

MATLAB クイックリファレンス

配列、行列、プロット、ファイルI/O、制御フロー

基本

コマンドウィンドウ

```
x = 5; % assign (semicolon suppresses output)
x = 5 % assign and display result
disp('Hello') % print to console
clc % clear command window
clear % clear all variables
```

ヘルプと情報

```
help sin % quick help for function
doc sin % open documentation
who % list variables in workspace
whos % list with details (size, type)
```

演算子

```
+ - * / ^  算術演算 (行列演算)
.* ./ .^  要素ごとの演算
== ~== < > <= >= 比較演算子
&& || ~ 論理AND、OR、NOT (スカラー)
& | ~ 要素ごとの論理演算 (配列)
```

変数と型

数値型

```
x = 3.14; % double (default)
n = int32(42); % 32-bit integer
z = 2 + 3i; % complex number
tf = true; % logical
```

型チェック

```
class(x) 型名を文字列として返す
isa(x, 'double') xが特定の型かどうかをチェック
isnumeric(x) 数値型ならtrue
ischar(x) 文字配列ならtrue
islogical(x) 論理型ならtrue
```

特殊定数

```
pi 3.14159...
Inf / -Inf 無限大
NaN 非数
eps マシンイプシロン (~2.2e-16)
i / j 虚数単位
```

配列と行列

配列の作成

```
v = [1 2 3 4 5]; % row vector
v = [1; 2; 3]; % column vector
A = [1 2; 3 4]; % 2x2 matrix
r = 1:5; % [1 2 3 4 5]
r = 0:0.5:2; % [0 0.5 1 1.5 2]
```

組み込みコンストラクター

```
zeros(3) % 3x3 of zeros
ones(2, 4) % 2x4 of ones
eye(3) % 3x3 identity
rand(2, 3) % 2x3 uniform random
linspace(0,1,5) % 5 evenly spaced [0..1]
```

インデックスとスライス

```
A(2, 3) % row 2, col 3
A(1, :) % entire first row
A(:, 2) % entire second column
A(1:2, 1:2) % submatrix
A(end, :) % last row
```

行列演算

```
A' 転置 (共役付き)
A.' 転置 (共役なし)
inv(A) 逆行列
det(A) 行列式
eig(A) 固有値と固有ベクトル
A \ b Ax=bを解く
size(A) 次元 (行数 列数)
numel(A) 要素の総数
```

制御フロー

if / elseif / else

```
if x > 0
    disp('positive')
elseif x == 0
    disp('zero')
else
    disp('negative')
end
```

for と while

```
for i = 1:10
    fprintf('i = %d\n', i);
end
while x > 0
    x = x - 1;
end
```

switch

```
switch grade
    case 'A'
        disp('Excellent')
    case {'B', 'C'}
        disp('Good')
    otherwise
        disp('Try harder!')
end
```

ループ制御

```
break 最内ループを抜ける
continue 次のイテレーションにスキップ
return 関数から即座に抜ける
```

関数

関数ファイル

```
% Save as myfunc.m
function result = myfunc(x, y)
    result = x.^2 + y.^2;
end
```

複数の出力

```
function [m, mx] = minmax(v)
    mn = min(v);
    mx = max(v);
end
[lo, hi] = minmax([3 1 4 1 5]);
```

無名関数

```
f = @(x) x.^2 + 1; % returns 10
g = @(x,y) x + y; % apply to each element
arrayfun(f, [1 2 3])
```

便利な組み込み関数

```
sum(v) 要素の合計
mean(v) 平均値
max(v) / min(v) 最大値 / 最小値
sort(v) 昇順ソート
find(v > 3) 条件が真になるインデックス
length(v) ベクトルの長さ
```

プロット

2D プロット

```
x = 0:0.1:2*pi;
plot(x, sin(x), 'r-', 'LineWidth', 2)
xlabel('x'); ylabel('sin(x)')
title('Sine Wave'); grid on
legend('sin(x)')
```

複数のプロット

```
hold on
plot(x, sin(x), 'b-')
plot(x, cos(x), 'r--')
hold off
subplot(1,2,1); plot(x, sin(x))
subplot(1,2,2); plot(x, cos(x))
```

その他のプロット種類

```
bar(x, y) 棒グラフ
histogram(data) ヒストグラム
scatter(x, y) 散布図
pie(data) 円グラフ
surf(X, Y, Z) 3D サーフェスプロット
imagesc(A) 行列を画像として表示
```

図の保存

```
saveas(gcf, 'plot.png')
exportgraphics(gcf, 'plot.pdf')
```

ファイルI/O

テキストファイル

```
data = readmatrix('data.csv');
writematrix(A, 'output.csv')
T = readtable('data.csv');
writetable(T, 'output.csv')
```

MAT ファイル

```
save('workspace.mat') % save all variables
save('data.mat', 'x', 'y') % save specific vars
load('data.mat') % load into workspace
S = load('data.mat'); % load into struct
```

低レベルファイルI/O

```
fid = fopen('log.txt', 'w');
fprintf(fid, 'Value: %f\n', 3.14);
fclose(fid);
lines = readlines('log.txt');
```

文字列操作

string と char 配列

```
s = 'Hello'; % string (double quotes)
c = 'Hello'; % char array (single quotes)
s + ' World' % 'Hello World' (string)
[C, ' World'] % 'Hello World' (char concat)
```

文字列関数

```
strlength(s) 文字列の長さ
upper(s) / lower(s) 大文字 / 小文字変換
contains(s, pat) パターンが見つければtrue
replace(s, old, new) 部分文字列を置換
split(s, delim) 配列に分割
join(arr, delim) 文字列配列を結合
strip(s) 前後の空白を除去
```

書式化

```
sprintf('x = %2f', 3.14159) % "x = 3.14"
fprintf('i = %d\n', 42) % print to console
num2str(3.14) % number to string
str2double("3.14") % string to number
```

セルと構造体

セル配列

```
C = {'hello', [1 2 3]}; % mixed types
C{2} % access: 'hello'
C(end+1) = true; % append element
cellfun(@length, C) % apply func to each
```

構造体

```
s.name = 'Alice';
s.age = 30;
s.scores = [90 85 92];
fieldnames(s) % {'name', 'age', 'scores'}
rmfield(s, 'age') % remove field
```

構造体配列

```
people(1).name = 'Alice'; people(1).age = 30;
people(2).name = 'Bob'; people(2).age = 25;
{people.name} % {'Alice', 'Bob'}
{people.age} % [30, 25]
```

よく使うパターン

ベクトル化演算

```
% Avoid loops - use vectorization
v = 1:1000;
result = sum(v.^2); % fast
idx = v(v > 500 & v < 600); % logical indexing
```

テーブル操作

```
T = table([25;30], ["A";"B"], 'VariableNames', ...
    {'Age','Grade'});
T.Age % access column
T(T.Age > 25, :) % filter rows
```

エラー処理

```
try
    result = riskyFunction(x);
catch ME
    fprintf('Error: %s\n', ME.message);
end
```

コードの計測

```
tic
heavyComputation();
toc % prints elapsed time
```