

第2章 配列による画像の表示

ここまでの計算は数値を扱うものだけであった。現在のコンピュータは画像や音声のようなデータ、ワードプロセッサや電子メールのように文字列データなどを扱うこともできる。ここでは画像データの表現方法と、それを一般化した配列について見てゆこう。

2.1 画像の表現

コンピュータにおける画像の表現方法は、画像を小さなマス目に区切り、各マス目が塗りつぶされているかどうかという情報を並べて表わすのが一般的である。例えば「」という画像を拡大してみると図 2.1 のように 8×8 のマスの何箇所かが塗りつぶされているかどうかによって表現されていることが分かる。

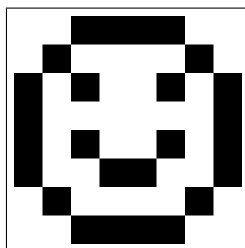


図 2.1: 画像の例

プログラムの中からこのような画像を表示する方法は一通りではなく、プログラミング言語によっても異なる。本書では画像を簡単に表示するための機能を組み込んだ `isrb` というソフトウェアを `irb` のかわりに使用するの
で、それに従って説明する。

まず `isrb` を使うために、一度 `irb` を終了し、`isrb` という名前のソフトウェアを起動する。

```
1 irb(main):002:0> Control D
2 cm12345$ isrb ↵
3 >> █
```

この `isrb` は本教科書のために作成されたソフトウェアであり、Ruby 処理系に付属しているものではない。一般のコンピュータで利用するための情報については本書冒頭 (p.1) に記載されている web ページを参照されたい。

ソフトウェアの都合上、プロンプトが「`irb(main):001:0>`」のようなものから「`>>`」に変わったが、`irb` と全く同じように式や命令を入力して計算・実行させることができる。

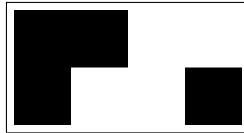


図 2.2: 簡単な画像

早速、図 2.2 のような画像を表示させよう。`isrb` では、この画像をマス目の白い部分を 1、黒い部分を 0 として次のような式で表わす。

```
1 >> a = [[0,0,1,1],
2         [0,1,1,0]]
3 => [[0, 0, 1, 1], [0, 1, 1, 0]]
```

上の画面で変数 `a` に代入されている右辺の式が 2×4 の画像データである。式が 2 行に渡っていることに注意せよ。なお、白が 1 で黒が 0 であるのは、マス目の明るさ (明度) を 0 から 1 までの数値で表わしているためである。

このデータを表示するには、`isrb` の関数 `show` を使う。

```
4 >> show(a)
5 => nil
```

これで図 2.2 のような画像が表示される。

練習 2.1 (画像データの作成) a) 図 2.1 のような画像データを作成し、表示する命令を作れ。

b) 次のようなデータを作成し、画像として表示させよ。

```
1 w = [[0,1,1,1,1,1],
2       [0,1,0,0,0,1],
3       [0,1,0,1,0,1],
4       [0,1,1,1,0,1],
5       [0,0,0,0,0,1]]
```

2.2 画像の操作

画像データは表示するだけでなく、大きさに関する情報を得たり、各点 (マス目) に関する情報を得たり変更したりするといった基本的な操作を行うことができる。各点の明度は、画像の左上の座標を (0,0) として次のような式で調べることができる。

```

6 >> a[0][0]      #座標(0,0)の明度を参照
7 => 0
8 >> a[0][2]      #座標(2,0)の明度を参照
9 => 1

```

式としては Y 座標、X 座標の順に書くことに注意せよ。この順序は奇異に思えるかも知れないが、数学で行列中の要素を「2 行 3 列」のように行番号を先に書くのと同じだと思っておけばよい。

画像データの各点の明度を変更することもできる。

```

10 >> a[1][2]=0.5  #座標(2,1)の明度を変更
11 => 0.5
12 >> show(a)     #再表示
13 => nil

```

10 行目は、「画像 a の座標 (2,1) の点」に 0.5 を代入するという意味になる。この命令を実行した後は、図 2.3 のように座標 (2,1) の点の明度が半分、つまり灰色になる。



図 2.3: 変更後の画像

画像データの内容を変更しただけでは画面に表示されている画像は変化しないが、12 行目のようにもう一度 show を実行すると変更された内容に表示が変わる。

練習 2.2 (画像データの操作) 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` は画像データ、座標 (x, y) は常に画像の内側の点になっていると仮定してよい。つまり `image` の幅と高さが h, w だったとき、 $1 \leq x \leq w-2$, $1 \leq y \leq h-2$ だと仮定してよい。(整数を整数で割ると値が切り捨てられてしまうので、そうならないように工夫せよ。)

```

1 irb(main):012:0> b = [[0,1,0],
2 irb(main):013:1*      [1,0,1],
3 irb(main):014:1*      [0,1,0]]
4 => [[0, 1, 0], [1, 0, 1], [0, 1, 0]]
5 irb(main):015:0> image_average9(b,1,1)
6 => 0.4444444444444444

```

作成した関数を使って練習 2.1a で作成した画像の座標 (4,3)、座標 (5,2) を中心とした明度の平均値を計算せよ。

(章末の練習問題: 2.3)

2.3 カラー画像の表現 *

isrb に用意されている関数 show は、カラー画像も表示できる。次の例は 3×2 のカラー画像を表示する。

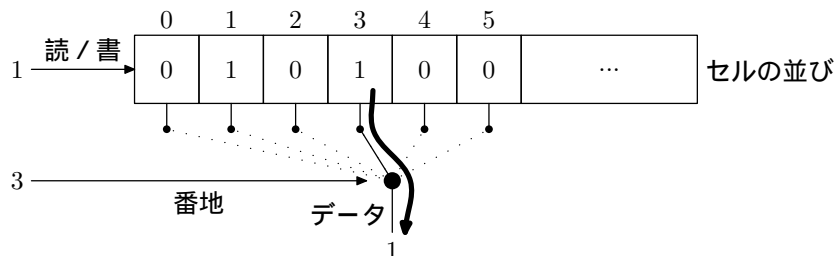
```

1 >> d=[[0,0,0],[0,1,0],[0,0,1]],
2     [[1,0,0],[1,1,0],[1,0,1]]
3 => [[0,0,0],[0,1,0],[0,0,1],[1,0,0],
4     [1,1,0],[1,0,1]]
5 >> show(d)
=> nil

```

括弧が多くて分かりづらいが、配列 d の大きさは $2 \times 3 \times 3$ の 3 次元配列である。いままで 1 つの点は 0 から 1 までの明度によって表わされていたが、カラー画像の場合は、赤、緑、青の 3 原色の明度を表わす大きさ 3 の配列にな

ここで紹介した画像を表わすデータは、一般に配列と呼ばれ、コンピュータの主記憶装置を抽象化したものである。今日のコンピュータの主記憶装置は、ランダムアクセスメモリ (RAM) という電子回路によって構成されている (「情報」(川合編, 東京大学出版会)7 章参照。)。RAM は、1 ビットの情報を記憶するセルを沢山並べ、1 つのセルを番地と呼ばれる整数によって指定して読み書きできるようにした回路である (下図)。テープやハードディスクのように頭出しが必要な記憶装置と異なり、どの番地のセルに対しても即座に読み書きができる点が特徴である。配列は添字を指定して値を読み書きできるデータであるが、基本的には添字の大小に関らずに即座に読み書きができる点で RAM と同じ特徴を持っている。



る。このような表現方法を、red, green, blue の頭文字から RGB 画像と呼ぶ。このとき、画像の座標 (x, y) の赤成分は $d[y][x][0]$ 、緑成分は $d[y][x][1]$ 、青成分は $d[y][x][2]$ になる。例えば座標 $(2, 1)$ の色は $d[1][2][0]$ は 1, $d[1][2][1]$ は 0, $d[1][2][2]$ は 1 なので、赤と青を合成した紫になる。

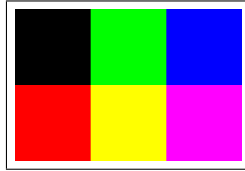


図 2.4: カラー画像

(章末の練習問題: 2.4)

2.4 定義のまとめ

配列を作る: $[\text{式}_0, \text{式}_1, \dots, \text{式}_{n-1}]$ 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 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: $\text{式}_0[\text{式}_1]$ 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: $\text{式}_0[\text{式}_1] = \text{式}_2$ という命令は、配列の先頭を 0 番目として、 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: $\text{式}.length()$ 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: $\text{show}(\text{式})$ 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 の中からでないと使用できない。

2.5 章末問題

練習 2.3 (配列) a) a は数値が並んだ 1 次元の配列とする。 a の x 番目の値と、その前後の値の合計値を求める $\text{sum3}(a, x)$ を作れ。ただし x の値は 1 以上かつ (a の大きさ - 1) 未満であり、必ず前後に値があるものと仮定せよ。例えば a が $[1, 2, 3, 4]$ のとき $\text{sum3}(a, 1)$ は 6 に、 $\text{sum3}(a, 2)$ は 9 になる。

b) a の x 番目の値と、その前後の値の平均値を求める $\text{array_average3}(a, x)$ を作れ。ただし x の値は 1 以上かつ (a の大きさ - 1) 未満であると仮定せよ。例えば a が $[1, 2, 3, 4]$ のとき $\text{array_average3}(a, 1)$ は 2.0, $\text{array_average3}(a, 2)$ は 3.0 になる。

練習 2.4 (国旗) カラー画像表現を使って簡単な国旗を描いてみよ。例:

