

配列（リスト）で条件処理や繰り返し処理を利用する

リストとこれまで学んできた条件処理や繰り返し処理を組み合わせたプログラムについて学んでいきましょう。

前回までは、リストに格納されているデータにアクセスするためには、1 つずつ指定していました。

リストの大きさ（リストに含まれている要素の数）が大きくなると、記述することが多くなる可能性があります。

そこで、繰り返し処理を使ってアクセスすることを考えます。

(例 1) 次のプログラムを入力し実行してみよう

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



(例 2) 次のプログラムを入力し実行してみよう

```
1 lst = [1, 2, 3, 4, 5]
2 for i in range(5):
3     print(lst[i])
```

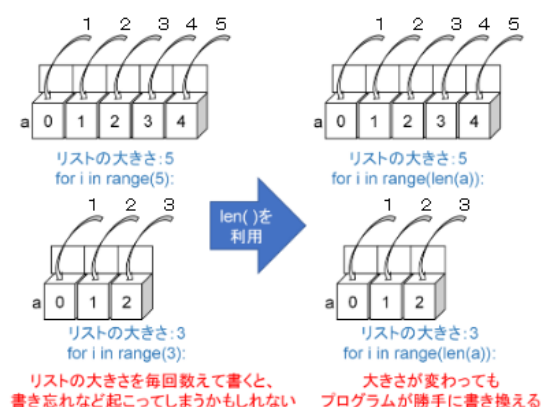
ここで、(例 2) ですが、リストの大きさを変えるたびに「range()」の数値を変える必要があります。

Python では、リストの大きさを教えてくれる処理として「len()」というものがあります。これを使うと

(例 2) は次のように書き換えることができ、リストの大きさが増えても自動で計算してくれます。

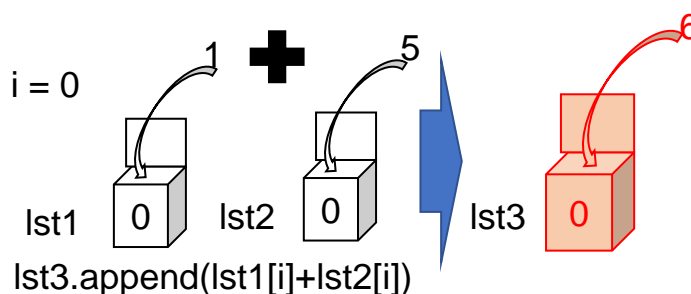
(例 3) 次のプログラムを入力し実行してみよう

```
1 lst = [1, 2, 3, 4]
2 for i in range(len(lst)):
3     print(lst[i])
```



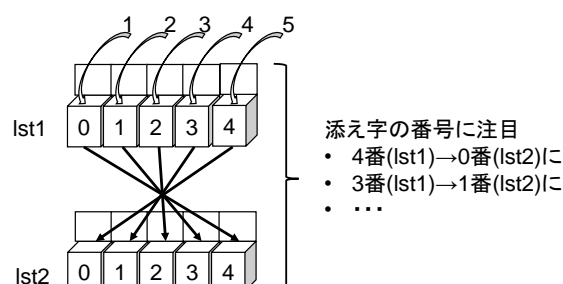
(例 4) 次のプログラムを入力し実行してみよう (No.8 の例 8 を、for を利用したもの)

```
1 lst1 = [1, 2, 3, 4]
2 lst2 = [5, 6, 7, 8]
3 lst3 = []
4
5 for i in range(len(lst1)):
6     lst3.append(lst1[i]+lst2[i])
7
8 print(lst3)
```



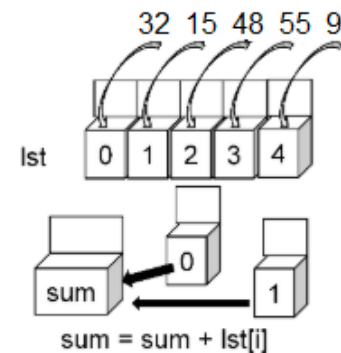
(問 1) (例 4) を参考に、

lst1 = [1, 2, 3, 4, 5] を使用して、lst2 = [5, 4, 3, 2, 1] となるようにプログラムをしてみましょう。



(問2) lst = [32, 15, 48, 55, 9]の

合計値と平均値を求めてみよう。sum に合計値を入れます。



```

1 lst = [32, 15, 48, 55, 9]
2 sum = 0
3
4
5 ave =
6 print("合計:", sum)
7 print("平均:", ave)

```

繰り返し処理を使うことで、リスト内のすべての要素にアクセスすることが簡単にできました。

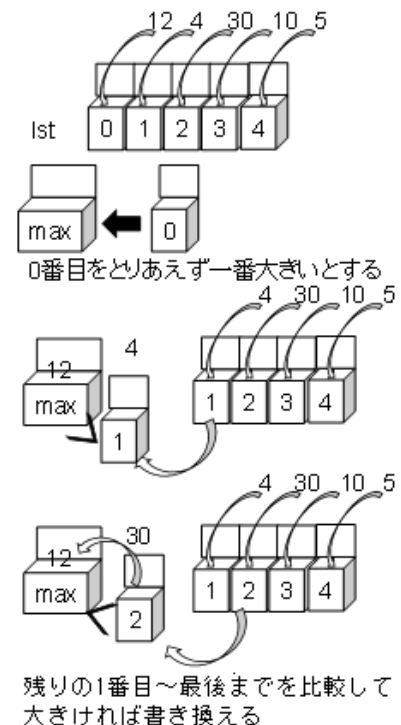
次に分岐処理を使ってみましょう。

(例5) 配列の中から最大値を見つけ出すプログラム

```

1 lst = [12, 4, 30, 10, 5]
2 max = lst[0]
3
4 for i in range(1, len(lst)):
5     if (max < lst[i]):
6         max = lst[i]
7 print("max = ", max)

```



(問3) (例5)を参考に、一番小さい数値を見つけ出すプログラムを書いてみましょう。

リストデータは(例5)と同じとします。

一番小さい数値を入れる変数は min とします。

(例6) 次のプログラムを入力して実行してみよう。

(キーボード入力から配列を作るプログラム)

0 が入力されるまで数値を配列に入れていきます。

```

1 lst = []
2 num = input("入力:")
3 while (num != 0):
4     lst.append(int(num))
5     num = input("入力:")
6
7 print("入力された数値の配列は、", lst)
8 print()

```

発展 (例6)の続きに、さらにキーボードから入力した数値が、配列の先頭から何番目にあるのかをチェックするプログラムを追加してみましょう。