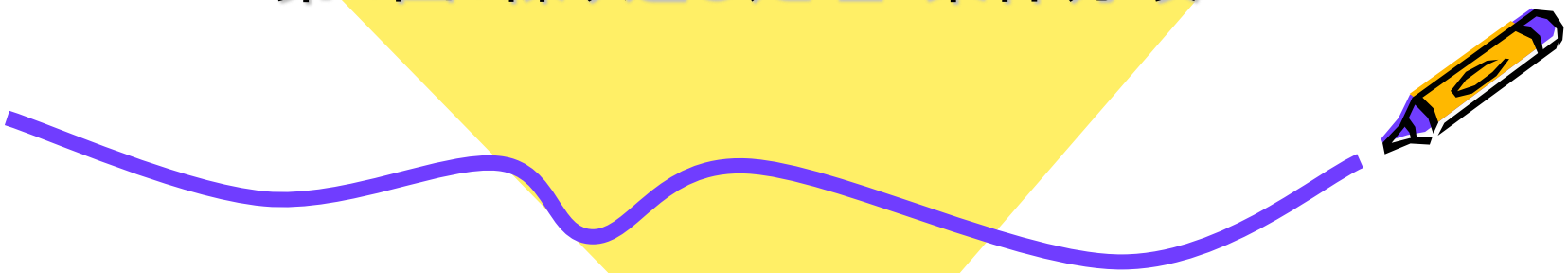
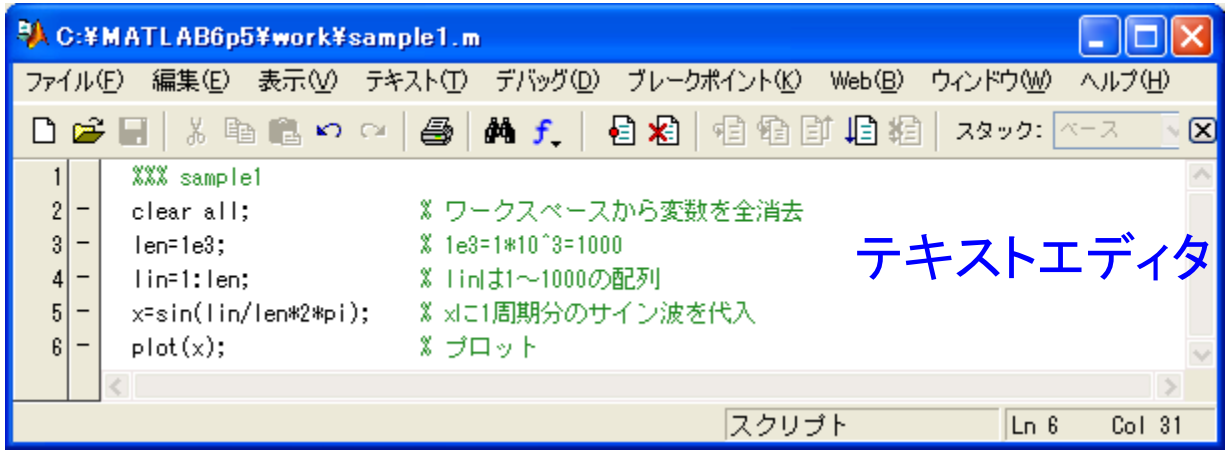
MATLABの使い方

第4回：繰り返し処理・条件分岐



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

第3回の復習



テキストエディタ

・MATLAB付属エディタの使用

・スクリプト M-ファイルの作成と実行

今回は

このエディタを使って、C言語などでよく使われるfor, while, if, switchなどの繰り返し処理・条件分岐の関数の使用法を覚える

http://www.mathworks.com/access/helpdesk_ja_JP/help/techdoc/index.html?/access/helpdesk_ja_JP/help/techdoc/matlab_prog/brqy1c1-1.html&http://www.google.co.jp/search?hl=ja&num=50&q=matlab+for+while&lr=&aq=0&oq=MATLAB+for+w

☆3

for文

・一般的な繰り返し文（for~end）

・forの右に書かれた横配列を順に読み上げ

C:¥MATLAB6p5¥work¥sample_for.m

ファイル(F)　編集(E)　表示(V)　テキスト(T)　デバッグ(D)
ヘルプ(H)

```
1   % sample_for
2   clear all;
3   state={'slow' 'medium' 'fast'};
4   for i=1:3
5       disp(state{i});
6   end
```

sample_for.mの名前で保存

↓

エディタでsample_forを実行、
またはF5、実行アイコン（  ）

```
>> sample_for
slow
medium
fast
```

while文

・whileの右に書かれた文が真の間繰り返し（while~end）

・誤差の収束などに利用

C:¥MATLAB6p5¥work¥sample_while.m

ファイル(F)　編集(E)　表示(V)　テキスト(T)　デバッグ(
ヘルプ(H)

```
1   % sample_while
2 - clear all;
3 - s=0; i=0;
4 - while s<=3
5 -     i=i+1; s=s+i;
6 - end
7 - s,i
```

```
>> sample_while

s =

     6

i =

     3
```

- i=0, s=0　・・・s<=3は真
- i=1, s=1　・・・s<=3は真
- i=2, s=1+2=3　・・・s<=3は真
- i=3, s=3+3=6　・・・s<=3は偽

ループにはまった場合はCtrl+Cで強制脱出

if文

C:¥MATLAB6p5¥work¥sample_if.m

ファイル(F) 編集(E) 表示(V) テキスト(T) デバッグ(D) ブ
ヘルプ(H)

```
1   % sample_if
2   clear all;
3   state=3
4   if state==1
5       disp('Yes');
6   elseif state==2
7       disp('No');
8   elseif state==3
9       warning('caution'); % 警告メッセージ
10  else
11      error('bad'); % エラーメッセージ
12  end
```

```
>> sample_if

state =

     3

警告: caution
> In C:¥MATLAB6p5¥work¥sample_if.m at line 9
```

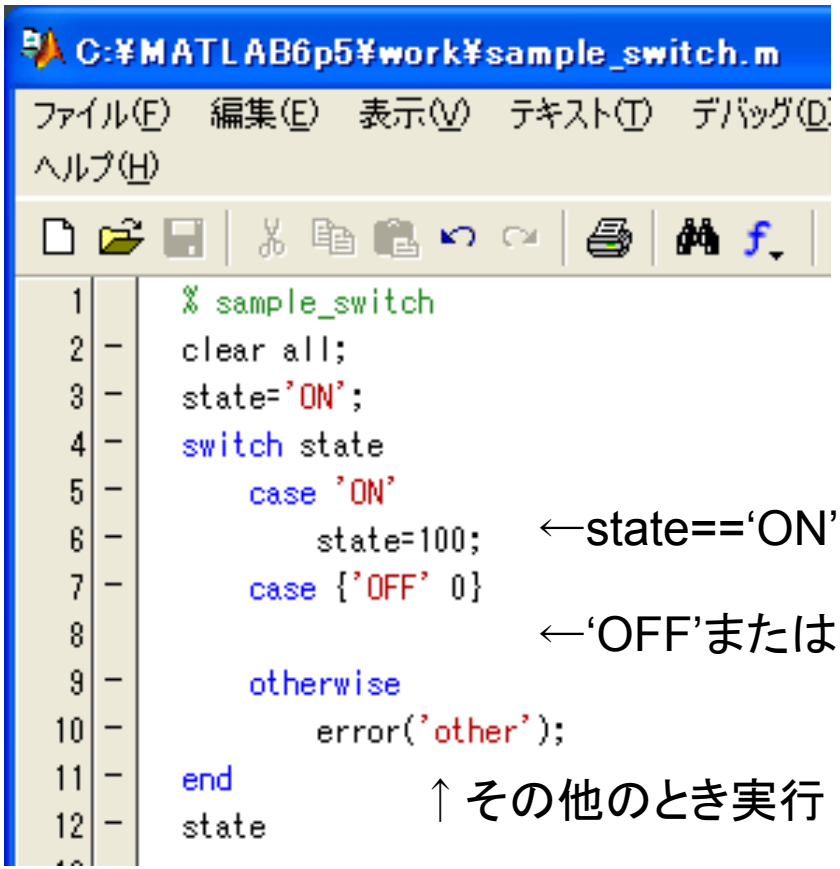
・ifの右文が真であれば実行
（if, elseif, elseif, …, else, end）

・for文、breakと組み合わせて繰り返し処理の中断に使用可能

9行目のwarningによる警告

☆3

switch文



←state=='ON'のとき実行

←'OFF'または0のとき実行

↑その他のとき実行

・switchの右の変数がcaseの右の値と一致したとき実行(switch, case, case, …, otherwise, end)

・条件の変数として、長さの異なるキャラクタ配列や数値配列との混合が可能

➡

```
>> sample_switch

state =

   100
```

☆1

try/catch文

C:¥MATLAB6p5¥work¥sample_try.m

ファイル(F) 編集(E) 表示(V) テキスト(T) デバ
ヘルプ(H)

```
1  % sample_try
2  clear all;
3  try
4      a=a*3;
5  catch
6      mes=lasterr
7      a=3;
8  end
9  a
```

・一部言語には無い例外処理の構文、(try,catch,end)

・try後の文をまず実行

・try後の文中でエラーが起きればエラーメッセージをlasterrに代入しcatch後の文を実行

```
>> sample_try

mes =

'a' は未定義の関数、または変数です.


a =

     3
```

無理やりプログラムを
動かすときに便利

繰り返し代入処理

方法①：配列への代入
※配列サイズが既知のとき

```
1  % sample_roop1
2  clear all;
3  len=10;
4  f=zeros(len,1); % 配列の確保
5  for i=1:len
6      f(i,1)=i^2;
7  end
8  f
```

```
>> sample_roop1

f =

     1
     4
     9
    16
    25
    36
    49
    64
    81
   100
```

プログラムの高速化、
エラー回避のため重要

方法②：空行列への連結
※配列サイズが既知でないとき

```
1  % sample_roop2
2  clear all;
3  len=10;
4  f=[]; % 空行列の作成
5  for i=1:len
6      f=[f;i^2];
7  end
8  f
```

```
>> sample_roop2

f =

     1
     4
     9
    16
    25
    36
    49
    64
    81
   100
```

・原則、方法①を用いる

・今回の例はfor文よりベクトル演算（f=(1:len)'.^2）のほうが有利