

# いまさら Fortran 入門

# 目次

- 入出力
- 条件分岐
- 繰り返し

**入出力**

# 入出力

- データ解析, 数値計算等の処理では, データの入力, および結果の出力のために, 画面, キーボード, ファイルからの/への入出力が必須である.
- これまでは, 画面への出力として簡単な print 文を用いてきたが, ここでは下のことを学ぶ.
  - ファイルを開く/閉じる
  - 画面やファイルへの出力
  - キーボードやファイルからの入力

# Fortran の入出力文

- read 文
  - キーボードやファイルからの入力に使用
  - 例
    - `read( 5, * ) line`
- print 文
  - 画面への出力に使用
  - 例
    - `print *, "Hello world"`
- write 文
  - 画面やファイルへの出力に使用
  - 例
    - `write( 6, * ) "Hello world"`

# read 文

- キーボードやファイルからの入力に使用
- 使用例

```
read( 5, * ) line
```

– キーボードから適当な書式で値を読み込んで line に代入

- 使用法

```
read( <装置番号>, <書式>, ... ) <入力される変数>
```

– <装置番号>

- 入力先を指定する番号(自然数)

- 例

- 5 : キーボードからの入力(決まっている番号)

- その他 : ファイルからの入力に使用(5, 6 以外を推奨)

– <書式>

- 入力される値の書式の指定

- 例

- \* : 「適当」にやってくれる

**ファイル名ではなく、装置番号で出力先を指定**

# print 文

- 画面への出力に使用
- 使用例

```
print *, "Hello world"
```

- 画面に適切な書式で “Hello world” を出力

- 使用法

```
print <書式>, <出力する値/変数>
```

- <書式>

- 出力される値の書式の指定
- 例

- \* : 「適当」にやってくれる

- 備考

- 画面の出力にしか使えない.
- write は画面とファイルの両方に出力できる.

# write 文

- 画面やファイルへの出力に使用
- 使用例

```
write( 6, * ) "Hello world"
```

– 画面に適切な書式で “Hello world” を出力

- 使用法

```
write( <装置番号>, <書式>, ... ) <出力する値/変数>
```

– <装置番号>

- 入力先を指定する番号(自然数)

- 例

- 6 : 画面への入力(決まっている番号)
- その他 : ファイルへの出力に使用(5, 6 以外を推奨)

– <書式>

- 出力する値の書式の指定

- 例

- \* : 「適当」にやってくれる

**ファイル名ではなく、装置番号で出力先を指定**



# ファイルからの入力/への出力

- ファイルからの入力/への出力には下の手順が必要
  - ファイルを開く (open 文)
  - ファイルからの入力/への出力 (read/write 文)
  - ファイルを閉じる (close 文)

# open 文

- 使用例

```
open( 11, file="data.txt", status="unknown")
```

- data.txt の名前のファイルを装置番号 11 番で開く

- 使用法

```
open( <装置番号>, file=<ファイル名>, status=<ステータス> )
```

- <装置番号>

- ファイルを指定する番号

- <ステータス>

- 例

- “unknown” : 「適当」にやってくれる
- “old” : 読み込み用に開く
- “replace” : 書き込み用に開く

read/write で使う装置番号とファイル名を関係づける

# close 文

- 使用例

```
close( 11 )
```

- 装置番号 11 番のファイルを閉じる

- ファイルを開く

```
close( <装置番号> )
```

- <装置番号>

- ファイルを指定する番号
    - open で指定した番号を使用する

# 条件分岐

# 条件分岐

- データ解析, 数値計算等の処理では, 様々な処理の中で場合分けして処理することがよくある.
- ここでは, Fortran での条件分岐の方法を簡単に解説する.

# Fortran での代表的な条件分岐

- 使用例

```
if ( i > 0 ) then  
    write( 6, * ) i  
end if
```

- i が 0 よりも大きいときに i の値を画面に表示

- 使用法

```
if ( [条件文] ) then  
    [実行文]  
end if
```

- [条件文]
  - 次ページで説明

# 比較演算子・論理演算子と使用例

演算子	旧来の書き方	意味	使用例
<code>==</code>	<code>.eq.</code>	等しい	<code>a == b</code>
<code>/=</code>	<code>.ne.</code>	等しくない	<code>a != b</code>
<code>&gt;</code>	<code>.gt.</code>	より大きい	<code>a &gt; b</code>
<code>&gt;=</code>	<code>.ge.</code>	以上	<code>a &gt;= b</code>
<code>&lt;</code>	<code>.lt.</code>	より小さい	<code>a &lt; b</code>
<code>&lt;=</code>	<code>.le.</code>	以下	<code>a &lt;= b</code>
<code>.not.</code>		以外(否定)	<code>.not. (a==b)</code>
<code>.and.</code>		かつ	<code>(a==b) .and. (c==d)</code>
<code>.or.</code>		もしくは	<code>(a==b) .or. (c==d)</code>

**繰り返し**



# 繰り返し

- データ解析, 数値計算等の処理では, 多数のデータ・数値に同じ処理を行うことがよくある.
- 計算機は, 同じことを多数回実施することが得意である.
- ここでは, Fortran での繰り返しの方法を簡単に解説する.

# Fortran での代表的な繰り返し

- 使用例

```
do i = 1, 10  
  write( 6, * ) i  
end do
```

- $i = 1, 2, \dots, 10$  まで  $i$  の値を画面に表示

- 使用法

```
do <変数> = <開始>, <終了>, <間隔>  
  [実行文]  
end do
```

- <間隔>

- 省略可能

- 例えば 2 とすると、上の例の場合では  $i = 1, 3, 5, 7, 9$  の場合に実行

# 実習

- 実習を通して, 入出力, 繰り返しや条件分岐に慣れましょう.