

# Fortran 入門

# 目次

- 変数
- 配列
- 演算子と組み込み関数

**変数**

# 変数

- Fortran に限らず, コンピュータ言語では, 「変数」を用いて様々な処理を行います.

– 例えば

- 数値の計算,

```
num1 = 2  
num2 = 3  
num3 = num1 * num2
```

- » num1, num2, num3 は変数
- » num1 と num2 の積を num3 に格納

- 文字の扱い

```
line = "Hello world"
```

- » line は変数

# 変数を扱うときの注意

- 代入する値に適した型の変数を使う
- 変数を使う前に宣言する
- 変数名には制限がある
  - 最大 31 文字
  - 使える文字は英数字とアンダースコア ( \_ )
- 変数は必ずしも正確な実数値を持ってない。

# Fortran で用意されている 変数の種類 (データ型)

種類	データ型	値の範囲
基本整数型	integer	-2,147,483,648 ~ 2,147,483,647
8 バイト整数型	integer(8)	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807
単精度実数型	real	$1.17549435 \times 10^{-38} \sim 3.40282347 \times 10^{38}$ 精度は10進数で約6桁(2進数で24桁)
倍精度(8 バイト) 実数型	real(8) または double precision	$2.225073858507201 \times 10^{-308}$ ~ $1.797693134862316 \times 10^{308}$ 精度は10進数で約14桁(2進数で53桁)
複素数型	complex	a + b i の形式の複素数(a, b は実数の範囲)
倍精度複素数型	complex(8) または double complex	a + b i の形式の複素数(a, b は倍精度実数の範囲)
論理型	logical	.true. (真) または .false. (偽)
文字型	character	1 バイト文字

# 変数の型

- 変数は、代入する値の種類に適したものを使わなければならない。
  - 整数を扱うときには整数型変数を使う
  - 実数を扱うときには実数型変数を使う
  - 文字を扱うときには文字型変数を使う
- 間違えると...
  - 文字型変数に数値を代入するとコンパイルできない。
  - 整数型変数に実数を代入すると小数点以下が無視される。
    - 間違った結果が計算される。

# 変数の宣言

- プログラムの中で変数を使うためには, 変数を使う前に「宣言」.
- 例
  - integer :: number    整数型変数 number の宣言
  - real :: value        実数型変数 value の宣言
- 宣言方法

<データ型> :: <変数名>



コロンが二つ



# 変数名

- 変数名
  - 最大 31 文字
  - 使える文字は英数字とアンダースコア ( \_ )
- 実際には、31 文字を超える変数名を使ってもコンパイルできることがある。
  - Fortran の規則では最大 31 文字.
  - 実際には、コンパイラによってコンパイルできることも多い。
    - 各コンパイラの独自拡張
    - gfortran では、31 文字を超える文字数の変数を使ってもコンパイルできる。
      - オプションを使って文法を正確にチェックするとコンパイルできない。
        - » 様々な環境で使えるプログラムを作るためには、規則に従ったプログラムを作ることが有益.

# 計算機における数値の表現

- 計算機は有限な 2 進数で数値を表現.
  - 整数は正確に表現できる.
    - 表現できる最大値を超えない限り
  - 実数は必ずしも正確には表現できない.
    - 実数の表現方法の制限から誤差がある.
    - 小数点以下の桁が無限にあるような数値は表現できない.
- 以後, 簡単のため 1 バイト変数で説明.

# 計算機における数値の表現 (符号なし) 1 バイト整数

2 進数

0 1 0 0 0 1 0 1

× × × × × × × ×

桁

$2^7$   $2^6$   $2^5$   $2^4$   $2^3$   $2^2$   $2^1$   $2^0$

|| || || || || || || ||

0 64 0 0 0 4 0 1



10 進数

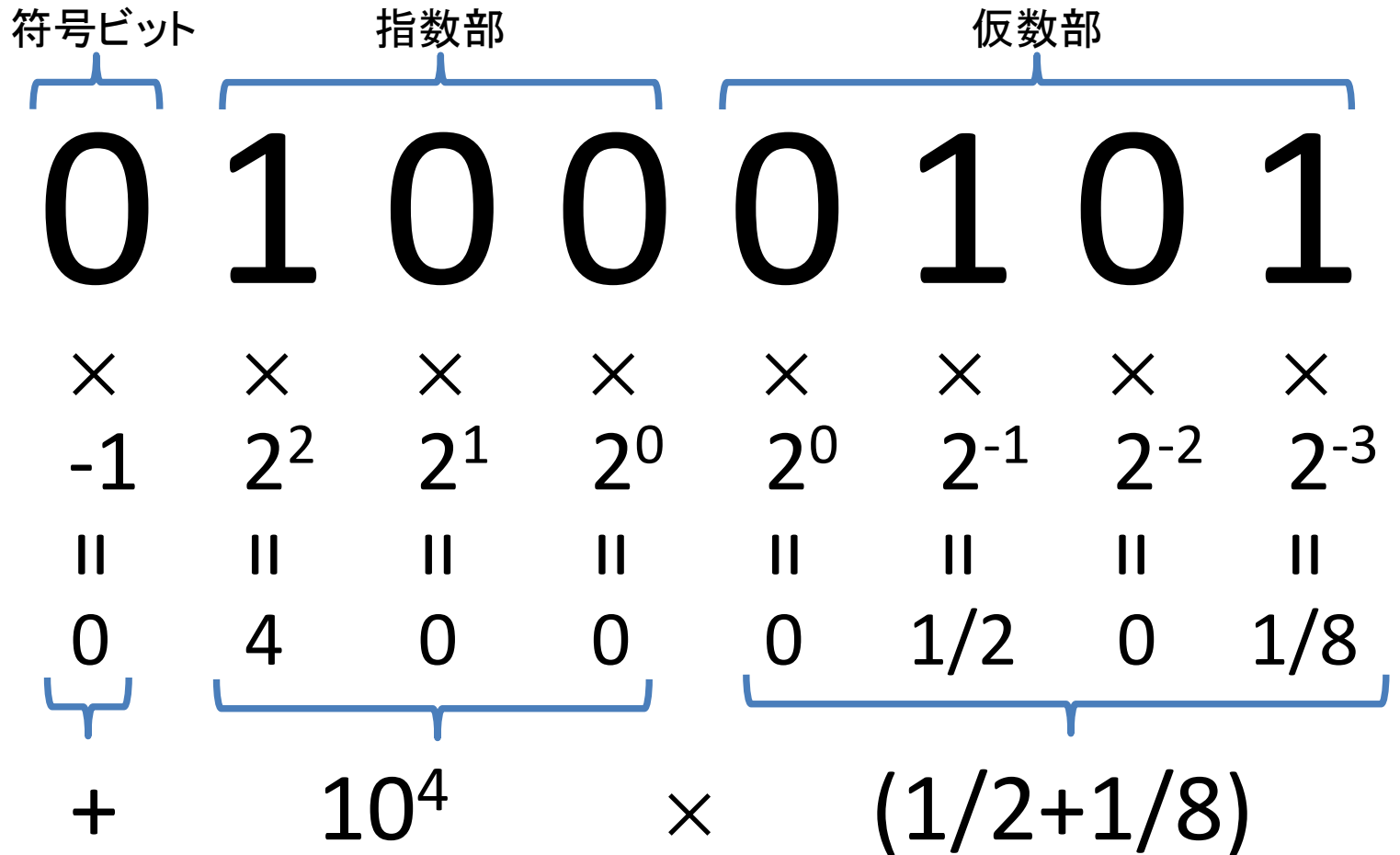
$$64+4+1=69$$

# 計算機における数値の表現 (符号なし)1バイト整数

ビットの状態	10進数
00000000	0
00000001	1
00000010	2
00000011	3
00000100	4
00000101	5
00000110	6
...	...
01000101	69
...	...

- 0, 1, 2, ... のように(255までの)すべての整数を表現

# 計算機における数値の表現 (符号付き) 1バイト実数



$$= 0.625 \times 10^4$$

# 計算機における数値の表現 (符号付き)1バイト実数

ビットの状態	10進数
0 000 0000	$0.000 \times 10^0$
0 000 0001	$0.125 \times 10^0$
0 000 0010	$0.250 \times 10^0$
0 000 0011	$0.375 \times 10^0$
0 000 0100	$0.500 \times 10^0$
0 000 0101	$0.625 \times 10^0$
0 000 0110	$0.750 \times 10^0$
...	...
0 100 0101	$0.625 \times 10^4$
...	...

- 0.000, 0.125, 0.250, ... のように、飛び飛びに値を表現
  - 例えば, 0.1251 を正確に表現できない(誤差がある).
    - 仮数部により多くのビットに持っていれば, より真値に近づく.
- 0.1251 のような, 正確に表現できない値は, 近い値 (0.125) となる.
  - 0.1251 と 0.1250 を区別できない.
- また, 小数点以下の桁が無限にある数値 (例えば,  $1/3 = 0.33\dots$ ) も正確には表現できない.

# Fortran で用意されている 変数の種類 (データ型)

種類	データ型	値の範囲
基本整数型	integer	-2,147,483,648 ~ 2,147,483,647
8 バイト整数型	integer(8)	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807
単精度実数型	real	$1.17549435 \times 10^{-38} \sim 3.40282347 \times 10^{38}$ 精度は10進数で約6桁(2進数で24桁)
倍精度(8 バイト) 実数型	real(8) または double precision	$2.225073858507201 \times 10^{-308}$ ~ $1.797693134862316 \times 10^{308}$ 精度は10進数で約14桁(2進数で53桁)
複素数型	complex	a + b i の形式の複素数(a, b は実数の範囲)
倍精度複素数型	complex(8) または double complex	a + b i の形式の複素数(a, b は倍精度実数の範囲)
論理型	logical	.true. (真) または .false. (偽)
文字型	character	1 バイト文字

**配列**



# 配列 その1

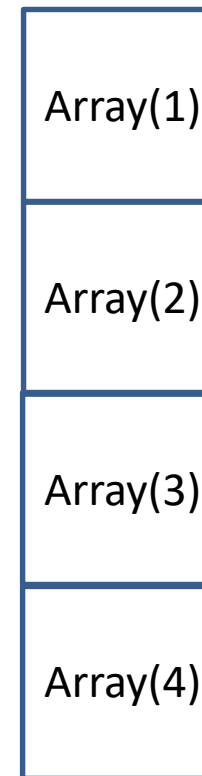
- 複数のデータをまとめて扱いたいことがある。
  - データの例
    - 温度の高さ分布
      - 複数の高度の値や複数の温度の値を用いる。
    - 地震の揺れの時間変化
      - 複数の時間の値や複数の揺れの加速度の値を用いる。
    - ...
- こういった複数の数値をまとめて扱う際に便利なものが「配列」

# 配列 その2

- 配列

- 同じデータ型の値をまとめて一つの変数名で扱う.
  - 時系列データを表現
  - ベクトルを表現
  - ...
- それぞれの値(要素)が値を持てる.
- それぞれの要素は, 添え字で指定.
  - Fortran のデフォルトでは, 添え字は, 1, 2, 3, ...
    - 添え字が 0 (ゼロ) から始まる言語も少なくない.

## 配列 Array(4) のイメージ



# 配列 その3

- 多次元配列
  - 2次元以上の配列も定義できる.
    - 行列を表現
    - ...

配列 Array(4,3)のイメージ

Array(1,1)	Array(1,2)	Array(1,3)
Array(2,1)	Array(2,2)	Array(2,3)
Array(3,1)	Array(3,2)	Array(3,3)
Array(4,1)	Array(4,2)	Array(4,3)

# 配列の宣言

- 例

- integer :: number(5)

- 5 個の要素を持つ整数型変数配列の宣言

- real :: value(11,3)

- 11 × 3 個の要素を持つ実数型変数配列の宣言

- real(8) :: value(2,3,4,5)

- 2 × 3 × 4 × 5 個の要素を持つ倍精度実数型変数配列の宣言

- 宣言方法

<データ型> :: <変数名> (<要素数>)

# 配列の使用例

```
integer :: number(4)
real    :: value(5)
```

...

配列の宣言

配列の要素数

number(1) = 2

number(2) = 1

number(3) = 5

number(4) = number(1) \* number(2) + number(3)

配列の要素番号の指定

# 演算子と組み込み関数

# Fortran で用意されている 演算子, 組み込み関数

- Fortran では, 様々な演算のための演算子や関数が予め用意されている.
  - 予め用意されている関数を「組み込み関数」と呼ぶ.
  - ユーザが関数を定義することもできる.
    - これに関しては後日説明予定.

# Fortran で用意されている 演算子

演算子	意味	用例
+	足す	$x + y$
-	引く	$x - y$
*	掛ける	$x * y$ (x 掛ける y)
/	割る	$x / y$ (x 割る y)
**	べき乗	$x ** y$ (x の y 乗)
=	代入	$x = y$ (x に y の値を代入)



# コンピュータ言語における“=”

- コンピュータ言語での“=”は「代入」を意味し、数学とは意味が異なります。

$$x = 5$$

変数  $x$  に 5 を代入

$$x = x + 1$$

変数  $x$  に  $x+1$  の値 (6) を代入

– 上の二つ目の数式は、数学では成り立ちませんが、コンピュータ言語としては意味が通ります。

# Fortran で用意されている 組み込み関数 その1

Fortran 関数名	意味	備考
$\sin(x)$	正弦 (sine)	x の単位は radian
$\cos(x)$	余弦 (cosine)	x の単位は radian
$\tan(x)$	正接 (tangent)	x の単位は radian
$\text{asin}(x)$	逆正弦 (arcsine)	出力の単位は radian
$\text{acos}(x)$	逆余弦 (arccosine)	出力の単位は radian
$\text{atan}(x)$	逆正接 (arctangent)	出力の単位は radian
$\text{sqrt}(x)$	平方根	
$\text{exp}(x)$	指数関数	
$\log(x)$	自然対数	
$\log_{10}(x)$	10 を底とする対数	
$\text{mod}(x,y)$	x を y で割った余り	
$\text{abs}(x)$	x の絶対値	

# Fortran で用意されている 組み込み関数 その2

Fortran 関数名	意味	備考
max(x1, x2, ...)	x1, x2, ... の最大値	
min(x1, x2, ...)	x1, x2, ... の最小値	
int(x)	整数への変換(切り捨て)	
nint(x)	整数への変換(四捨五入)	
real(x)	実数への変換	
dbble(x)	倍精度実数への変換	
cmplx(x)	複素数への変換	
len(a)	文字列長	a は文字型変数
len_trim(a)	後ろの空白を除いた文字長	a は文字型変数
trim(a)	後ろの空白を除く	a は文字型変数

# 実習

- 実習を通して, Fortran での変数と配列の使い方や Fortran での演算に慣れましょう.