

## 最初の一步一代入,四則演算とevalf—

Copyright ©2006 by Shigeto R. Nishitani

### ▼ 簡単な代入

変数への数値の代入は:=を使う。

```
> a:=10;
a:=10 (1.1.1)
```

式の定義も同様。

```
> f:=-a*m;
f:=-10m (1.1.2)
```

以下は $a*x...=c*x^2...$ という式をeq1として定義している。

```
> eq1:=a*x+b=c*x^2+d*x+e;
eq1:=10x+b=cx^2+dx+e (1.1.3)
```

### ▼ 初期化

数値をいれたすべての変数をなにも入れていない状態に一度に戻すには、

```
> restart;
とする。ひとつの変数だけを元に戻すには、シングルクォート'でくくる。
> a:='a';
a:=a (1.2.1)
```

### ▼ 一時的代入(subs)

一時的な代入はsubsでおこなう。

```
> force:=-accel*mass;
subs(accel=10,mass=14,force);
force:=-accel mass
-140 (1.3.1)
```

この場合、

```
> force;
-accel mass (1.3.2)
```

と、それぞれの変数に数値が代入されるのではなく、変数のまま残る。逆に一時的に数値ではなく変数を代入するにはシングルクォートを使う。

```
> x:=2;y:=3;
x:=2 (1.3.3)
```

```
y:=3 (1.3.3)
```

```
> f:='x+y';g:=x+y;
f:=x+y
g:=5 (1.3.4)
```

### ▼ 四則演算とevalf

四則演算は+\*/である。ただし、割り切れない割り算は有理数で表示される。

```
> 3/4;
3/4 (1.4.1)
```

これを小数点数にするには、浮動小数点数に変換する関数evalfに入れる。

```
> evalf(3/4);
0.7500000000 (1.4.2)
```

桁数を指定することもできる。以下では $\pi$ を100桁表示させている。

```
> evalf[100](Pi);
3.14159265358979323846264338327950288419716939937510\
5820974944592307816406286208998628034825342117\
068 (1.4.3)
```

### ▼ 例題

$\exp(-a*x)$ をf1として定義し、 $a=3$ を一時的に代入した場合、どのような表式となるか。

```
> restart;
f1:=exp(-a*x);
f1:=e-ax (2.1)
```

```
> subs(a=3,f1);
e-3x (2.2)
```

### ▼ 演習

$a=3$ ,  $b=4$ として $a,b$ の四則演算をおこなえ。べき乗 $a^b$ を試せ。

$eq1=x+3$ ,  $a=4$ とした場合の $eq1*a,eq1/a$ を求めよ。