

条件分岐と繰り返し

教科書第3章(33ページから)

自習状況の確認

1. 教科書の練習1.3, 1.9など, すべて試した
2. 教科書の練習1.3は試した
3. スライドの bmi と heron までは試した
4. 検査プログラム check.rb は実行した
5. 検査プログラムも実行していない

条件分岐 --- 場合分けを使った計算

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

```
=> true
```

```
irb(main):004:0> max(123, 456)
```

```
=> 456
```

```
irb(main):005:0> max(max(12, 34), max(56, 78))
```

```
=> 78
```

```
def max(x,y)
  if y < x
    x
  else
    y
  end
end
max.rb
```

3 通りの場合分け

```
def sign(x)
  if x < 0
    -1
  else
    if 0 < x
      1      # not(x<0) and 0<x
    else
      0      # not(x<0) and not(0<x)
    end
  end
end
end
```

sign.rb

複雑な条件

- 色々な比較

書き方	数学	意味
$x > y$	$>$	x が y より大きい
$x \geq y$	\geq	x が y 以上
$x == y$	$=$	x と y が等しい (x=y でないことに注意)
$x < y$	$<$	x が y より小さい
$x \leq y$	\leq	x が y 以下
$x != y$	\neq	x と y が異なる

- 条件式の組合せ

書き方	意味
$x > y \ \ x == 0$	$x > y$ <u>または</u> $x == 0$
$x < y \ \&\& \ y < z$	$x < y$ <u>かつ</u> $y < z$
$!(x < y \ \&\& \ y < z)$	$(x < y \ \text{かつ} \ y < z)$ <u>でない</u>

複雑な条件

= でないことに注意
= は代入

- 色々な比較

書き方	数学	意味
$x > y$	$>$	x が y より大きい
$x \geq y$	\geq	x が y 以上
$x == y$	$=$	x と y が等しい (x=y でないことに注意)
$x < y$	$<$	x が y より小さい
$x \leq y$	\leq	x が y 以下
$x != y$	\neq	x と y が異なる

- 条件式の組合せ

書き方	意味
$x > y \ \ x == 0$	x > y <u>または</u> x == 0
$x < y \ \&\& \ y < z$	x < y <u>かつ</u> y < z
$!(x < y \ \&\& \ y < z)$	(x < y <u>かつ</u> y < z) <u>でない</u>

真偽値を与える論理演算

irb(main):003:0> $x = 3$

=> 3

irb(main):004:0> $1 < x$

=> true

irb(main):005:0> $x == 2$

=> false

どれが正しいか？

x の値が 7、y の値が 5、z の値が 3 であるとして

1. $x < y$
2. $z == y$
3. $z <= x$

どれが正しいか？

x の値が 7、y の値が 5、z の値が 3 であるとして

1. $x < y$
2. $x \leq y$
3. $y \neq z$
4. $z > x$
5. $z == x$

どれが正しいか？

x の値が 7、y の値が 5、z の値が 3 であるとして

1. $x > y \ \&\& \ z == x$
2. $x > y \ || \ z == x$
3. $!(x > y)$

どれが正しいか？

x の値が 7、y の値が 5、z の値が 3 であるとして

1. $x < y$
2. $x \leq y$
3. $x < y \ \&\& \ y \neq z$
4. $x \leq y \ || \ y == z$
5. $!(x < y \ \&\& \ y == z)$

```
def is_even(x)
  x%2 == 0
end
```

is_even.rb

```
load ("./ is_even .rb ")
```

```
def tnpo (n)
  if is_even (n)
    n/2
  else
    3*n + 1
  end
end
end
```

tnpo.rb

2.3 定義のまとめ

条件分岐: $\boxed{\text{式}_1}$ が成り立つときは $\boxed{\text{式}_2}$ を、そうでないときは $\boxed{\text{式}_3}$ を計算する条件分岐は次のように書く。このとき $\boxed{\text{式}_1}$ を条件式という。

```
if  $\boxed{\text{式}_1}$   
   $\boxed{\text{式}_2}$   
else  
   $\boxed{\text{式}_3}$   
end
```

練習

実数

- 2次方程式 $ax^2 + bx + c = 0$ の解の個数を求める `solutions(a,b,c)`. 判別式の値だけでなく、1次方程式になっている場合にも対応せよ。(quadratic.rb というファイルを作ろう。)
- 3つの異なる値 x, y, z が与えられたときの中央値を求める `median(x,y,z)`. 中央値とは大きさ順に並べたときに真ん中に来る値のことである。(median.rb というファイルを作ろう。)

教科書44ページ 練習3.3 a), c)

進捗状況の確認

1. solutions ができた時点で投票してください。

教科書44ページ 練習3.3 - a)

進捗状況の確認

1. 両方できた(できた時点で投票してください)
2. solutions だけ
3. median だけ
4. まだ何もできていない

教科書44ページ 練習3.3 - a)

今週の課題

ウェブページの説明にしたがって、2つの関数 `median1(x,y,z)` と `median2(x,y,z)` からなるプログラム `median.rb` を作成せよ.

配列・繰り返し・文字列

配列 - 1

irb(main):001:0> a = [1, 7, 3, 2]

=> [1, 7, 3, 2]

irb(main):002:0> a

=> [1, 7, 3, 2]

irb(main):003:0> a[0]

=> 1

irb(main):004:0> a[1]

=> 7

irb(main):005:0> a[3]

=> 2

配列 - 2

```
irb(main):006:0> a.length()
```

```
=> 4
```

```
irb(main):007:0> a[4]
```

```
=> nil
```

```
irb(main):008:0> a[-1]
```

```
=> 2
```

```
irb(main):009:0> a[-4]
```

```
=> 1
```

```
irb(main):010:0> a[-5]
```

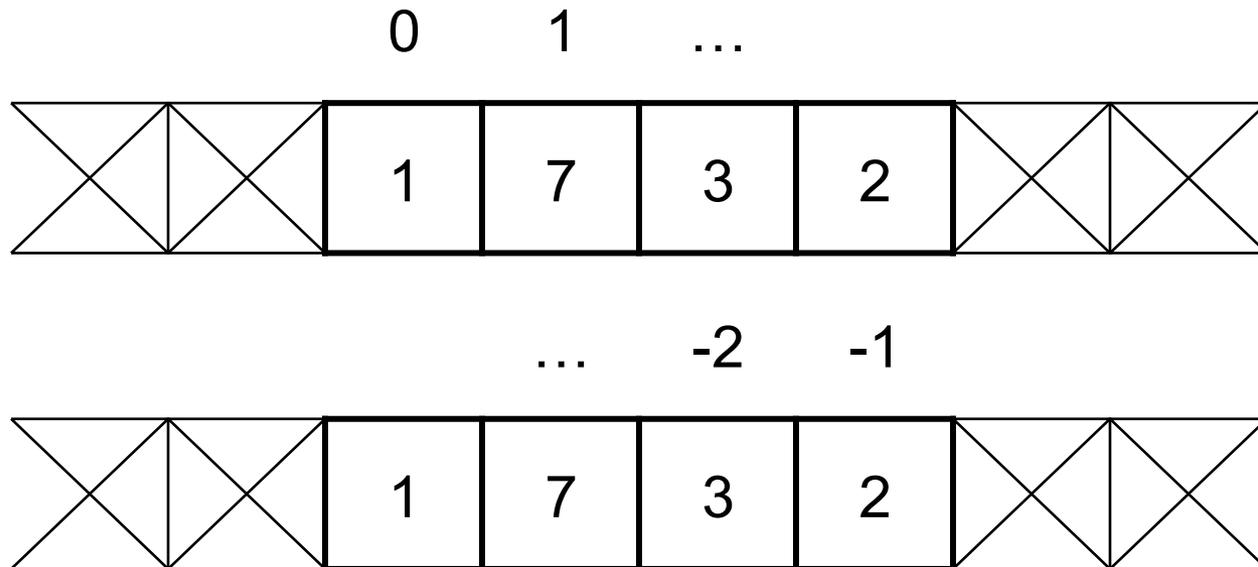
```
=> nil
```

配列 - 3

一列に並んだ容器(変数)

先頭から 0,1,..., 末尾から -1,-2,...

範囲外の場合には nil



次の結果は何？

```
irb(main):001:0> a = [3,1,4,1,5,9]
```

```
=> [3, 1, 4, 1, 5, 9]
```

```
irb(main):002:0> a.length()
```

1. nil
2. 1
3. 6
4. 9
5. [3, 1, 4, 1, 5, 9]

次の結果は何？

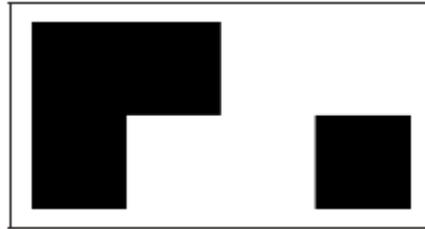
```
irb(main):003:0> a[0] = a[4]
```

```
=> 5
```

```
irb(main):004:0> a[0]+a[2]
```

1. nil
2. 1
3. 6
4. 8
5. 9
6. [3, 1, 4, 1, 5, 9]

データの表現 --- 画像の表現



isrb の
プロンプト

```
isrb(main):002:0> コントロールD
```

```
ci121164m:~ yama$ isrb
```

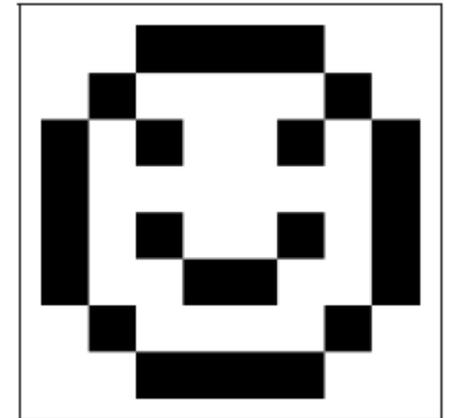
```
isrb(main):001:0> a = [[0,0,1,1],
```

```
isrb(main):002:0* [0,1,1,0]]
```

```
[[0 , 0, 1, 1], [0, 1, 1, 0]]
```

```
isrb(main):003:0> show(a)
```

```
[[0 , 0, 1, 1], [0, 1, 1, 0]]
```



単なる参考

画像の操作

isrb(main):004:0> a[0][0]

座標(0,0)の明度を参照

0

isrb(main):005:0> a[0][2]

座標(2,0)の明度を参照

1

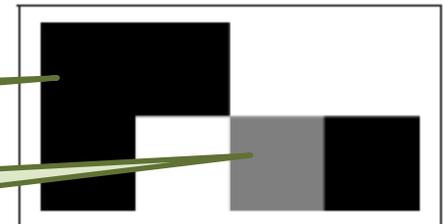
isrb(main):006:0> a[1][2]=0.5

座標(2,1)の明度を変更

0.5

座標(0,0)

座標(2,1)



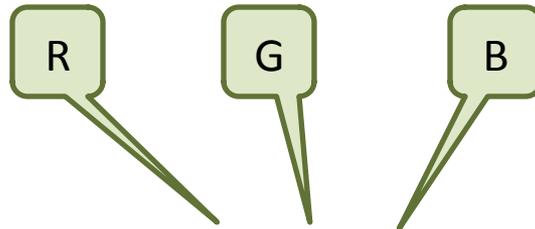
練習

- 次のようなデータを作成し、画像として表示させよ。教科書27ページ 練習2.1 b)

$$w = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

できたら 1 を投票して、練習2.1 a) を試す

カラー画像の表現 – 3次元配列



```
isrb(main):010:0> d=[[0,0,0],[0,1,0],[0,0,1]],
```

```
isrb(main):011:0* [[1,0,0],[1,1,0],[1,0,1]]
```

```
[[[0,0,0],[0,1,0],[0,0,1]],[[1,0,0],
```

```
[1,1,0],[1,0,1]]
```

```
isrb(main):012:0> show (d)
```

```
[[[0,0,0],[0,1,0],[0,0,1]],[[1,0,0],
```

```
[1,1,0],[1,0,1]]
```

練習(時間があれば...)

- カラー画像表現を使って簡単な国旗を描く。
 - たとえば、フランス、ドイツ、イタリア...



– 参考:

イエロー [1, 1, 0]

イタリアができたなら, 1を投票する

終わったら, 教科書31ページ練習2.4

どこまで進んだか？

1. 1カ国描いた
2. 2カ国描いた
3. 3カ国描いた
4. 練習3.10の画像を書いた

配列を作る: [式₀, 式₁, ..., 式_{n-1}] という式によって大きさ n の配列を作ることができる。作られた配列の i 番目には式 _{i} の値が入っている。例えば [1+2, 3+4] という式は大きさ 2 の配列を作る。

配列の参照: 式₀ [式₁] という式は、式₀が表わす配列の式₁番目の値を参照する。このような配列中の要素の位置のことを添字そえじという。例えば a[1] は変数 a に代入されている配列の 1 番目を参照する。添字が配列の範囲外だった場合には、値がないことを示す nil という特別な値を得る。

配列の中身の変更: 式₀ [式₁] = 式₂ という命令は、式₀が表わす配列の式₁番目の値を式₂の値に変更する。例えば a[1] = 5*6 は、変数 a に代入されている配列の 1 番目を 30 に変更する。

配列の大きさ: `式.length()` という式は、`式` が表わす配列の大きさを求める。

高次元の配列: 配列を作る式の中に配列を作る式を書くと、2次元以上の配列を作ることができ婦。例えば `m = [[0, 1, 2], [3, 4, 5]]` という命令は、2行3列の配列を作り、`m` に代入する。参照は `m[1][2]` のように、参照を行う `[]` を並べて書く。