベクトルの要素ごとに掛け算や割り算をやる場合には、(.*）や(./)(.¥)といった演算子を用いる必要がある．

```
>>g.* h
ans =
     3     8    15    24
     8    15    24    35
```

ベクトルや行列の大きさを知る方法として、_____コマンドがある．例えば

```
>> size(g)
ans =
     2     4
```

行と列の情報を別々に取り出したい場合には以下のようにする．

```
>>[row col]=size(g);
>>row
row =
     2
>>col
col =
     4
```

ベクトル中の要素のサーチ

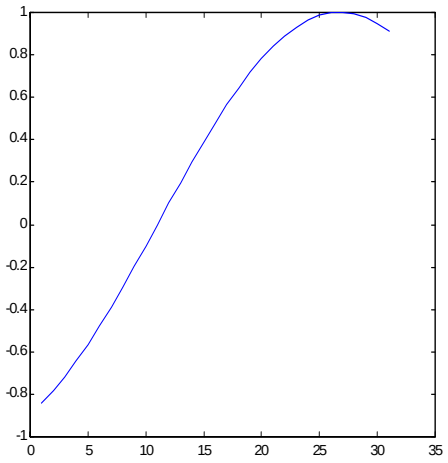
しばしば、ベクトル中の何番目の点が一番大きいのかサーチしたりしたい場合がよくある．（例えば FFT をかけて最大ピーク周波数を調べたい場合など）　MATLAB では、C や Basic などにある FOR 文（後で詳しく説明する）を使う方法もあるが、MATLAB に備わっている<u>ベクトル対応関数を使った方が計算速度の面でかなり有利</u>になる．ここでは、ベクトル対応関数を使った例を説明していく。

<u>最大値の検出</u>

ここでは、sin(-1:0.1:2)のベクトルの中でどの点がピークになるか調べてみよう．まず簡単な sin(-1:0.1:2)のベクトルの形を見てみる、数値ではちょっと大変なのでここでは、plot 関数（plot 関数は後でもっと詳しく説明する）を使うことにする．簡単な使用法は、

```
>>plot(sin(-1:0.1:2))
```

である．すると、以下のようなグラフが現れる．



x 軸はベクトルの要素の数、で y 軸はその大きさを示している．この図から判断すると____から____の間にピークがあるのがわかる．それでは、最大値をサーチする MATLAB のコマンド_____を使ってサーチしてみよう．

```
>>max(sin(-1:0.1:2))
ans =
      0.9996
```

このコマンドでは、最大値（y 軸）はわかるが、ベクトルの要素番号（x 軸）がわからない、要素を知るには、以下のように書き直すとよい．

```
>>[yjiku xjiku]=max(sin(-1:0.1:2))
yjiku =
      0.9996
xjiku =
      27
```

ということで、ベクトルの要素は____番目であることがわかる．
ちなみに、max 関数を使わずに書くと次のようになる。

```
data=sin(-1:0.1:2);
yjiku=data(1);
xjiku=1;
for i=2:length(data)
   if(data(i)>yjiku)
        yjiku = data(i);
        xjiku = i;
   end
end
[yjiku,xjiku]
```

行列操作演算子

元々MATLAB が開発された初期の頃は、単に行列と線形代数の計算を簡単化するために書かれた．その中で最も共通の線形代数の問題は、連立方程式の解法である．例えば、いま

$$\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 0\end{bmatrix}\cdot\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}=\begin{bmatrix}366\\ 804\\ 351\end{bmatrix}$$

このような式があるとする．この方程式を MATLAB で解くには、式の変形をする必要がある．この式の解法は様々な解法がある．例えば、ガウスの消去法を用いる方法や、LU 分解による方法、逆行列を求める方法等である．まずこれらの式を解くために上の式を、A*x=b の式に当てはめてみる．

```
>>A=［1 2 3;4 5 6;7 8 0];
>>b = [366;804;351];   % you can also rewrite   b = [366 804 351]'
```

ここで B 行列は３×１の行列なので数字と数字の間はセミコロンで区切る．別の方法としては、数字と数字の間はスペースで区切り最後にベクトルを転置してもよい. MATLAB には、$Ax=b$ の解法が２つある．最も直感的な方法は、$x=A^{-1}b$ である．

```
>>x=inv(A)*b
x =
          25.0000
          22.0000
          99.0000
```

ここで inv(A)は逆行列を計算する MATLAB 関数である．これ以外の方法としては、以下のものがある．

>>_____
￥円マークは、バックスラッシュとして定義されている．MATLAB でのバックスラッシュは、逆の割り算を意味する．A や B がスカラーの場合には A／B と B＼A は同じ解を意味するが、行列の場合には異なる．ここでは、連立方程式の解の求めかたとして、inv 関数を用いる方法と、¥演算子を用いる方法を説明した．実はこのような問題の場合には、inv 関数を用いるより**¥演算子を使った方が計算精度が__くなる**．　理由は、¥演算子を使った場合には、逆行列を求めず直接解を求めるアルゴリズムが使われるためである．

MATLAB の制御フロー

MATLAB には４つのタイプの制御フロー構造がサポートされている．For 文と、While 文それに_____、switch 文ある．これらのコマンドは複数行にわたることが多いのでスクリプトファイルでかいた方がよい．（コマンドラインでもできないわけではないが）

If 文

```
if(条件)  文;end
```

となる。条件が真、つまりゼロでない時、文が実行される。

```
kesigomu=100;kazu=input('Please input number of purchase of kesigomu > ');cost=kesigomu*kazu;if kazu> 9;cost=kesigomu*kazu*0.8;end;cost

Please input number of purchase of kesigomu > 10
cost   =            800
```

この場合では、___個以上消しゴムを買った場合___%のディスカウントをする例である．
MATLAB には、if-else 文も使える。

```
if(条件)  文 1;
else  文 2;
end
```

となる。条件が真、ゼロでない時、文 1 が実行され、それ以外の時、文 2 が実行される。
この例では、__個以上の消しゴムを買った場合__%のディスカウントをするが、もしそれ未満の場合には、__%の割増料金になる例である。

```
kesigomu=100;
kazu=input('Please input number of purchase of kesigomu > ');
cost=kesigomu*kazu;
if kazu> 9;cost=kesigomu*kazu*0.8;
else cost=kesigomu*kazu*1.5;
end;
cost
```

また、MATLAB では、この構文以外にも if-elseif 文なども使うことができる。

```
tensuu=input('Please input testnoten > ');
if tensuu>= 80; disp('評価は A');
elseif tensuu >= 70;disp('評価は B');
elseif tensuu >= 60;disp('評価は C');
else disp('評価は D');
end
```

テストの点を入力し、80 点以上の場合には、A、それ以外で、70 点以上場合には、B、それ以外で 60 点以上場合には C,それ以外の場合は、D と評価するスクリプトの一例である。この構文を使わなくても elseif 構文を使わずに、else 構文と if 構文を複数使っても同様の記述をすることができる。ただし、その場合には、対応する end 文を複数書く必要が出てくる。

switch 文

elseif 文など複数の if 文を使う場合には、switch 文を使うとすっきり表せる場合が多い。

```
switch(式)
   case 1,文１；
   case 2,文２ ；
    ・・・
```

```
   otherwise, 文ｎ
end
```

MATLAB の switch 文は、C 言語や Java 言語で実装されている switch 文とは若干異なるので注意を要する。MATLAB の switch 文では、case 文は、通常 break 文を明示的に書かないと、次の case 文に書かれている文を実行していしまうが MATLAB の場合は、case 文にかかれた対応する文のみを実行し switch 文から抜ける。また、その他の文は、スペルミスをしやすい default キーワードではなく、othewise をキーワードとして使っている点も異なる。
以下に elseif 文の場合のスクリプトと同様のスクリプトを示す。この場合、点数を入力し、10 で割り算をし、整数化する。この際、整数の計算は floor 関数による切り捨てを行い、elseif 文と同じ動作をするようにしている。

```
tensuu=input('Please input testnoten > ');
tensuu=floor(tensuu/10);
switch(tensuu)
 case 10,disp('評価は A');
 case  9,disp('評価は A');
 case  8,disp('評価は A');
 case  7,disp('評価は B');
 case  6,disp('評価は C');
 otherwise, disp('評価は D');
end
```

for 文

```
for 変数＝初期値：(ステップ数:)最終値
文
end
```

MATLAB での for 文は、if 文と同様に end で囲まれた個所を繰り返し実行する。
変数 i を 1 から 10 まで変化させ表示する for 文の例を示す。

```
for i=1:10;i,end
```

文と文の区切りは、セミコロンまたは、カンマを用いる。カンマで i と end の間にカンマを入れている理由は、セミコロンでは、結果を表示しない。これは、次のように書いた場合と同じである。

```
for i=1:10
 i
end
```

例えば、サンプリング時間 0.1 秒で 1Hz のサイン波のデータを 10 点のベクトルを作ると、
$\omega = 2\pi f$, $\sin(\omega \Delta t \cdot n)$ なので

```
》omega=2*pi*1;deltaT=0.1;
》clear x;for n=1:10;x(n)=sin(omega*deltaT*n);end
》x
x = 0.5878 0.9511 0.9511 0.5878 0.0000 -0.5878 -0.9511 -0.9511 -0.5878 -0.0000
```

と計算できる.
ループ命令は、Basic や C それに Fortran など他の言語で学んだ人には、親しみやすい制御構造であるので多用しがちであるが、実は MATLAB では、ベクトルを用いて同様のことが解決できる場合が多い．例えば、サイン波の例では、

```
》omega=2*pi*1;deltaT=0.1;
》x=sin(omega*deltaT*(1:10))
x = 0.5878 0.9511 0.9511 0.5878 0.0000 -0.5878 -0.9511 -0.9511 -0.5878 -0.0000
```

このようなベクトルを使った方が、直感的でありシンプルである．かつ、**MATLAB ではベクトルを用いた方が計算処理が＿＿くなる．**
MATLAB の for 文では、コロン演算子を 2 つ使い、逆順にリピートすることも可能である。

```
for i=10:-1:1
i
end
```

MATLAB の for 文はベクトルを指定できるため、次のような書き方をしてもよい。

```
for i=[1,2,3,4,5,6,7,8,9.10]
i
end
```

MATLAB の for 文では、C 言語の for 文などは異なり、**繰り返しのための条件比較を行わない**ため、たとえば、

```
for i=[1,-1,2,-2,3,4,-5]
i
end
```

といった形で任意の数値を順序どおりに表示することもできる。ちなみに、MATLAB では、ベクトル以外にも行列を変数として使うこともできる。

```
for i=[1,2,3,5;4,3,4,5]
i
end
```

この場合 i の値は、[1;4],[2;3],[3,4]といった具合に縦方向のベクトルを１つの変数とみなし繰り返し実行される。
また、ループを複数使う、ループの_____も可能である。たとえば、２つのサイコロの目のパターンをすべて生成するには、ループの____を使うことで次のようにして実現できる。

```
for i=1:6
   for j=1:6
     [i,j]
   end
end
```

この場合、重複も含み全ての目の 36 パターンを表示する。ループのネストの応用例として_____の表を表示するには、

```
》for n=1:9;for m=1:9;fprintf(' %3d',n*m);end;fprintf('¥n');end
    1    2    3    4    5    6    7    8    9
    2    4    6    8   10   12   14   16   18
    3    6    9   12   15   18   21   24   27
    4    8   12   16   20   24   28   32   36
    5   10   15   20   25   30   35   40   45
    6   12   18   24   30   36   42   48   54
    7   14   21   28   35   42   49   56   63
    8   16   24   32   40   48   56   64   72
    9   18   27   36   45   54   63   72   81
```

としてできる。
ちなみにこの例では以下のようにベクトル化することで for 文を１つ省略することができる。

```
》for n=1:9;fprintf(' %3d',n*(1:9));fprintf('¥n');end
```

ちなみに、サイコロの目の重複をなくすには、
if 文と continue 文を使うと、次のように書くことができる。

```
for i=1:6
   for j=1:6
      if(i >j) continue;end
      [i,j]
    end
end
```

また、このように書いても同様に表示できる。

```
for i=1:6
   for j=i:6
     [i,j]
end
```

```
end
```

ただし、MATLAB の for 文は、C 言語の for 文とは異なり、**繰り返しのための条件比較を行わない**ため、変数の値を直接変更し、for ループの途中で変更しループ回数を変更させることはできない。下記の例では、変数 i の値を 4 としても一時的に変数の値は変化するが、ループ回数そのものには影響されていない。

```
>>    for i=1:3
   for j=1:3
     [i,j]
end
end

ans =     1     1
ans =     1     2
ans =     1     3
ans =     2     1
ans =     2     2
ans =     2     3
ans =     3     1
ans =     3     2
ans =     3     3
```

```
>>    for i=1:3
   for j=1:3
     [i,j]
     i=4;
end
end
ans =     1     1
ans =     4     2
ans =     4     3
ans =     2     1
ans =     4     2
ans =     4     3
ans =     3     1
ans =     4     2
ans =     4     3
```

While 文

while 文は、for 文と同様に指定した文を繰り返す構文である。

```
while(条件)
文；
end
```

while 文では、条件が真つまりゼロでない場合に限り指定した文を実行する。以下に i=1 として i が 10 になるまで 1 づつ加算し表示する例を示す。

```
i=1;
while(i<=10)
i
i=i+1;
end
```

このような例の場合には、わざわざ while 文を使わなくても for 文で同様のふるまいを技術することができる。

```
for i=1:10
i
end
```

ちなみに、for 文と while 文は、次のような公式により相互に置き換えることができる。

For 文と While 文の表現の違い（ステップ間隔が 1 の場合）

For 文	while 文

for variable=start_num:end_num
%リピートさせたいコマンドを入れる.
end
	variable=start_num;
	while(variable < end_num+1)
	%リピートさせたいコマンドを入れる.
	variable=variable+1;
	end

for i=1:10
　disp(i);
end
	i=1;
	while(i < 10+1)
	　disp(i);
	　i=i+1;
	end

ただし、MATLAB での For 文では、ベクトルを変数とするため有限の繰り返しループしか構成できないのに対し、while 文では、while(1)と書くことで無限ループを構成することが出来る点が他の言語（例えば、C など）とは異なる。

演習問題

１．以下のスクリプトを実行，結果，HELP を用い意味を調べ書け

```
》2.1+3.2*4.3+5.4/6.5-7.6^8
》sqrt(2)
》abs(2+3i)
》close all;x=0:0.01:2*pi;plot(x,sin(x))
》close all;[xx,yy]=meshgrid(0:0.1:2*pi,0:0.1:2*pi);mesh(sin(xx).*cos(yy))
》close all;data = 1./(1:30);plot(data);
》close all;plot(data,'x');
》close all;plot(data,'x');axis([1 30 0 1])
》close all;plot(data,'linewidth',10);axis([1 30 0 1]);
》close all;plot(x,sin(x));axis([0 2*pi -1 1]);xlabel('¥it{x}');ylabel('¥it{y(t)}');
》close all;plot(x+sin(2*pi*x))
》[2 3;3 -2]¥[7;11]
》roots([1 0 3 5 -7])
》roots([1 0 0 3 0 5 -7])
》close all;w0=2*pi;zeta0=2/10;step(w0*w0,[1 2*zeta0*w0 w0*w0],0:0.1:10)
```

2. input 文で商品の値段を入力し消費税込みの値段を計算するスクリプトを書け.

3. $A=\begin{bmatrix}-5 & 1 & 0\\ 0 & -2 & 1\\ 0 & 0 & 1\end{bmatrix}$, $b=\begin{bmatrix}0\\0\\1\end{bmatrix}$, $c=\begin{bmatrix}-1 & 1 & 0\end{bmatrix}$のベクトル・行列を定義した時,

(a) $Ax=b$ としたときの x を求めよ

(b) $yA=c$ としたときの y を求めよ

4. 時系列データ data=sin((1:1024)*2*pi/1024*gakusei)を定義したとき，最初のピークとなる数値の値と何番目の値かをもとめよ．なお gakusei は，学生証番号下 1 桁+1 の数値を使う.

5. while 文を用い，九九の表を表示せよ？