

宣言と代入

% 文字式は `syms` で宣言する

```
syms x y
```

% 関数は定義した文字式を使って定義できる

```
f = x^2 + y
```

$$f = x^2 + y$$

% `subs` を使って文字に数値を代入できる

```
subs( f, [x,y], [2,1] )
```

```
ans = 5
```

% 関数 `f` を具体的には定義せず、`x,y` に依存することだけ宣言する

```
syms f(x,y)
```

% 出力を見るとあたかも `f(1,2)` によって代入できるように思える

```
f
```

$$f(x, y) = f(x, y)$$

% `f(1,2)` は実はエラーを出さない

```
f(1,2)
```

```
ans = f(1,2)
```

% しかし、具体的に表現を与えるとエラーがでる

```
f = x^2 + y
```

$$f = x^2 + y$$

```
try
```

```
    f(1,2)
```

```
catch ME
```

```
    ME.message
```

```
    fprintf('f(1,2) によっては代入できない。行列アクセスのように挙動する\n')
```

```
end
```

```
ans =
```

```
'添字が範囲外です。'
```

```
f(1,2) によっては代入できない。行列アクセスのように挙動する
```

% 1 変数の場合も同様にエラーがでる

```
f = x^2
```

$$f = x^2$$

```
try
```

```

    f(2)
catch ME
    ME.message
    fprintf('f(2) によっては代入できない。行列アクセスのように挙動する\n')
end

```

```

ans =
'Index exceeds the number of array elements. Index must not exceed 1.'
f(2) によっては代入できない。行列アクセスのように挙動する

```

% z を t の関数とする

```
syms z(t)
```

% z(t) によって関数 f を定義してみる

% f(2) が計算できるように思える

```
f = z^2
```

$$f(t) = z(t)^2$$

% 実は f(2) は計算できる

% z(2) のような挙動をする

```
f(2)
```

```
ans = z(2)^2
```

% 前に宣言した x の関数としても問題ない

```
syms z(x)
```

```
f = z^2
```

$$f(x) = z(x)^2$$

```
f(3)
```

```
ans = z(3)^2
```

% しかし、直接数値が出るような状況ではエラーが出る

```
syms f(x)
```

```
f = x^2
```

$$f = x^2$$

```
try
```

```
    f(3)
```

```
catch ME
```

```
    ME.message
```

```
    fprintf('f(3) によっては代入できない。行列アクセスのように挙動する\n')
```

```
end
```

```

ans =
'Index exceeds the number of array elements. Index must not exceed 1.'
f(3) によっては代入できない。行列アクセスのように挙動する

```

宣言と仮定

```
% theta を角度, r を距離として宣言し、位置ベクトルを横ベクトルで定義する
syms theta r
position = r * [cos(theta), sin(theta)];
```

```
% 位置ベクトルの自身との内積を計算する。 r^2 を期待する
% でもそうはならない
position * position'
```

$$\text{ans} = r \cos(\bar{\theta}) \bar{r} \cos(\theta) + r \sin(\bar{\theta}) \bar{r} \sin(\theta)$$

```
% conj(r), conj(theta) は複素数を考慮している
% そこで r, theta を実数だと仮定して計算する
assume( theta, 'real' )
assume( r, 'real' )
position * position'
```

$$\text{ans} = r^2 \cos(\theta)^2 + r^2 \sin(\theta)^2$$

```
% cos(theta)^2 + sin(theta)^2 = 1 は勝手に反映されてほしい
% simplify によって単純化すると反映される
simplify( position * position' )
```

$$\text{ans} = r^2$$

```
% 実は syms の時点で実数だと仮定できる
syms theta r real
position = r * [cos(theta), sin(theta)];
simplify( position * position' )
```

$$\text{ans} = r^2$$

```
% 何かのパラメータである文字には仮定を適用できない
syms theta(t) r(t) real
```

警告: 仮定を実行できるのは変数名のみで、'theta(t)' には実行できません。
警告: 仮定を実行できるのは変数名のみで、'r(t)' には実行できません。

```
position = r * [cos(theta), sin(theta)];
simplify( position * position' )
```

$$\text{ans}(t) = |r(t)|^2 \cos(\overline{\theta(t)} - \theta(t))$$

```
% 文字に仮定が設定できないのは微分を考慮する際に面倒になる。
% 詳しくは TODO:replace<微分に関するファイル> を参照
```