

Programming—for-loopの基本技—

Copyright ©2006 by Shigeto R. Nishitani

▼ ランダムな配列の生成

1から100までの整数5個からなる配列の生成.

```
> restart;
roll:=rand(1..100);
n:=5;
A:=Array([seq(roll(),i=1..n)]);
A := [ 93 45 96 6 98 ] (2.1)
```

```
> A:=[93,45,96,6,98];
としても同じだが、この後、色々なランダム要素を使うので、ここで馴染んでおいて。
A := [93,45,96,6,98] (2.2)
```

▼ 要素数の取り出し

for-loopで配列を使うときには、配列の大きさ(要素数)がfor-loopの終了条件になることが多い。

```
リスト構造では単純にnopsとすればよい。
> nops(convert(A,list));
5 (3.1)
```

Arrayの場合は、厳密には最初と最後が明示されている。これを取り出すには以下の通りにする。

```
> ArrayDims(A);
op(ArrayDims(A))[2];
op(ArrayDims(A))[2]-op(ArrayDims(A))[1]+1;
1..5
5
5 (3.2)
```

ただし、listは不特定の要素数、Arrayは特定の要素数に対する配列という原則からすると、Arrayの要素数はどこかで宣言しているはずであるのでそれを使うほうが正しい。

▼ すべての要素の表示

配列はおなじ箱が沢山用意されていると考えればよい。配列をfor-loopで使うときは、箱を指す数(示数, index)をいじっているのか、箱の中身(要素)をいじっているのかを意識すれば、動作を理解しやすい。

```
> for i from 1 to n do
print(i,A[i]);
end do;
1,93
2,45
3,96
4,6
5,98 (4.1)
```

逆順の表示

```
> for i from n by -1 to 1 do
print(i,A[i]);
end do;
5,98
4,6
3,96
2,45
1,93 (4.2)
```

逆順の表示2

```
> for i from 1 to n do
print(n-i+1,A[n-i+1]);
end do;
5,98
4,6
3,96
2,45
1,93 (4.3)
```

▼ 和

```
> sum1:=0;
for i from 1 to n do
sum1:=sum1+A[i];
end do;
sum1;
```

338

(5.1)

課題：積を求めよ。

▼ 値の代入

```
> k:=64:
  for i from 1 to n do
    A[i]:=A[i]/k;
  end do:
  A;
```

$$\begin{bmatrix} 93 & 45 & 3 & 3 & 49 \\ \hline 64 & 64 & 2 & 32 & 32 \end{bmatrix}$$

(6.1)

課題：先の和と組み合わせて、全要素の和が1になるように規格化せよ。

課題：配列Bへ逆順に代入せよ。

```
> B:=Array(1..n,0):
```

▼ 一桁の整数5個から5桁の整数を作る

まず、一桁の整数でできるランダムな配列を作成する。

```
> roll:=rand(0..9):
  n:=5:
  A:=Array([seq(roll(),i=1..n)]);
```

$$A := [3 \ 5 \ 4 \ 0 \ 7]$$

(7.1)

```
> sum1:=0;
  for i from 1 to n do
    sum1:=sum1*10+A[i];
  end do:
  sum1;
```

sum1 := 0

35407

(7.2)

▶ 課題：上記と同様にして、10桁の2進数を10進数へ変換せよ

▼ 255以下の10進数をランダムに生成して、8桁の2進数へ変換せよ。

```
> n:=8:
  2^n;
```

256

(8.1)

```
> roll:=rand(0..255):
  A:=Array(1..n):
  B:=roll();
```

B := 161

(8.2)

```
> for i from 1 to n do
  A[n-i+1]:=irem(B,2);
  B:=iquo(B,2);
end do:
  A;
```

$$[1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1]$$

(8.3)

課題：8桁の整数のそれぞれの桁の値を配列に格納せよ。8桁の整数は以下のようにして作られる。

```
> n:=8;
  roll:=rand(10^(n-1)..10^n):
  B:=roll();
```

n := 8

B := 17914675

(8.4)

▼ 小数点以下8桁のそれぞれの桁の数を配列に格納せよ

```
> n:=8:
  roll:=rand(10^(n-1)..10^n):
  A:=Array(1..n):
  B:=evalf(roll()/10^n);
```

$$A := [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

B := 0.6308447100

(8.1.1)

```
> B:=10*B;
  for i from 1 to n do
    A[i]:=floor(B);
    B:=(B-A[i])*10;
  end do:
  A;
```

B := 6.308447100

$$[6 \ 3 \ 0 \ 8 \ 4 \ 4 \ 7 \ 1]$$

(8.1.2)

▼ 最大数

```
> roll:=rand(1..100):
```

```

n:=5:
A:=Array([seq(roll(),i=1..n)]);
i_max:=A[1]:
for i from 2 to n do
  if (A[i]>i_max) then
    i_max:=A[i];
  end if;
end do;
i_max;

```

```

A := [ 10 15 64 9 12 ]
      64

```

(9.1)

課題：最小値を求めよ。

ある値の上下で分けた個数

```

> roll:=rand(1..100):
n:=5:
A:=Array([seq(roll(),i=1..n)]);
i_div:=50:
i_low:=0:i_high:=0:
for i from 1 to n do
  if (A[i]>i_div) then
    i_high:=i_high+1;
  else
    i_low:=i_low+1;
  end if;
end do;
print(i_low,i_high);

```

```

A := [ 52 25 72 90 18 ]
      2,3

```

(10.1)

素数かどうかの判定

```

> n:=10;
for i from 1 to n do
  if (isprime(i)) then
    print(i);
  end if;
end do;

```

2つの要素の入れ替え

```

> roll:=rand(1..100):
n:=5:
A:=Array([seq(roll(),i=1..n)]);
sel:=rand(1..n):
isel:=sel();
jsel:=sel();
a:=A[isel];
b:=A[jsel];
A[isel]:=b;
A[jsel]:=a;
A;

```

```

A := [ 59 66 12 49 90 ]
      isel:=3
      jsel:=2
      a:=12
      b:=66
      A3:=66
      A2:=12
      [ 59 12 66 49 90 ]

```

(12.1)

より短くするには、

```

> roll:=rand(1..100):
n:=5:
A:=Array([seq(roll(),i=1..n)]);
sel:=rand(1..n):
isel:=sel();
jsel:=sel();
a:=A[isel];
A[isel]:=A[jsel];
A[jsel]:=a;
A;

```

```

A := [ 32 98 9 53 3 ]
      isel:=2
      jsel:=5
      a:=98

```

$$\begin{aligned} A_2 &:= 3 \\ A_5 &:= 98 \\ &[32 \ 3 \ 9 \ 53 \ 98] \end{aligned} \quad (12.2)$$

▼ コインの表向きの枚数

```
> roll:=rand(0..1):
n:=10:
up:=0:
for i from 1 to n do
  trial:=roll():
  if (trial=1) then
    up:=up+1:
  end if:
end do:
up;
```

5 (13.1)

上のやり方は、あまりに冗長。もっと短くすると以下でも等価な結果になる。

```
> roll:=rand(0..1):
n:=10:
up:=0:
for i from 1 to n do
  up:=up+roll():
end do:
up;
```

4 (13.2)

▶ 課題：1..6のサイコロを20回振って、出た目を記録せよ。記録には、要素が0の配列を最初に用意し、出た目を示数にして配列の要素をひとつずつ増やす。

▼ 2次元配列

2次元配列に対しても同様の操作ができる。ここでは列に対する規格化を示す。

```
> roll:=rand(0..1):
n:=3:
A:=Array(1..n,1..n,[seq([seq(roll(),i=1..n)]j=1..n)]);
```

(14.1)

$$A := \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \quad (14.1)$$

```
> for i from 1 to n do
  for j from 1 to n do
    print(i,j,A[i,j]);
  end do;
end do;
```

1, 1, 0
1, 2, 0
1, 3, 1
2, 1, 0

(14.2)

i,jの順序に注意。

```
> for j from 1 to n do
  tmp:=0;
  for i from 1 to n do
    tmp:=tmp+A[i,j];
  end do;
  for i from 1 to n do
    A[i,j]:=A[i,j]/tmp;
  end do;
end do;
A;
```

$$\begin{bmatrix} 0 & 0 & \frac{1}{3} \\ 0 & 1 & \frac{1}{3} \\ 1 & 0 & \frac{1}{3} \end{bmatrix}$$

(14.3)