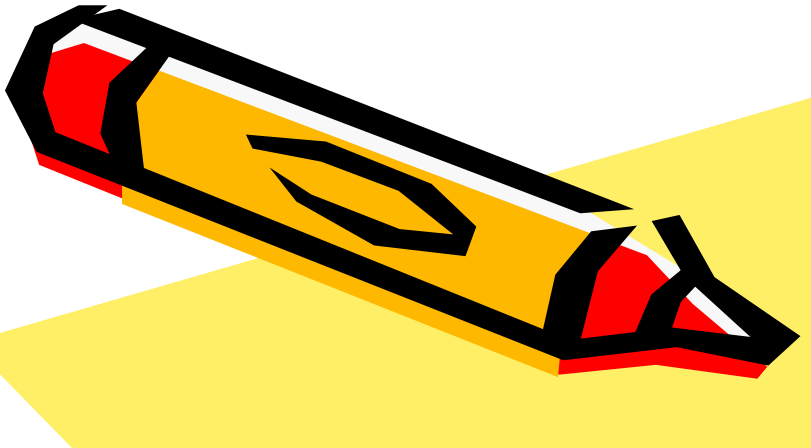
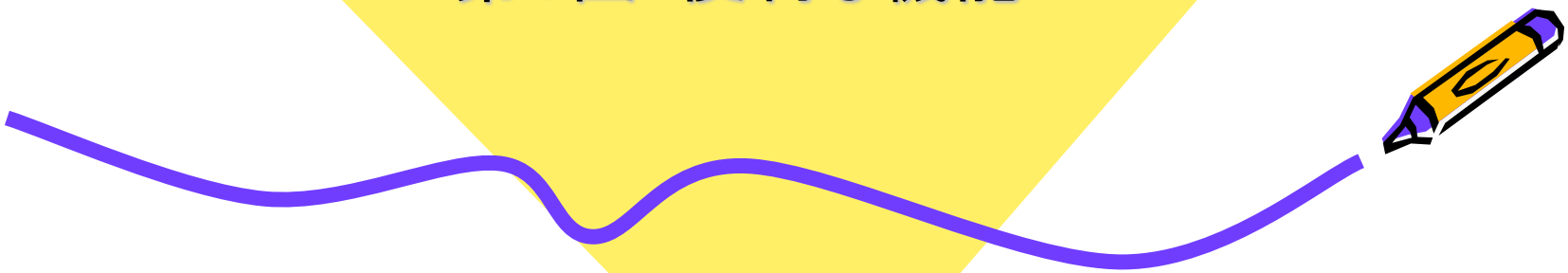


MATLABの使い方

第8回：便利な機能

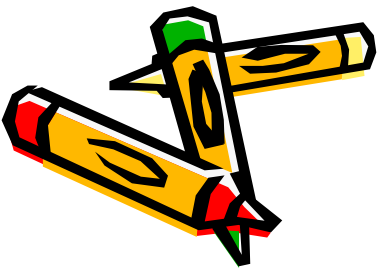


まとめページ：
http://www.nuee.nagoya-u.ac.jp/labs/plaene/koukai/purakaku85/tsukaikata/

MATLABの便利機能

- ブレークポイント(break point)
  - ・・・デバッグに重宝
- プロファイラ
  - ・・・計算時間かかっている箇所を調査
- 自動実行ファイル（startup.m，finish.m）
  - ・・・MATLAB起動時or終了時に実行

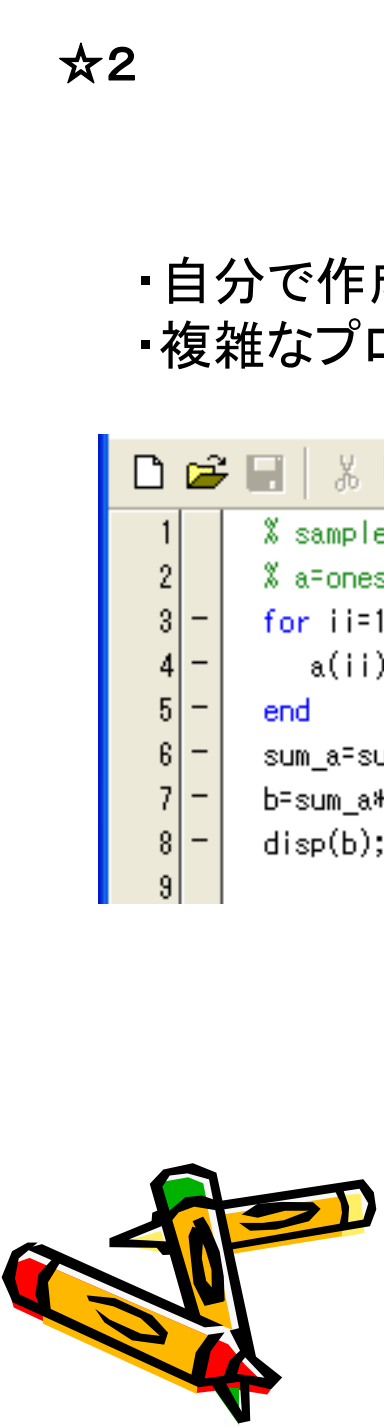
特に、「ブレークポイント」と「startup.m」は使いこなそう！



☆2

ブレークポイント1

・自分で作成したプログラムのデバッグ時に使うと便利
・複雑なプログラムほど効果を発揮



```
1   % sample_break
2   % a=ones(10,2); を入力した後実行するとエラー
3   for ii=1:5
4      a(ii)=ii;
5   end
6   sum_a=sum(a);
7   b=sum_a*sum_a;
8   disp(b);
9
```

```
>> sample_break
   225

>> a=ones(10,2);
>> sample_break
??? エラー: ==> *
内部行列の次元は同じである必要があります

エラー: ==> C:¥Documents and Settings¥admin¥My Documents¥Dropbox¥sample_break.m
行番号: 7  ==>b=sum_a*sum_a;
```

例題

(1+2+3+4+5)×(1+2+3+4+5)=255

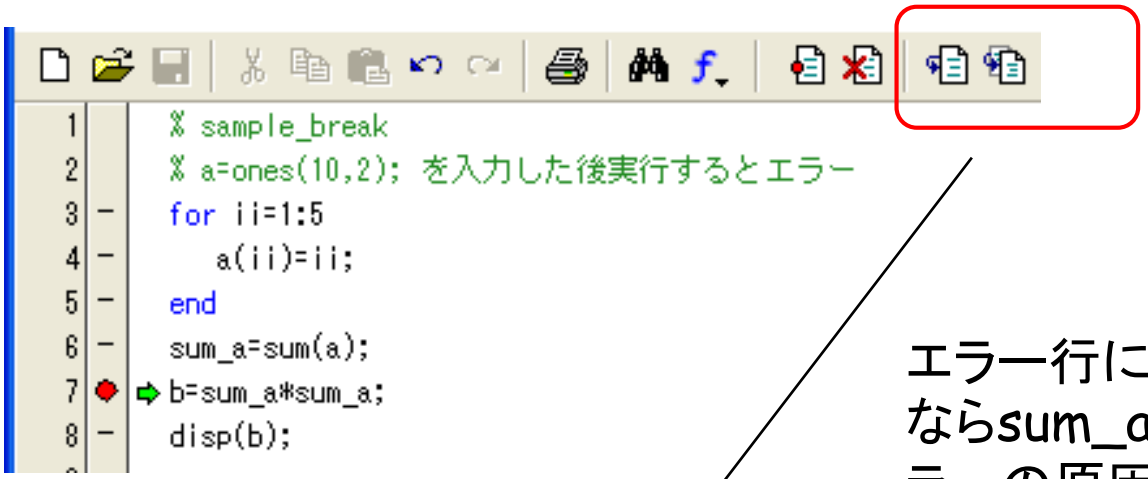
を返す(無駄の多い)プログラム

普通に実行するとエラーしないが、あらかじめaに複数列代入されているとエラーする(エラー行は7)

☆2

ブレークポイント2

・使うときは行番号の右をクリック
・実行するとプログラムが指定行直前でストップ




```
>> sample_break
K>> sum_a

sum_a =

    20     10

K>>
```

K:デバッガモード
（ctrl+Cで脱出可）

エラー行に含まれている変数（今ならsum_a）の中身を確認してエラーの原因を調べる

ステップ：1行ずつ実行
ステップイン：その行のfunction内に移動
ステップアウト：functionから脱出
継続：次のブレークポイントまで実行

dbstop: ブレークポイントの設定

☆1

プロファイラ

profview: プロファイラの起動

・profviewコマンドなどで実行

・プログラム内でどこに時間がかかっているかわかる

```
1   % sample_prof
2   a=[];
3   for ii=1:1e4
4       a=[a,ii];
5   end
6   pause(1e-1);
```

変数の代入と一時待機のプログラム

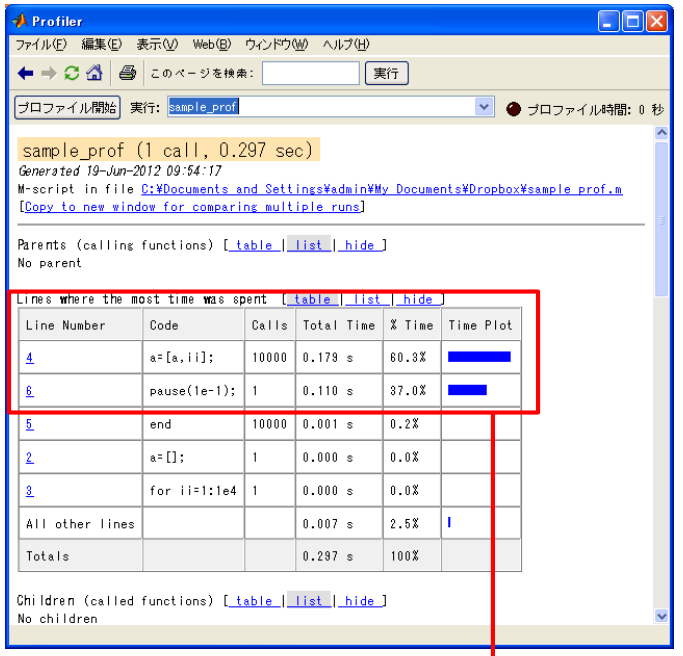
全計算時間はtic, tocで確認

```
>> tic; sample_prof; toc;

elapsed_time =

    0.3290
```

プロファイラ画面



Line Number	Code	Calls	Total Time	% Time	Time Plot
4	a=[a,ii];	10000	0.179 s	60.3%	
6	pause(1e-1);	1	0.110 s	37.0%	
5	end	10000	0.001 s	0.2%	
2	a=[];	1	0.000 s	0.0%	
3	for ii=1:1e4	1	0.000 s	0.0%	
All other lines			0.007 s	2.5%	
Totals			0.297 s	100%	

☆3

自動実行ファイル

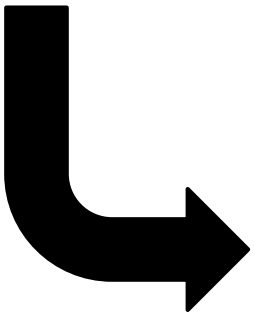
・startup.mはMATLAB起動時に起動

・finish.mはMATLAB終了時に起動

・**いつも行う操作**を記述しておくと便利

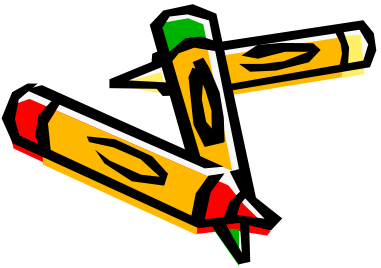
MATLAB起動時のカレントディレクトリに入れておく

MATLAB終了時のカレントディレクトリに入れておく



特に

- ○ カレントディレクトリ移動
- ○ 他のプログラムの実行
- ○ サーチパス登録
- ○ グラフィックスプロパティ設定　etc…



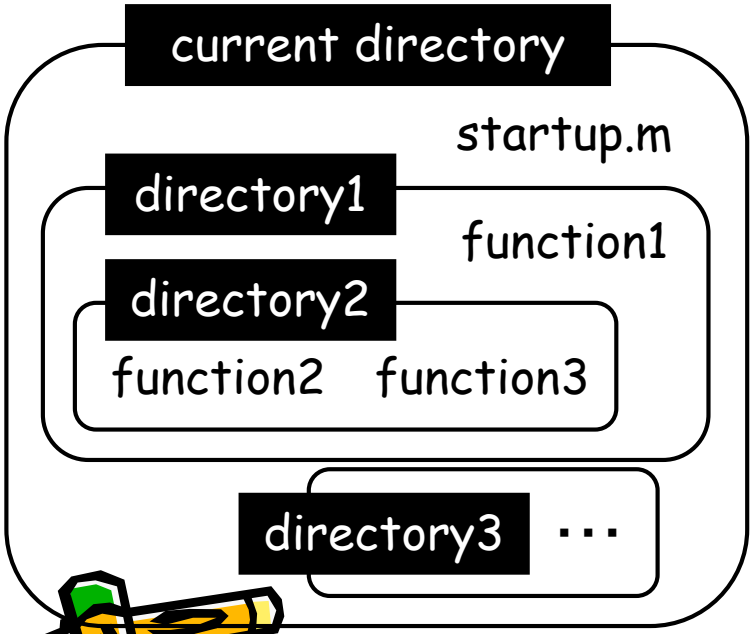
サーチパス

pass: サーチパスの取得

addpath: サーチパスの追加

genpath: ディレクトリ内のパスの文字列を作成

・MATLABではコマンド実行時に「サーチパスに登録済みのディレクトリ内に含まれるプログラム」＋「カレントディレクトリ内のプログラム」が実行される
・汎用的な自作関数をディレクトリでわけて整理する場合にサーチパス設定は有効



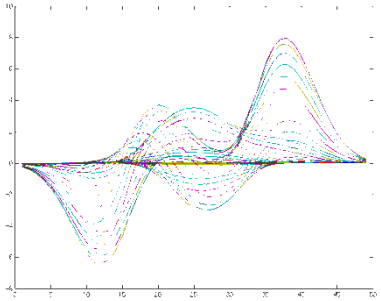

```
adddir={'directory1' 'directory3'};
for ii=1:length(adddir)
    addpath(genpath([pwd '¥' adddir{ii}]));
end
```

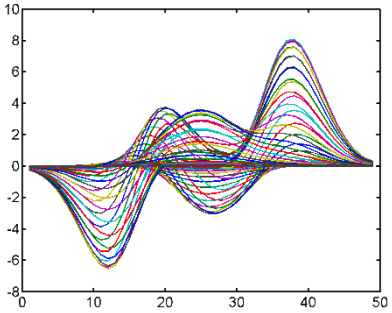
例えば上記のような記述をstartup.mに書いておけば、directory1-3は全てサーチパスに登録され、下流のプログラムは全てどこからでも実行できるようになる

☆2

グラフィックスプロパティ

・set(0,'Default~~~)によりデフォルトのグラフィックスプロパティを変更可能
・MATLAB終了時には変更はリセットされるため、startup.mに記述しておくと違和感なく使用できる





一例

```
set(0,'DefaultAxesLineWidth',2,...
    'DefaultLineLineWidth',1.5,...
    'DefaultLineMarkerSize',12,...
    'DefaultUicontrolFontName','Helvetica',...
    'DefaultUicontrolFontSize',18,...
    'DefaultAxesFontName','Helvetica',...
    'DefaultAxesFontSize',18,...
    'DefaultTextFontName','Helvetica',...
    'DefaultTextFontSize',18);
```