

基礎プログラミングI

第3回 メソッド、値の型変換

メソッドとは

あらかじめ決められた(複雑な)処理を行ってくれるもの。

- Rubyにあらかじめ備わっているもの (printfなど)もある。
- 自作することもできる。

メソッドの呼び出し方

何に対して.どうする(必要な情報)

オブジェクト.メソッド名(引数)

基本的メソッド

場所.gets

指定された場所から文字列を読み込んでその値を返す。

- 「場所.」を省略するとプログラム起動時に与えたファイルから読み込む。
- 起動時にファイルを与えない場合は標準入力から読み込む。
- 文字列の末尾は必ず改行文字 (`\n`)が入る。

getsメソッド使用例

```
foo = STDIN.gets
```

標準入力 (次項で説明) から読み込んだ文字列をfoo変数に代入

文字列.chomp

文字列の末尾が改行文字 (`\n`) である場合にそれを切り取る。改行文字でなければ何もしない。

chompメソッド使用例

```
line = gets.chomp
```

標準入力から1行読み込んで得た文字列の末尾から改行文字を除去した文字列を作成し、line変数に代入

文字列.to_i

文字列を整数に変換した値を返す。

to_iメソッド使用例

```
hoge = "123"
```

```
x = hoge.to_i
```

文字列“123”を整数123に変換した結果を返し、xに代入

値の型変換

文字列から数値へ

to_i

数字が並んでいる文字列を10進数と見なしして整数に変換する。

“123”.to_i → 123

“3.1415”.to_i → 3

to_f

数字が並んでいる文字列を10進数と見なして浮動小数点数に変換する。

“3.1415”.to_f → 3.1415

数値から文字列へ

printf

最初の引数の文字列の中にある%Xを、2個目以降の引数で与えた値に置き換えて出力する。

例：

```
printf("%3d年\t%d人\n", y+=1, p)
```

最初の%3dがy+=1の値に置き換えられ、次の%dがpの値に置き換えられる。

sprintf

printfで出力されるものを文字列として返す。

例：

```
month = 3
```

```
day = 1
```

```
printf("%02d%02d", month, day)
```

```
# (文字列"0301"が出力される)
```

```
s = sprintf("%02d%02d", month, day)
```

```
# (文字列"0301"が変数sに代入される)
```

to_s

全ての型の値を文字列に変換する。

まとめ

メソッド：

あらかじめ決められた処理を行ってくれるもの

値の型変換：

あるデータ型を別のデータ型に変更すること