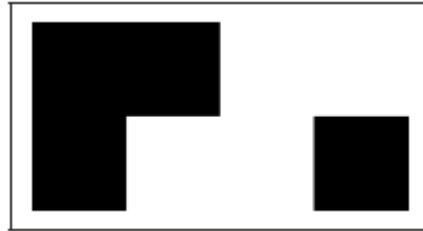


# 配列による画像の表示

# 画像の表現



isrb の  
プロンプト

```
irb(main):002:0> コントロールD
```

```
cm12345$ isrb
```

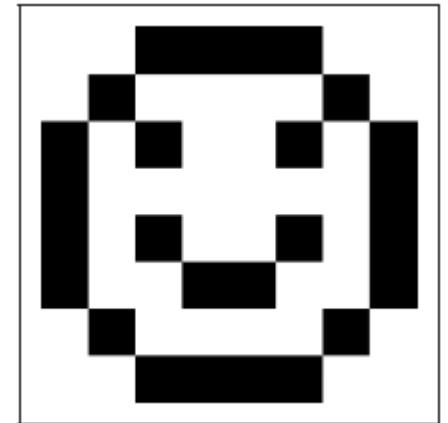
```
>> a = [[0,0,1,1],
```

```
?>       [0,1,1,0]]
```

```
=> [[0 , 0, 1, 1], [0, 1, 1, 0]]
```

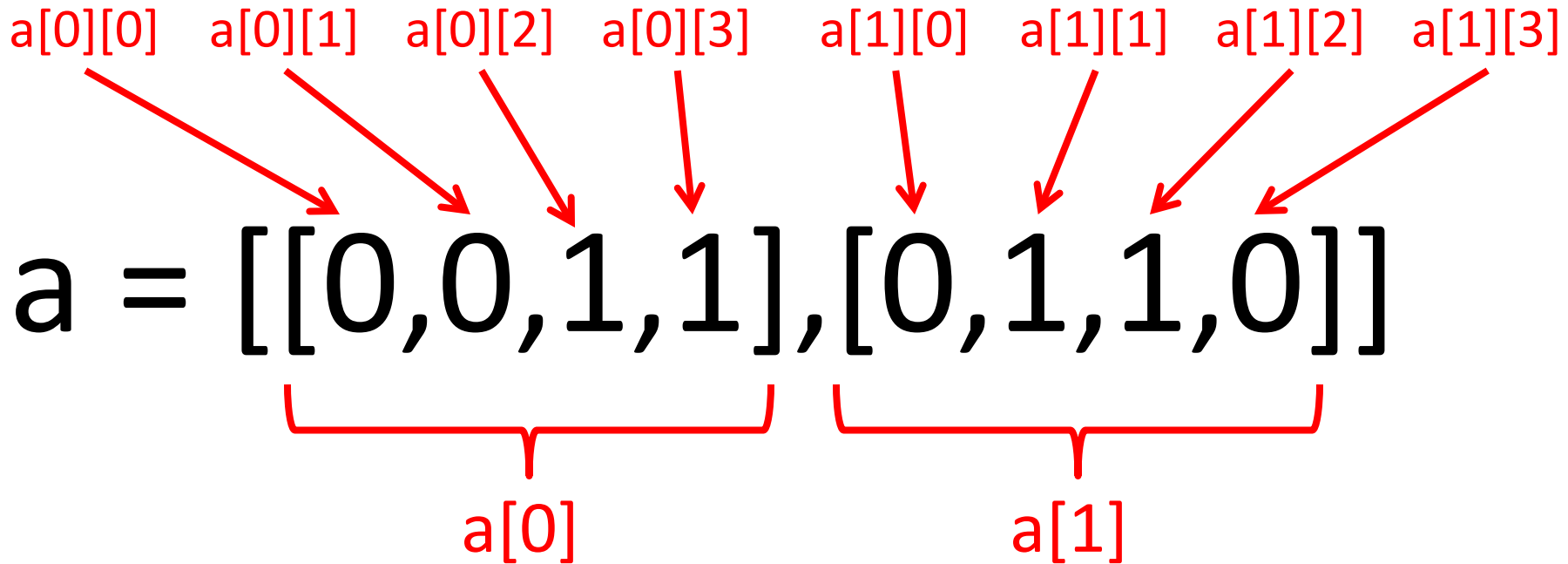
```
>> show(a)
```

```
=> nil
```



単なる参考

# 配列の要素



# 画像の操作

>> a[0][0]

座標(0,0)の明度を参照

=> 0

>> a[0][2]

座標(2,0)の明度を参照

=> 1

>> a[1][2]=0.5

座標(2,1)の明度を変更

=> 0.5

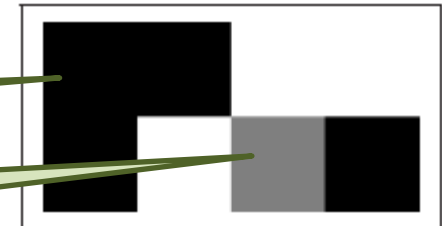
再表示

>> show(a)

=> nil

座標(0,0)

座標(2,1)



# 練習2.1

- 次のようなデータを作成し、画像として表示させよ。

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

## 練習2.2

- 画像データの座標  $(x, y)$  の点と、その周囲の8点の合計9点の明度の平均値を計算する関数 `image_average9(image,x,y)` を作れ。
  - 定義を簡単にするために...
  - `image_average.rb` というファイルに定義を格納せよ。
  - `image_average9(w,1,1)` を試してみよ。

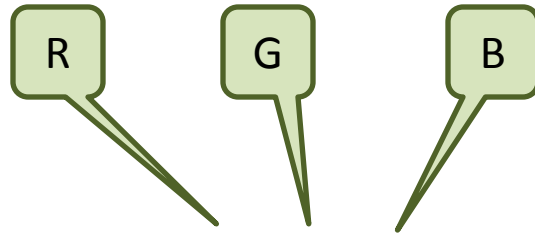
# 進捗状況の確認

1. `image_average9(w,1,1)` が動いた
2. `image_average9` が定義できた
3. `w` が表示できた
4. できない...

できてしまった人は、練習2.3をやってください。

さらに、練習問題確認プログラム `ex02.rb`

# カラー画像の表現



```
>> d=[[0,0,0],[0,1,0],[0,0,1]],
```

```
?> [[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]]]
```

```
>> show (d)
```

```
=> nil
```



## 練習2.4

- カラー画像表現を使って簡単な国旗を描いてみよ。
  - たたとえば、ドイツ、フランス...

## 2.4 定義のまとめ

配列を作る:  $[ \boxed{\text{式}_0}, \boxed{\text{式}_1}, \dots, \boxed{\text{式}_{n-1}} ]$  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  $\boxed{\text{式}_i}$  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:  $\boxed{\text{式}_0} [ \boxed{\text{式}_1} ]$  という式は、 $\boxed{\text{式}_0}$  が表わす配列の  $\boxed{\text{式}_1}$  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<sup>そえじ</sup>添字という。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:  $\boxed{\text{式}_0} [ \boxed{\text{式}_1} ] = \boxed{\text{式}_2}$  という命令は、配列の先頭を 0 番目として、 $\boxed{\text{式}_0}$  が表わす配列の  $\boxed{\text{式}_1}$  番目の値を  $\boxed{\text{式}_2}$  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:  $\boxed{\text{式}} .\text{length}()$  という式は、 $\boxed{\text{式}}$  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る。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。

**配列の表示:** `show(式)` という命令は、`式` が表わす配列 `a` の内容を画像として表示する。配列の中身は0から1までの数値でなければいけない。配列が2次元のときは、`a[y][x]` の数値が座標  $(x, y)$  の点の明るさに対応した濃淡画像が表示される。3次元のときは `a[y][x][c]` の数値が座標  $(x, y)$  の点の `c` 番目の原色の明るさに対応したカラー画像になる。ただし `c` は0が赤、1が緑、2が青に対応している。

なお、`show` は `isrb` の中からでないといけない。

# 練習問題確認プログラム

- ex02.rb (練習2.2と2.3)の結果を以下のアドレスに送れ。

[is-komaba@lyon.is.s.u-tokyo.ac.jp](mailto:is-komaba@lyon.is.s.u-tokyo.ac.jp)

- Subject:欄は、

[is-komaba](#)

としてください。

- 〆切: **11月XX日**

– 練習問題確認プログラムの結果は出席点の一部とします。結果の内容は問いません。(習得状況を知りたい。)

# 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]

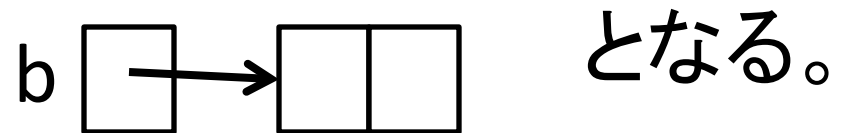
=> 5

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

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

# 値としての配列

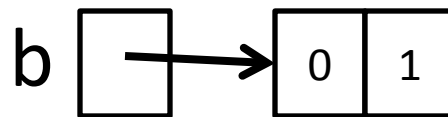
- 配列の実体は、コンピュータのメモリのどこかに割りつけられている。
- 配列を変数に代入したり、配列を関数の引数として渡したり、関数の値として返したりするときは、実体の「場所」がやりとりされる。
- たとえば、変数  $b$  に配列の実体の場所が値として入っているとき、



# 値としての配列

- 配列の実体は、コンピュータのメモリのどこかに割りつけられている。
- 配列を変数に代入したり、配列を関数の引数として渡したり、関数の値として返したりするときは、実体の「場所」がやりとりされる。
- たとえば、変数  $b$  に配列の実体の場所が値として入っているとき、

たとえば、  
 $b = [0, 1]$   
とすると、



となる。

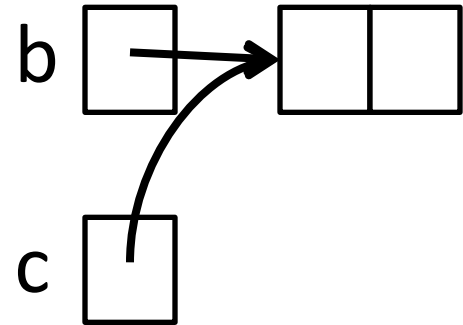


# 配列の代入

- たとえば、

$$c = b$$

とすると、同じ配列を二つの  
変数が指す。



# 配列の代入

- たとえば、

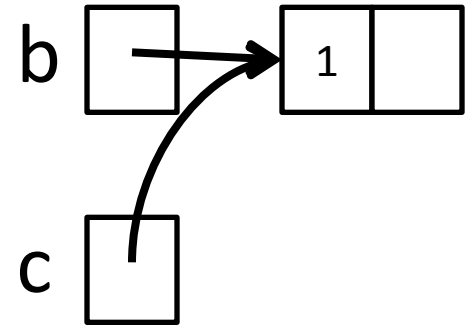
$$c = b$$

とすると、同じ配列を二つの  
変数が指す。

- このとき、

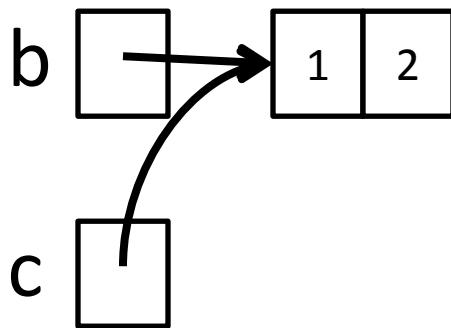
$$b[0] = 1$$

とすれば、 $c[0]$  も同じく 1 になる。



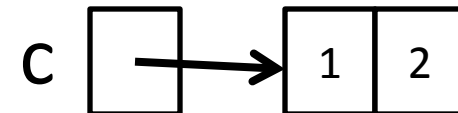
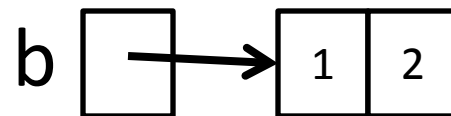
# 配列の表示

- 配列を表示させると、その要素が順に並べて表示される。場所の情報は明示されない。



$b = [1, 2]$

$c = b$

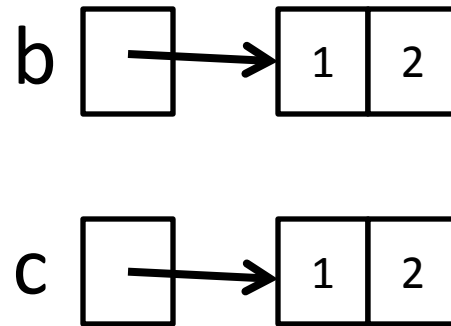
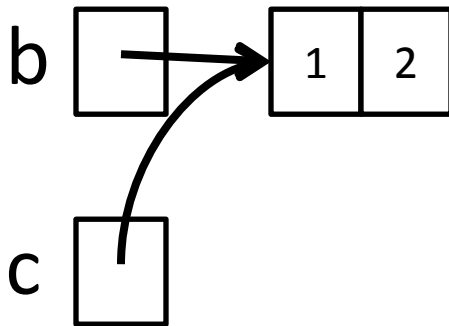


$b = [1, 2]$

$c = [1, 2]$

# 配列の表示

- 配列を表示させると、その要素が順に並べて表示される。場所の情報は明示されない。



- 表示させても、両者は区別できない。

**b** => [1, 2]      **c** => [1, 2]