# 1年 LC情報 Python「配列③」　　　　　　　　　　　　　　　　　　　　　　　No.10

## ダイアグラム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。

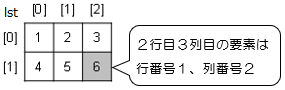
縦方向は「行」で、横方向が「列」になります。

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。

1. 次のプログラムを入力して、実行してみよう。

内側ループ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 00 | 01 | 02 | 03 | 04 |
| 10 | 11 | 12 | 13 | 14 |
| 20 | 21 | 22 | 23 | 24 |
| 30 | 31 | 32 | 33 | 34 |
| 40 | 41 | 42 | 43 | 44 |

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | lst = [    [1, 2, 3],    [4, 5, 6]  ]  print(lst) #2次元リストを表示  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

外側ループ

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。

1. 次のプログラムを入力して実行してみよう。

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

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

**１ 年（　　　）組（　　　）番　名前（　　　　　　　　　　　　　）**

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( ) の値はどうなるでしょうか。

1. 次のプログラムを入力して実行してみよう。

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

ダイアグラム

自動的に生成された説明実行した結果

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。

1. （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。

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

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]

]

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

|  |  |
| --- | --- |
| 8  9  10 | if(i >= 3 and i <= 8):        if(j == 4 or j == 5):          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。