

2 次元リスト

前回までは、リスト内の値が直線的（1 次元的）に並んでいるリストを扱いました。今回は碁盤の目やオセロのように縦方向と横方向に部屋があるリスト（授業では 2 次元リストと呼ぶ）について学びます。

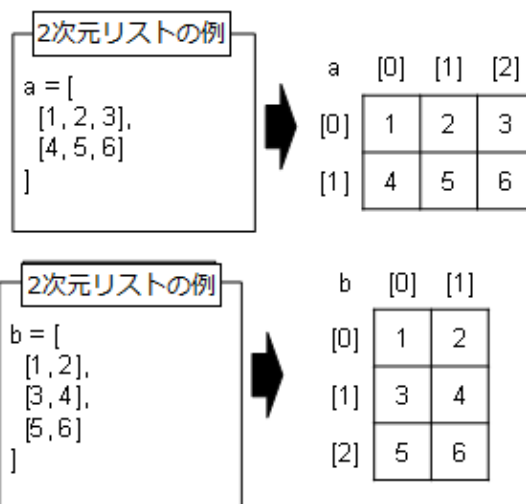
右側の図が 2 次元リストのイメージになります。

縦方向は「行」で、横方向が「列」になります。

右図のリスト a では、2 行目 3 列目の値は「6」です。

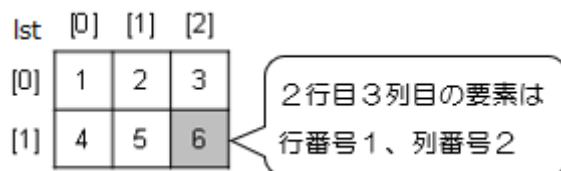
ここで、注意しなければならないのは、プログラムでは行番号と列番号は 0 からスタートすることです。

プログラムでは次のように書きます。

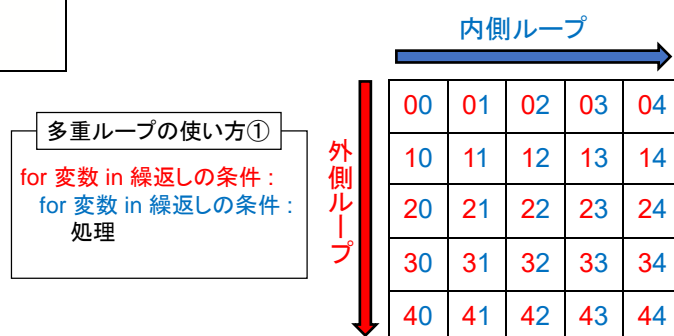


(例 1) 次のプログラムを入力して、実行してみよう。

```
1 lst = [
2     [1, 2, 3],
3     [4, 5, 6]
4 ]
5
6 print(lst) #2 次元リストを表示
7 print(lst[1][2]) #2 行目 3 列目を表示
```



ここで、前回まで習ったように繰り返し処理を用いてリストの各要素を取得する方法を考えていきます。以前の授業で用いた「多重ループ」を使うことでリストの各要素を順に取得していくことができます。



(再掲) 2 重ループですが、外側のループが縦方向に対応しており、内側のループが横方向に対応しています。

(例 2) 次のプログラムを入力して実行してみよう。

```
1 lst = [
2     [1, 2, 3],
3     [4, 5, 6]
4 ]
5 #2 重ループを用いた 2 次元リストの各要素の取得
6 for i in range(2): #行(縦)
7     for j in range(3): #列(横)
8         print(lst[i][j], end= " ")
9     print()
```

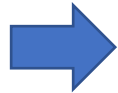
(問 1) 次の 2 次元リストの各要素を、(例 2) のような形で出力するプログラムを作ってみよう

```
1 lst = [
2     [1, 2, 3, 4, 5],
3     ['a', 'b', 'c', 'd', 'e']
4     [1.1, 1.2, 1.3, 1.4, 1.5]
5 ]
```

前回の授業で学習したように、リストを使った繰り返し処理でループ回数を数値で書いてしまうと、変更する部分が増えたり、書き換え忘れが起こったりする可能性があります。そこで、前回と同様にリストの要素数を取得する `len()` を使います。2次元リストにおける `len()` の値はどうなるのでしょうか。

(例3) 次のプログラムを入力して実行してみよう。

```
1 lst = [
2     [1, 2, 3],
3     [4, 5, 6]
4 ]
5 print(len(lst))
6 print(len(lst[0]))
7 print(len(lst[1]))
```

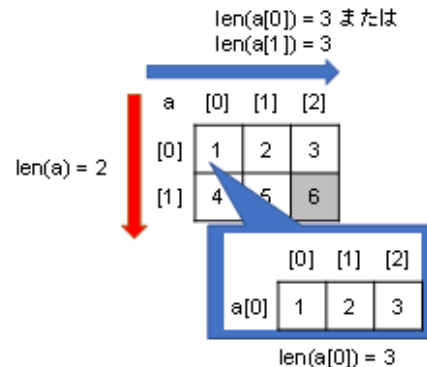


実行した結果

2次元リストの例

```
a = [
    [1, 2, 3],
    [4, 5, 6]
]

len(a)
len(a[0])
```



2次元リストで `len(lst)` を実行すると、行方向（縦方向）の要素数がわかりました。行番号 `N` の行における、列方向（横方向）の要素数は、`len(lst[N])` で取得できます。

(問2) (例2) のプログラムについて、`range()`内の数値を、`len()` を使った形に書き直そう。

(例4) 2次元リストは、画像を表現するために使われます。ここでは、白黒画像を表現することを考えてみます。白黒画像は白 (1) か黒 (0) で表現できます。次のプログラムのように、要素がすべて0の2次元リスト `lst` からは、10×10の真っ黒な画像が作られます。

```
lst [10][10] = [
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
]
```

```
1 #10×10のすべて黒(0)で画像を作成
2 lst = [[0] * 10 for i in range(10)]
3
4 #画像として表現(模擬)
5 for i in range(len(lst)):
6     for j in range(len(lst[i])):
7         if lst[i][j] == 0:
8             print("■", end=" ")
9         else:
10            print(" ", end=" ")
11 print()
```

(問3) 次のプログラムはどんな形になるか考えてみよう。考えてみたら、上のプログラムの3行目の空白の部分（#画像として表現(模擬)の上）に入力して確認してみよう

```
3 for i in range(len(lst)):
4     for j in range(len(lst[i])):
5         if(i == 1 or i == 2):
6             if(j > 0 and j < 9):
7                 lst[i][j] = 1
```

```
8         if(i >= 3 and i <= 8):
9             if(j == 4 or j == 5):
10                lst[i][j] = 1
```

※8行目のifの先頭は5行目のifの先頭と合わせる

発展 白くしたい部分を1にすることで様々な絵が描ける。これまで習った、2次元リストの各要素の取得、条件処理、繰り返し処理などを使って、文字や記号などを表す絵を1つ作成してみよう。