

Fortranプログラミング入門

-変数・定数-

2.7.18.2.8

変数とは...

メモリーのどこかに名前をつけて、様々なデータを格納及び利用できるようにしたもの.

変数名の付け方の注意

- 英字, 数字, アンダースコア_のみ
- 変数名の先頭は英字のみ
- Fortranで用意されている予約語(write等)は使えない
- プログラム名と同じ変数名は使えない

変数のデータ型とは...

- ・変数に格納するデータの種類の種類.
- ・変数を利用する際は, 宣言時にデータ型を指定する

データ型	説明	宣言文
整数型	32bitの整数	INTEGER :: 変数リスト
整数型(長い桁)	64bitの整数	INTEGER(8) :: 変数リスト
単精度実数型	32bit 有効数字約7桁の実数	REAL :: 変数リスト
倍精度実数型	64bit 有効数字約15桁の実数	REAL(8) :: 変数リスト
複素数型	実部, 虚部が単精度の複素数	COMPLEX :: 変数リスト
文字型	文字データ	CHARACTER :: 変数リスト
論理型	論理演算 真(.true.) 偽(.false.)	LOGICAL :: 変数リスト

暗黙の型宣言とは...

FORTRAN77では宣言文を書かなければ

- i, j, k, l, m, nから始まる変数は整数型
- それ以外は単精度浮動小数点数型

になる. 必要なときに限り宣言文を書いていた.

implicit noneとは

Fortran90以降は暗黙の型宣言を推奨しておらず,
「**暗黙の型宣言をなくす**」という文.

必ず書くようにしましょう!!

例えば...

```
program main
```

```
  implicit none
```

```
  integer :: i, j
```

```
  real(8) :: x, y
```

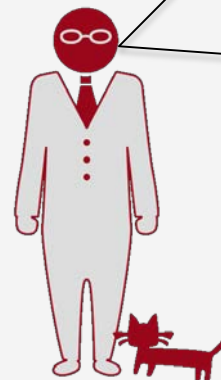
```
  実行文
```

```
  stop
```

```
end program main
```

← 暗黙な型宣言を無効化し、
整数型と倍精度実数型の変数を宣言

program プログラム名
 宣言文
 実行文
 stop
end program プログラム名



定数の書き方

数字を書く際, 書き方によってデータ型が変わる

整数型 (integer型)

1402

+39

-25

0

間違った例

1.2

1,000

定数の書き方

数字を書く際, 書き方によってデータ型が変わる

単精度実数型 (real型)

1.23

1.0

2.45e-3

2.45×10^{-3}

2.4e3

+5.43e2

-3.14e2

倍精度実数型(real(8)型)

2.45d-3

2.4d3

+5.43d2

-3.14d2

-3.14×10^2

定数の書き方

数字を書く際, 書き方によってデータ型が変わる

複素数型 (complex型)

(実部, 虚部)

(3.14, 2.5)

(-2.0, 4.2e3)

(1.58e-2, 40.3)

(1.56e-4, 6.88e4)

代入演算子 =

変数に値を代入する

変数 = 代入したい値

例

```
implicit none
```

```
integer :: i, j
```

```
i = 5
```

```
j = i
```

```
write(*,*) i, j
```

整数型の変数 i を宣言し,

5, 5.1, 5.0d-2, (1.0, 2.0)

をそれぞれ i に代入し, それぞれ出力するプログラムを作成せよ.

但し, ファイル名を subs.f90 とせよ.



```
program subs
  implicit none
  integer :: i
  i = 5
  write (*,*) i
  i = 5.1
  write (*,*) i
  ...
  stop
end program subs
```

