



po-5. 繰り返し処理, リスト, 辞書

(Python プログラミング体験学習, 全9回)

<https://www.kkaneko.jp/cc/po/index.html>

金子邦彦





アウトライン

5-1 復習

5-2 繰り返し (ループ)

5-3 リスト, for

5-4 辞書

- **データの形式のバリエーション**について知る.
- **リスト, 辞書**を Python で扱えるようになる.



5-1. 復習

リスト

リストは, データの並びである

list

0	1	2	3	4
8	6	4	2	3

list

0	1	2	3	4
8.8	6.6	4.4	2.2	3.3

繰り返しのプログラム例



リストの
組み立て

```
1 x = [8, 6, 4, 2, 3]
2 y = [0, 0, 0, 0, 0]
3 for i in [0, 1, 2, 3, 4]:
4     y[i] = x[i] * 1.1
```

「 $y[i] = x[i] * 1.1$ 」を
 i の値を変えながら
5回繰り返す



5-2. 繰り返し (ループ)

確認クイズ



次のプログラムで i の値はどのように変化する か？

i の値 : → → → →

```
1 x = [8, 6, 4, 2, 3]
```

```
2 y = [0, 0, 0, 0, 0]
```

```
3 for i in [0, 1, 2, 3, 4]:
```

```
4     y[i] = x[i] * 1.1
```

「 $y[i] = x[i] * 1.1$ 」を
i の値を変えながら
5回繰り返す

答え合わせ



次のプログラムで i の値はどのように変化する か？

i の値 : 0 → 1 → 2 → 3 → 4

```
1 x = [8, 6, 4, 2, 3]
2 y = [0, 0, 0, 0, 0]
3 for i in [0, 1, 2, 3, 4]:
4     y[i] = x[i] * 1.1
```

「 $y[i] = x[i] * 1.1$ 」を
i の値を変えながら
5回繰り返す



実習の指示

- 資料： **10**～**15**
- 次のことを理解しマスターする
 - 処理の繰り返しにより, リストのすべての要素を処理できる

実習



① **ウェブブラウザ**を起動する

② **Python Tutor** を使いたないので、次の URL を開く

<http://www.pythontutor.com/>

※ Internet Explorer でうまく動かない場合がある

→ うまく動かないときは Google Chrome を試してください

※ 途中で 「Server Busy . . .」 というメッセージが出る
ことがある。

→ 混雑している。 少し（数秒から数十秒）待つと自動で表示
が変わる（変わらない場合には、操作をもう一度行ってみ
る）

※ 日本語モードはない。英語で使う



③ 「Python Tutor」 をクリック

← → ↻ ⓘ 保護されていない通信 | pythontutor.com ☆ APP 2

VISUALIZE CODE AND GET LIVE HELP

Learn Python, Java, C, C++, JavaScript, and Ruby

[Python Tutor](#) (created by [Philip Guo](#)) helps people overcome a fundamental barrier to learning programming: understanding what happens as the computer runs each line of code.

Write code in your web browser, see it visualized step by step, and get live help from volunteers.

Related services: [Java Tutor](#), [C Tutor](#), [C++ Tutor](#), [JavaScript Tutor](#), [Ruby Tutor](#)

Over five million people in more than 180 countries have used Python Tutor to visualize over 100 million pieces of code, often as a supplement to textbooks, lectures, and online tutorials.

[Visualize your code and get live help now](#)



Get live help!

Start private chat

These Python Tutor users are asking for help right now. Please volunteer to help!
user_041 from Singapore, Singapore needs help with Python3 - 2 people chatting - requested 12 minutes ago)
user_91c from Vélizy-Villacoublay, France needs help with Python3 - [click to help](#) minutes ago)
user_985 from Seattle, Washington, US needs help with Python3 - [click to help](#) (-

「Python 3.6」になっている

Write code in Python 3.6

```
1 |
```

エディタ

Help improve this tool by completing a [short user survey](#)

実行のためのボタン

Visualize Execution

Live Programming Mode

hide exited frames [default] inline primitives but don't nest objects [default]

draw pointers as arrows [default]

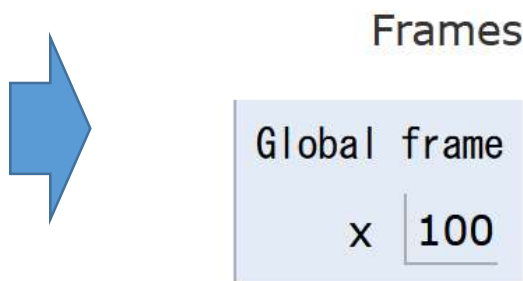
Python Tutor でのプログラム実行手順



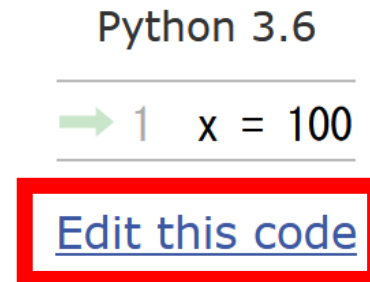
(1) 「**Visualize Execution**」をクリック。



(2) 「**Last**」をクリック。



(3) 結果を確認する。



(4) 「**Edit this code**」をクリックして戻る



④ 次のプログラムを実行し，結果を確認しなさい

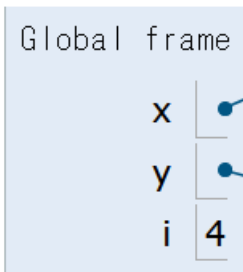
```
1 x = [8, 6, 4, 2, 3];
2 y = [0, 0, 0, 0, 0];
3 for i in [0, 1, 2, 3, 4]:
4     print(i)
5     y[i] = x[i] * 1.1
```

Print output (drag lower right corner to resize)

```
U
1
2
3
4
```

Frames

Objects



0	1	2	3	4
8	6	4	2	3

0	1	2	3	4
8.8	6.6	4.4	2.2	3.3



5-3. リスト, for



問いかけ

- 順序のあるデータを想像できますか？
- そのデータに**挿入**したり，**削除**したりがあります
- **サイズ**が**増えたり**，**減ったり**します

リストの性質



- 順序のあるデータ
- 要素の削除, 挿入によりサイズが増減する

list

0	1	2	3	4
15	8	6	32	23

リストの組み立て, 要素の挿入, 要素の削除



list

0	1	2	3	4
15	8	6	32	23

リストの組み立て

```
x = [15, 8, 6, 32, 23];
```



要素の挿入

```
x.append(4)
```

0	1	2	3	4	5
15	8	6	32	23	4



要素の削除

```
x.remove(8)
```

0	1	2	3	4
15	6	32	23	4



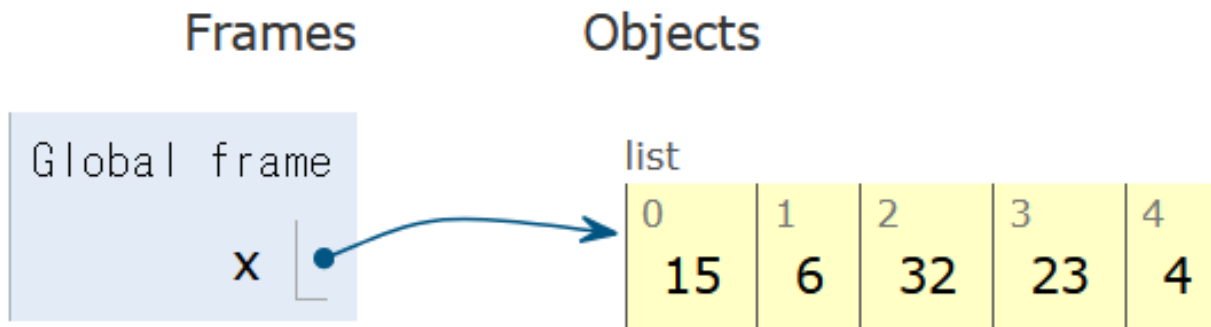
実習の指示

- 資料：20～25
- 次のことを理解しマスターする
 - リストでは，要素の追加と削除ができる
 - for は，リストに用いることができる



① 次のプログラムを実行し，結果を確認しなさい

```
1 x = [15, 8, 6, 32, 23];  
2 x.append(4)  
3 x.remove(8)
```



表示を確認



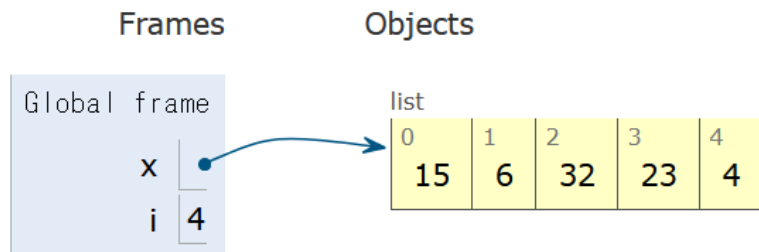
② 次のプログラムを実行し，結果を確認しなさい

```
1 x = [15, 8, 6, 32, 23];
2 x.append(4)
3 x.remove(8)
4
5 for i in x:
6     print(i)
```

「for」は，リストに用いることができる。

Print output (drag lower right corner to resize)

```
15
6
32
23
4
```



表示を確認



5-4. 辞書

キー	値 (バリュー)
1	Red
2	Yellow
3	Blue

- **キーと値 (バリュー) のペア**の集まり
- 同じ値の**キーは2回以上登場しない**

繰り返しのプログラム例

辞書の 組み立て



```
1 d = {1: "Red", 2: "Yellow", 3: "Blue"}
```

```
2
```

```
3 for i in d:  
4     print(i)
```

辞書のキー 1, 2, 3 のそれぞれについて, 値 (バリュー) を表示



実習の指示

- 資料：26～28
- 次のことを理解しマスターする
 - forは，辞書に用いることができる



① 次のプログラムを実行し，結果を確認しなさい

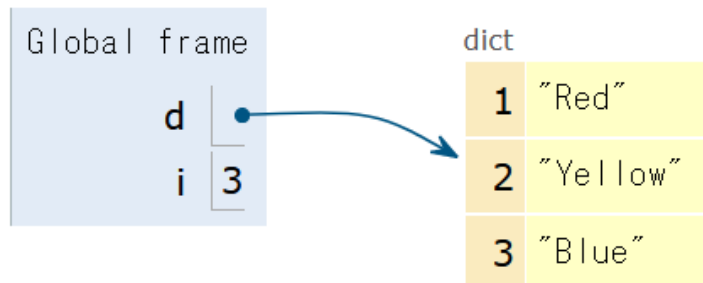
```
1 d = {1: "Red", 2: "Yellow", 3: "Blue"}
2
3 for i in d:
4     print(d[i])
```

Print output (drag lower right corner to resize)

```
Red
Yellow
Blue
```

Frames

Objects



表示を確認



② 次のプログラムを実行し，結果を確認しなさい

```
1 d = {1: "Red", 2: "Yellow", 3: "Blue"}
2 d[4] = "Orange"
3
4 for i in d:
5     print(d[i])
```

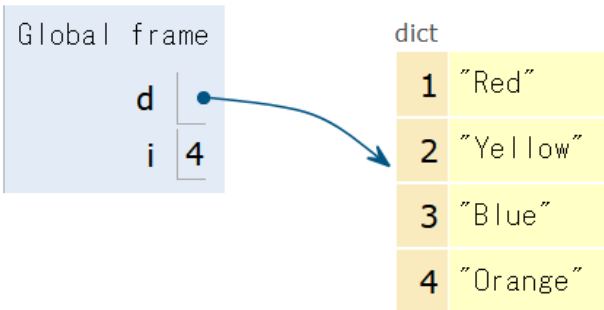
新しいキー，値（バリュー）
の追加

Print output (drag lower right corner to resize)

```
Red
Yellow
Blue
Orange
```

Frames

Objects



表示を確認



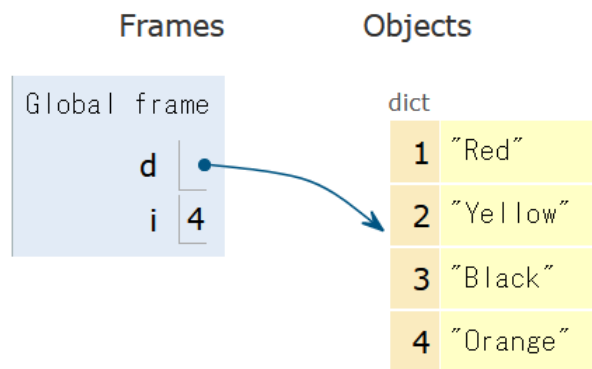
③ 次のプログラムを実行し，結果を確認しなさい

```
1 d = {1: "Red", 2: "Yellow", 3: "Blue"}
2 d[4] = "Orange"
3 d[3] = "Black"
4
5 for i in d:
6     print(d[i])
```

値（バリュー）の置き換え

Print output (drag lower right corner to resize)

```
Red
Yellow
Black
Orange
```



表示を確認

全体まとめ



- **処理の繰り返し**により, **リスト**や**辞書**のすべての要素を処理できる
- **リスト**や**辞書**で**処理の繰り返し**を行うときは, **for**が便利である
- **リスト**は, **順序のあるデータ**である. **要素の削除**, **挿入**により**サイズが増減**する
- **辞書**は, **キーと値 (バリュー) のペア**の集まりである. **同じ値のキーは2回以上登場しない**



「リスト」を実習できる オンラインサイトの紹介



「リスト」を実習できる オンラインサイトの紹介



リストとは、順序の付いたデータの並び

パソコン演習



- ① Chrome ウェブブラウザを起動する
- ② 次の URL を開く

<https://visualgo.net/ja>

- ③ 「**連結リスト**」をクリック



パソコン演習



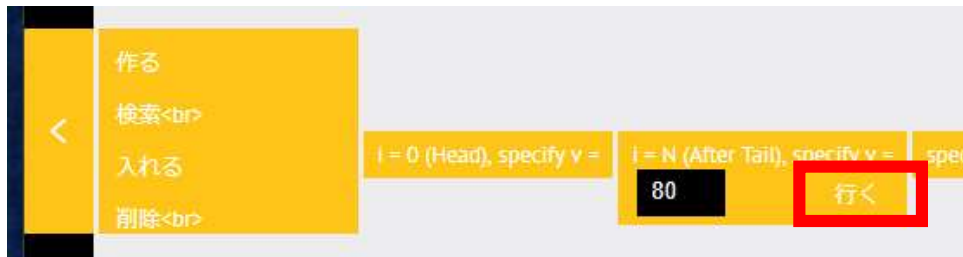
- ④ 説明が出る. **ESC キー**を押して, 説明を消す
- ⑤ 左下のメニューで「**入れる**」をクリックし,
「**i = N (After tail), specify v =**」を選ぶ



パソコン演習



- ⑥ 値が「80」のように表示されるので、確認したら「行く」をクリック



- ⑦ 末尾にデータが増えるので、確認する

