第5回　内包表記

目次

内包表記(コンプリヘンション)

- Pythonでは、リスト等に含まれる複数の要素に対して何か処理を行いたい場合はfor文を使うことができます。
- しかし、簡単な処理でもfor文の場合には複数行に渡るブロックの記述が必要です。

リストaの各要素を2乗したリストbを作成する例

```
a = [2, 4, 6, 8, 10]
b = []
for i in a:
    b.append(i*i)
print(b)
```

- 内包表記とは？
  - Pythonでは、内包表記という構文を使うと、リストやディクショナリを加工するような処理をブロックを使わずに1行で簡潔に書けるようになります。
  - しかも、実行速度も速くなります。

リスト内包表記(1)

- リスト内包表記は、あるシーケンスをもとにして、新しいリストを返す式として記述します。
  - 構文

リスト変数 = [一時変数による式 for 一時変数 in シーケンス]

  - 例:

```
a = [2, 4, 6, 8, 10]
b = []
for i in a:
    b.append(i*i)
print(b)
```

これまでのfor文の場合

＝

```
a = [2, 4, 6, 8, 10]
b = [i * i for i in a]
print(b)
```

内包表記の場合

リストa内の要素をiに取り出してきて、iを基にリストbを作る

リスト内包表記(2)

- 要素を追加する条件の指定
  - 内包表記にはifを追加することができ、条件がTrueの場合のみ要素が追加されます。
  - 構文

リスト変数 = [一時変数による式 for 一時変数 in シーケンス if 条件式]

  - 例:

```
a = [2, 4, 6, 8, 10]
b = []
for i in a:
    if i != 6:
        b.append(i*i)
print(b)
```

これまでのfor文の場合

=

```
a = [2, 4, 6, 8, 10]
b = [i * i for i in a if i != 6]
print(b)
```

iが6以外なら追加

内包表記の場合

リスト内包表記(3)

- 元となる要素の値によって追加する値の式を切り替えたい場合
  - 先ほどのifは要素を追加するかしないかを記述するための表記でしたが、追加する要素の式を切り替えたい場合には、forの前にifを書きます。
  - 構文

リスト変数 = [条件式が成り立つ場合の式 if 条件式 else 条件式が成り立たない場合の式 for 一時変数 in シーケンス]

  - 例:bmiが25.0以上の場合「太っている」それ以外「太っていない」という文字列に変換しなさい。

```
bmi = [25.4, 23.2, 26.7]
b = ["太っている" if i >= 25.0 else "太っていない" for i in bmi]
print(b)
```

```
['太っている', '太っていない', '太っている']
```

リスト内包表記(4)

- 元となる要素のインデックスも追加する要素の値や条件に加味したい場合
  - enumerate関数を使うことで、元となるリストのインデックス値を取り出すことができ、追加する要素の値や条件の材料にすることができます。
  - 構文

リスト変数 = [一時変数による式 for インデックス, 一時変数 in enumerate(シーケンス)]

  - 例：インデックス値をもとに、番号名簿を作る

```
a = ["鈴木", "田中", "加藤"]
b = [str(i+1) + "番：" + j for i, j in enumerate(a)]
print(b)
```

```
['1番：鈴木', '2番：田中', '3番：加藤']
```


リスト内包表記(5)

- 複数リストの要素全ての組合せから新たなリストを作る
  - 内包表記では、forを複数記述することで、複数のリストのすべての要素の組合せから新たなリストを作成することができます。
  - 構文

リスト変数 = [一時変数iとjによる式 for 一時変数i in シーケンスA for 一時変数j in シーケンスB]

    - この構文では、シーケンスAとシーケンスBのすべての要素の組合せが評価されます。
  - 例：ある大学にはAからCまでの棟があり、それぞれ3階建である。A棟1階からC棟3階までの要素を持つリストを作りなさい。

```
a = ["A棟", "B棟", "C棟"]
b = ["1階", "2階", "3階"]
c = [i + j for i in a for j in b]
print(c)
```

```
['A棟1階', 'A棟2階', 'A棟3階', 'B棟1階', 'B棟2階', 'B棟3階', 'C棟1階', 'C棟2階', 'C棟3階']
```

リスト内包表記(6)

- 複数リストの同じ順番の要素から新たなリストを作る
  - zip関数を使うことで、複数のリストから同じ順番に要素を取り出して、新たなリストを作成することができます。
  - 構文

リスト変数 = [一時変数による式 for 一時変数i, 一時変数j in zip(シーケンスA, シーケンスB)]

  - 例:3名の生徒の英語と数学の合計点を求める。

```
eigo = [80, 35, 65]
sugaku = [64, 76, 35]
goukei = [i + j for i, j in zip(eigo, sugaku)]
print(goukei)
```

```
[144, 111, 100]
```


ディクショナリ内包表記

- リストと同様にディクショナリにおいても内包表記が可能です。ディクショナリの場合には波括弧{}を使います。
  - 構文

ディクショナリ変数 = { 新たなキー:新たな値 for 一時キー変数,一時値変数 in 既存ディクショナリ.items() }

  - 例:市名をキー、市外局番を値とするディクショナリから、キーと値が逆のディクショナリを作る

```
a = {"静岡市":"054", "浜松市":"053", "沼津市":"055"}
b = {v:k for k, v in a.items()}
print(b)
```

```
{'054': '静岡市', '053': '浜松市', '055': '沼津市'}
```

課題5

- 3名の生徒の名前、英語の成績、数学の成績が、name、eigo、sugakuというリストで与えられたとします。
- 英語と数学の合計が120点以上の場合「名前：合格」、それ以外の場合「名前：不合格」というリストを内包表記で作ってください。
- whileを使った例を下記に示します。

```
name = ["鈴木", "田中", "加藤"]
eigo = [80, 35, 65]
sugaku = [64, 76, 35]
kekka = []
i = 0
while i < len(name):
    goukei = eigo[i] + sugaku[i]
    seiseki = ""
    if goukei >= 120:
        seiseki = "合格"
    else:
        seiseki = "不合格"
    kekka.append(name[i] + ":" + seiseki)
    i += 1
print(kekka)
```

```
['鈴木:合格', '田中:不合格', '加藤:不合格']
```