

情報科学
数の計算と関数

今週の目標:コンピュータに計算を させる方法を知る

- 数式を計算させる
- 関数を使った計算
- 変数を使った計算
- 関数を定義する
- プログラムを保存する
- 間違いを見つける

もう1つ
「ターミナル」の
ウィンドウを
開く

コンピュータとの対話

- ターミナルの起動 ⇒ irbの起動 ⇒ 数式の入力

ターミナルの
プロンプト

\$ **irb** 改行

irbの
プロンプト

irb (main) : 001 : 0 > **1+1** 改行

=> 2

irbの
返答

irb (main) : 002 : 0 > **コントロールD**

\$

入力は
赤で示す

- 今回は、指示されるまで、irbを使いながら...

数式の計算 — 電卓がわり

以下、改行は省略

irb(main):003:0> 7 - 2

=> 5

irb(main):004:0> 7 * 2

=> 14

irb(main):005:0> 7 / 2

=> 3

irb(main):006:0> 7 % 2

=> 1

irb(main):007:0> 7 ** 2

=> 49

irb(main):009:0> 7 - 2 * 3

=> 1

irb(main):010:0> (7 - 2) * 3

=> 15

irb(main):012:0> 7.0 / 2

=> 3.5

irb(main):013:0> 7 / 2.0

=> 3.5

7の2乗

7を2で割った余り

クイズ1: **17 - 17/3*3** の値は?

1: 0.0

2: 0

3: 2

4: 15.111111111111111

5: 16

irbを使わずに
答えよう!

クイズ2: 清原和博のBMIを求める式 正しいのは?

野球選手清原和博のBody Mass Index (BMI).
身長は188 cm体重は104 kgとせよ。
またBMIとは体重(kg)を身長(m)の自乗で割った値。

- 1 $104 / (188 / 100) ** 2$
- 2 $104 / (188 / 100.0) ** 2$
- 3 $104 / ((188 / 100) ** 2)$
- 4 $104 / 188 / 100.0 ** 2$

irbを
使わず
に
答えよう!

わけがわからなくなったら

- ともかく**コントロールC**を押す
irbはトップレベルに戻る

数学関数

数学関数を使う準備
irbを起動し直す
たびに必要

```
irb(main):003:0> include(Math)
```

```
=> Object
```

```
irb(main):004:0> sqrt(2)
```

```
=> 1.4142135623731
```

```
irb(main):005:0> cos(3.141592/3)
```

```
=> 0.50000018867511
```

黄金比の値は

1. 1.61803398874989
2. 1.61803398874988
3. 1.61803398874987
4. 1.61803398874986
5. 1.61803398874985

$$\frac{1 + \sqrt{5}}{2}$$

変数 — 値に名前を付ける

変数への値の代入

```
irb(main):003:0> h=188.0
```

```
=> 188.0
```

```
irb(main):004:0> w=104.0
```

```
=> 104.0
```

```
irb(main):006:0> w / (h / 100.0) ** 2
```

```
=> 29.4250792213671
```

変数を使うわけ

- 式の意味が理解しやすくなる

`w`

`weight`

`body_weight_in_pound`

- 違う値で計算のやり直しができる

```
irb(main):008:0> w=104.0-10
```

```
=> 94.0
```

```
irb(main):009:0> w / (h/100.0) ** 2
```

```
=> 26.5957446808511
```

- 小文字で始まること
- アンダースコアは
小文字と考える

↑キーで
入力が復活!

関数の定義 — BMIを求める関数

```
irb(main):003:0> def bmi(height, weight)
irb(main):004:1>     weight / (height/100.0) ** 2
irb(main):005:1> end
=> nil

irb(main):007:0> bmi(188.0, 104.0)
=> 29.4250792213671

irb(main):008:0> 1.1*bmi(174.0, 119.0 * 0.454)
=> 19.6289470207425
```

#以下
行末まで
コメント

ファイルに保存した関数定義 —ファイルからの読み込み

```
# BMI of a person with height (cm) and weight (kg)
def bmi(height , weight )
  weight / ( height / 100.0) ** 2
end
```

Emacsを使おう

ホームディレクトリへ
保存せよ!

bmi.rb

一度irbを終了し
再起動してから...

```
irb(main) :003:0> load("./bmi.rb")
```

```
=> true
```

```
irb(main) :005:0> bmi(188.0, 104.0)
```

```
=> 29.4250792213671
```

関数を使う関数

関数も、
小文字で始まる英数字列
アンダースコアは
小文字と考える

```
irb(main):010:0> def bmi_yp (f,i,p,o)
irb(main):011:1>     bmi(feet_to_cm (f,i) ,
irb(main):012:2*     pound_to_kg(p,o))
irb(main):013:1> end
=> nil
```

式の途中で改行すると
プロンプトが異なる

```
irb(main):015:0> bmi_yp(5,11,170,0)
=> 23.7099441655737
```

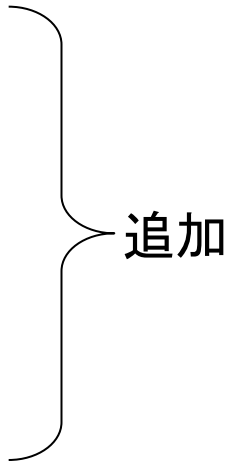
ファイルを読み込むファイル

```
load("./bmi.rb")  
load("./yardpound.rb")  
  
def bmi_yp(f,i,p,o)  
  bmi(feet_to_cm(f,i), pound_to_kg(p,o))  
end
```

bmi_yp.rb

定数関数

```
# BMI of a person with height (cm) and weight (kg)
def bmi(height , weight )
  weight / ( height /100.0) ** 2
end
def k_height ()      #K選手の身長
  188.0
end
def k_weight ()     #K選手の体重
  104.0
end
```



bmi.rb

定数関数

```
irb(main):004:0> load("./bmi.rb")
```

```
=> true
```

```
irb(main):005:0> k_weight()
```

```
=> 104.0
```

```
irb(main):006:0> bmi(k_height(), k_weight())
```

```
=> 29.4250792213671
```

局所変数

局所変数の定義

```
def heron (a,b,c)
  s = 0.5*(a+b+c)
  sqrt(s * (s-a) * (s-b) * (s-c))
end
```

heron.rb

練習

- 二次方程式 $ax^2 + bx + c = 0$ に関して
 - (a) 判別式 $b^2 - 4ac$ を求める $\text{det}(a,b,c)$.
 - (b) 解の1つを求める $\text{solution1}(a,b,c)$. (det を使って定義せよ。)
 - (c) もう1つの解を求める $\text{solution2}(a,b,c)$.
(solution1 と solution2 の共通部分を1つの補助関数にできるか?)
 - (d) 二次関数 $f(x) = ax^2 + bx + c$ の値を求める $\text{quadratic}(a,b,c,x)$.

1.6 定義のまとめ

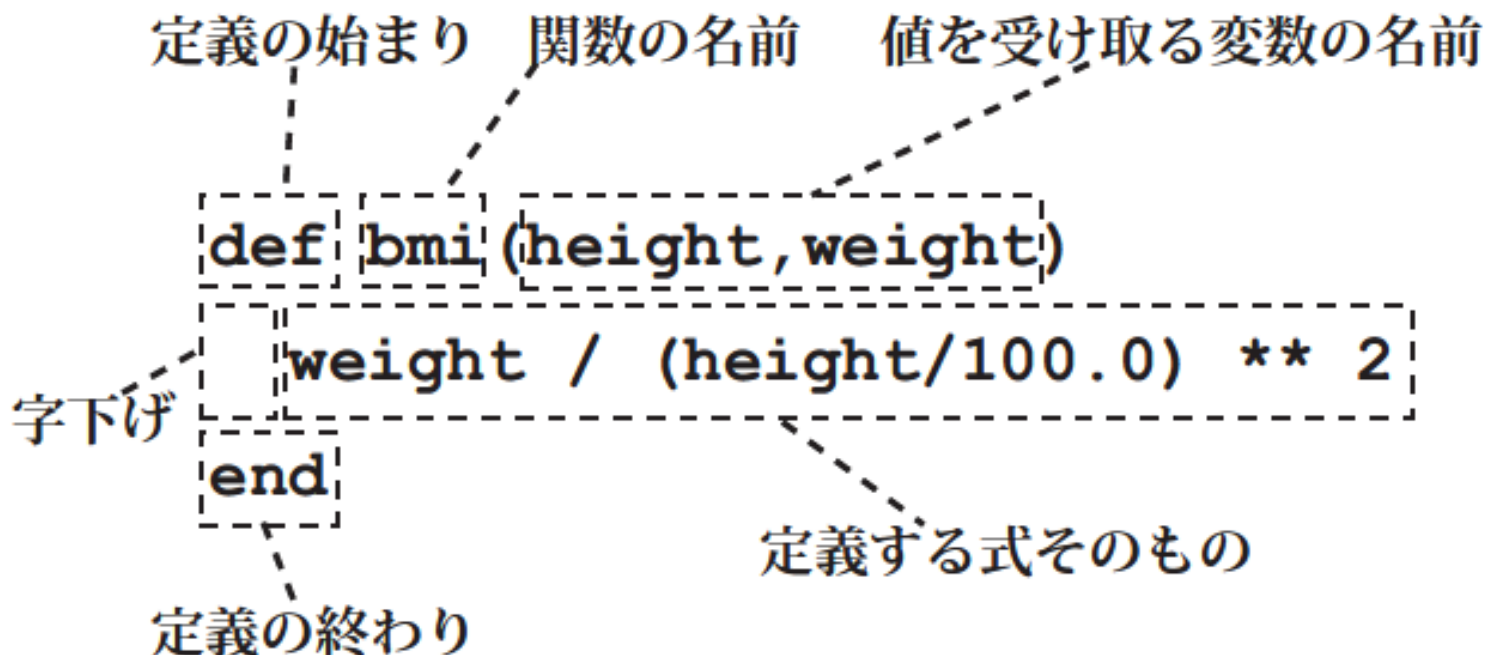
この章で紹介した式や命令をまとめておこう。

`include(Math)` : `cos` や `sqrt` などの数学関数を使う前に、実行しておかなければいけない命令。

`変数名` = `式` : 右辺の式を計算した値を、左辺に書かれた名前の変数にしまう代入命令。

`関数名` (`式1`, ..., `式n`) : `式1`, ..., `式n` を計算した値を、関数に渡す関数呼び出し式。関数に渡される式の値のことを^{ひきすう}引数という。`関数名` は `sqrt` のような予め定義されている関数でも、`bmi` のような自分で定義した関数でもよい。

def **関数名** (**式₁**, ..., **式_n**) **式** end : 関数定義。少し複雑なので、具体例で説明する。



`def` と `end` は、定義の範囲を示している。関数名(ここでは `bmi`) は、変数名と同様、好きな名前を付けることができる。次の `(height, weight)` は、関数が値を受け取るために使う変数名である。

さまざまなエラー

```
irb(main):001:0> 3/0
```

```
ZeroDivisionError: divided by 0
```

```
  from (irb):1:in `/'
```

```
  from (irb):1
```

```
irb(main):002:0> 7 - 2 3
```

```
SyntaxError: compile error
```

```
(irb):2: syntax error, unexpected tINTEGER,  
  expecting $end
```

```
  from (irb):2
```

```
irb(main):003:0> (7 -
```

```
irb(main):004:1* 2) * 3)
```

```
SyntaxError: compile error
```

```
(irb):4: syntax error, unexpected ')', expecting  
  $end
```

```
  from (irb):4
```

```
irb(main):005:0>
```

式の途中で改行すると
プロンプトが異なる

さまざまなエラー

```
irb(main):013:0> bm1(188.0, 104.0)
```

```
NoMethodError: undefined method 'bm1'  
for main: Object  
from (irb):13
```

irbへの入力

- ↑を入力すると、直前の入力が復活
- ←→でカーソルを移動
- コントロールDでカーソル位置の文字が削除
- バックスペースで直前の文字が削除