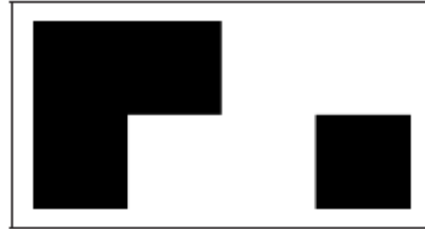


# 配列

# データの表現 --- 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
```

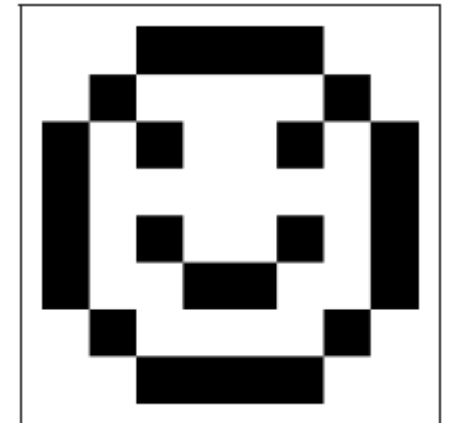
```
>> a = [[0,0,1,1],
```

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

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

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

```
=> nil
```



単なる参考

# 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

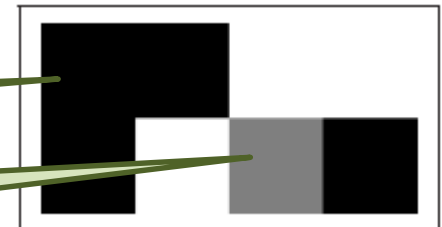
再表示

>> show(a)

=> nil

座標(0,0)

座標(2,1)



配列を作る: [ 式<sub>0</sub>, 式<sub>1</sub>, ..., 式<sub>n-1</sub> ] 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 <sub>$i$</sub> 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<sub>0</sub>[ 式<sub>1</sub> ] という式は、式<sub>0</sub>が表わす配列の式<sub>1</sub>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<sub>0</sub>[ 式<sub>1</sub> ] = 式<sub>2</sub> という命令は、式<sub>0</sub>が表わす配列の式<sub>1</sub>番目の値を式<sub>2</sub>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。

# 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

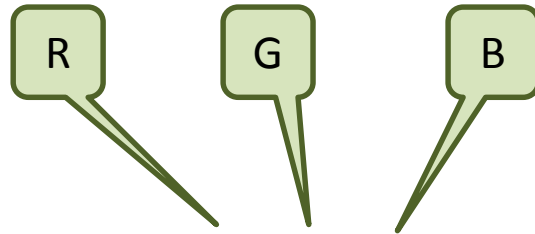
2. 1

3. 6

4. 9

5. [3, 1, 4, 1, 5, 9]

# 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

- 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

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

# 練習

- 配列中の2つの要素を入れ替える関数 `swap(a,i,j)` を作れ。
- `a` の  $x$  番目の値と、その前後の値の平均値を求める `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 `array_average3(a,1)` は 2.0, `array_average3(a,2)` は 3.0 になる。