

Fortranプログラミング入門

-配列(1)-



一次元配列の宣言

- ☆文法
- 1) データ型 :: 変数(寸法)
 - 2) データ型, dimension(寸法) :: 変数

- ・配列とはデータの集合。
同じ変数名と添え字で要素を区別する。

real(8) :: a(4)



配列 a

添え字	要素
1	?
2	?
3	?
4	?

指定しない場合
1番から

すべて
real(8)型

数学の
ベクトル
と同じ!!



一次元配列の宣言

- ☆文法 1) データ型 :: 変数(寸法)
2) データ型, dimension(寸法) :: 変数

- ・配列とはデータの集合。
同じ変数名と添え字で要素を区別する。



数学の
ベクトル
と同じ!!



部分配列

- ☆文法
- 1) 変数(下限:上限:増分)
 - 2) 変数(下限:上限)
 - 3) 変数(:)

- 配列の添え字を範囲で指定できる!!
- 初期化, 演算, 入出力に良く使う!!
- 増分を指定しない場合, 増分は1
- 3)コロンのみの場合, すべての要素で
という意味!

配列の
使いやすさ
がFortranの
真骨頂!



一次元配列の初期化

- ☆文法
- 1) 変数(数字) = 代入したい値
 - 2) 変数(数字1:数字2) = 代入したい値
 - 3) 変数(:) = 代入したい値

`real(8) :: a(4)`
`a(1) = 4d0`



配列 a

添え字	要素
1	4d0
2	?
3	?
4	?

すべて
real(8)型

初期化は
同時にできる!



一次元配列の初期化

- ☆文法
- 1) 変数(数字) = 代入したい値
 - 2) 変数(数字1:数字2:増分) = 代入したい値
 - 3) 変数(:) = 代入したい値

`real(8) :: a(4)`
`a(1:4:2) = 4d0`



配列 a

添え字	要素
1	4d0
2	?
3	4d0
4	?

すべて
real(8)型

初期化は
同時にできる!



一次元配列の初期化

- ☆文法
- 1) 変数(数字) = 代入したい値
 - 2) 変数(数字1:数字2) = 代入したい値
 - 3) 変数(:) = 代入したい値

`real(8) :: a(4)`
`a(:) = 4d0`



配列 a

添え字	要素
1	4d0
2	4d0
3	4d0
4	4d0

初期化は
同時にできる!

すべて
real(8)型



一次元配列の初期化

☆文法 変数(数字1:数字2) = (/代入したい数字, .../)

`real(8) :: a(4)`

`a(:) = (/1d0,2d0,3d0,4d0/)`



配列 a

添え字	要素
1	1d0
2	2d0
3	3d0
4	4d0

初期化は
同時にできる!



一次元配列の初期化

☆文法 $\text{変数}(\text{数字1}:\text{数字2}) = (/(\text{式}, \underline{\text{変数}} = \text{始値}, \text{終値})/)$

- 変数は整数型のみ利用可能!!
- 部分配列の要素の数と代入する要素の数に注意!!

```
real(8) :: a(4)
```

```
integer :: i
```

```
a(:) = (/(2d0*i, i=1,4)/)
```



配列 a

添え字	要素
1	2d0
2	4d0
3	6d0
4	8d0

初期化は
同時にできる!



例題1

要素数100の単精度実数型の変数 a を宣言しなさい。
さらに、 a の奇数番号の要素にはその奇数番号を代入、
それ以外には0を代入せよ。

```
program artest
  implicit none
  real :: a(100)
  integer :: i
  a(:) = 0e0
  a(1:100:2) = (/ (i,i=1,100,2) /)
  write(*,*) a
  stop
end program artest
```

配列の出力

例) 出力の書き方

- 1) `write(*,*) a`
- 2) `write(*,*) (a(i), i=1,3)`
- 3) `write(*,*) a(1:3)`
- 4) `write(*,100) a`
`100 format(3f10.5)`
- 5) `write(*,200) a`
`200 format(f10.5)`
- 6) `write(*,100) a(1:3)`
`100 format(3f10.5)`
- 7) `write(*,200) a(1:3)`
`200 format(f10.5)`

配列の演算

配列同士の演算:

Fortranには配列同士の演算が定義されている!

成分ごとの演算に対応する!!

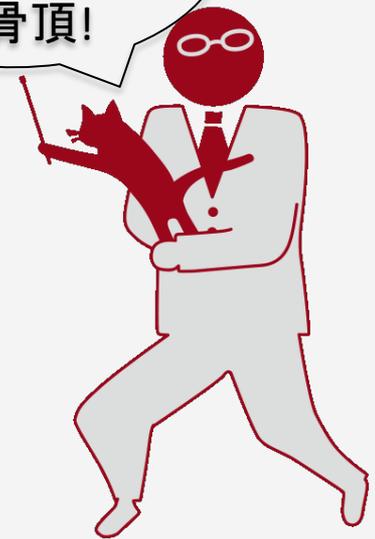
$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \dots \\ a_n \end{pmatrix} \circ \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{pmatrix} = \begin{pmatrix} a_1 \circ b_1 \\ a_2 \circ b_2 \\ a_3 \circ b_3 \\ \dots \\ a_n \circ b_n \end{pmatrix}$$

但し \circ は $+$, $-$, $*$, $/$, $**$ のいずれかである.

※ 要素数が同じ場合にのみ計算可能!!

※ 記号 $*$ や $/$, $**$ は数学とは意味が異なるので注意!!

配列の
使いやすさ
がFortranの
真骨頂!



配列の演算

スカラと配列の演算:

Fortranにはスカラと配列の演算が定義されている!
成分ごとの演算に対応する!!

$$a \circ \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{pmatrix} = \begin{pmatrix} a \circ b_1 \\ a \circ b_2 \\ a \circ b_3 \\ \dots \\ a \circ b_n \end{pmatrix} \quad \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \dots \\ a_n \end{pmatrix} \circ b = \begin{pmatrix} a_1 \circ b \\ a_2 \circ b \\ a_3 \circ b \\ \dots \\ a_n \circ b \end{pmatrix}$$

但し \circ は $+$, $-$, $*$, $/$, $**$ のいずれかである.

配列の
使いやすさ
がFortranの
真骨頂!



配列の数学関数

配列の数学関数:

Fortranには配列の数学関数が定義されている!

成分ごとの演算に対応する!!

例えば...

$$\sin \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \dots \\ a_n \end{pmatrix} = \begin{pmatrix} \sin(a_1) \\ \sin(a_2) \\ \sin(a_3) \\ \dots \\ \sin(a_n) \end{pmatrix} \quad \text{sqrt} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \dots \\ a_n \end{pmatrix} = \begin{pmatrix} \text{sqrt}(a_1) \\ \text{sqrt}(a_2) \\ \text{sqrt}(a_3) \\ \dots \\ \text{sqrt}(a_n) \end{pmatrix}$$

※maxなど使えない関数もあるので注意!!

配列の
使いやすさ
がFortranの
真骨頂!



例題2

要素数10の倍精度実数型の配列a,b,cを宣言せよ. aとbをそれぞれ

$$a(:) = (/ (2*i, i=1, 10) /)$$

$$b(:) = (/ (-i, i=11, 20) /)$$

とする. そのとき, 次の演算を行え:

$$c = a*b$$

$$c = 2+a**b$$

$$c = \sin(a)$$

但し, 出力は100 format(10f15.5)を用いよ.

```
program examp2
  implicit none
  integer :: i
  real(8), dimension(10) :: a, b, c
  a(:) = (/ (2*i, i=1, 10) /)
  b(:) = (/ (-i, i=11, 20) /)
  write(*, 100) a
  write(*, 100) b
  c = a*b
  write(*, 100) c
  c = 2+a**b
  write(*, 100) c
  c = sin(a)
  write(*, 100) c
  stop
  100 format(10f15.5)
end program examp2
```

関数名	意味	入力引数の型
sum(x)	ベクトルの総和	xは一次元配列 単/倍精度実数 単精度複素数
maxval(x)	ベクトルの最大値	xは一次元配列 整数 単/倍精度実数
minval(x)	ベクトルの最小値	xは一次元配列 整数 単/倍精度実数
dot_product(x,y)	内積	x,yは同じ要素数の 一次元配列 整数 単/倍精度実数 単精度複素数