

プログラミング (Python)

第4回 繰り返し for文 while文

早稲田大学本庄高等学院 2020年度版

飯島涼



復習 if else 文(条件分岐)

- 場合によって実行する内容を変える命令
 - もし…だったら, もしxxxだったら,

書き方

行末にコロンを入れる

	<code>if (条件):</code>	
Tab キー		条件を満たした時に実行される内容
	<code>else:</code>	
Tab キー		条件に合わなかった時に実行される内容

インデント

条件分岐で動作させたい内容は、if, elseを書いた行より
決まった数だけ右にずらして書く => Tab キーを押してずらす

行末にコロンを入れる

```
if (条件):
```

Tab
キー

条件を満たした時に実行される内容

```
else:
```

Tab
キー

条件に合わなかった時に実行される内容

条件の種類

コードの書き方	意味	例
<code>a == b</code>	aとbは等しい	<code>a == 4, s == "ryo"</code>
<code>a < b</code>	aはbより小さい	<code>a < 0.5</code>
<code>a > b</code>	aはbより大きい	<code>a > 0.5</code>
<code>a != b</code>	aはbと等しくない	<code>a != 0</code>
<code>a <= b</code>	aはb以下	<code>a <= 0.5</code>
<code>a >= b</code>	aはb以上	<code>a >= 0.5</code>

条件の注意

「=」：イコールが1つの時は代入

「==」：イコールが2つの時は「等しい」という条件

✕ if (a = 3): エラー

○ if (a == 3): aが3と等しい時にインデントした命令が実行される

自由課題2・3 コイントス

[2] コイントスをした結果を表示するプログラムを作成してください。

- `x = random.random()` を条件にうまく使ってください
 - 0.5 未満なら 表 (Head)
 - 0.5 以上なら 裏 (Tail) など

コードを実行すると、`x`という変数に0~1までのランダムな数字が入ることを使う

[3] それもできたら、`input()`で入力として H, T を受け取り、勝負を仕掛けてくるプログラムを作ってください。

プログラムは、ユーザが選ばなかったほうを選びます。

自由課題4 ずるくないじゃんけんゲーム

- さっき作ったコードを改良して、ランダムにじゃんけんの手を出すじゃんけんゲームを作成してください。
- `x = random.random()` を条件にうまく使ってください

今日の内容

- 繰り返し文(教科書 p.111 – p.131)
 - 同じ処理を何度も繰り返す場合を書く命令
 - 条件を満たす間繰り返す (while文)
 - リストから要素を取り出しながら繰り返す (for文)

while文

while (条件):

イン デ ン ト	命令文1
	命令文2

カッコ内で指定した条件を満たす間
インデントした命令文を繰り返す

【復習】 条件の種類

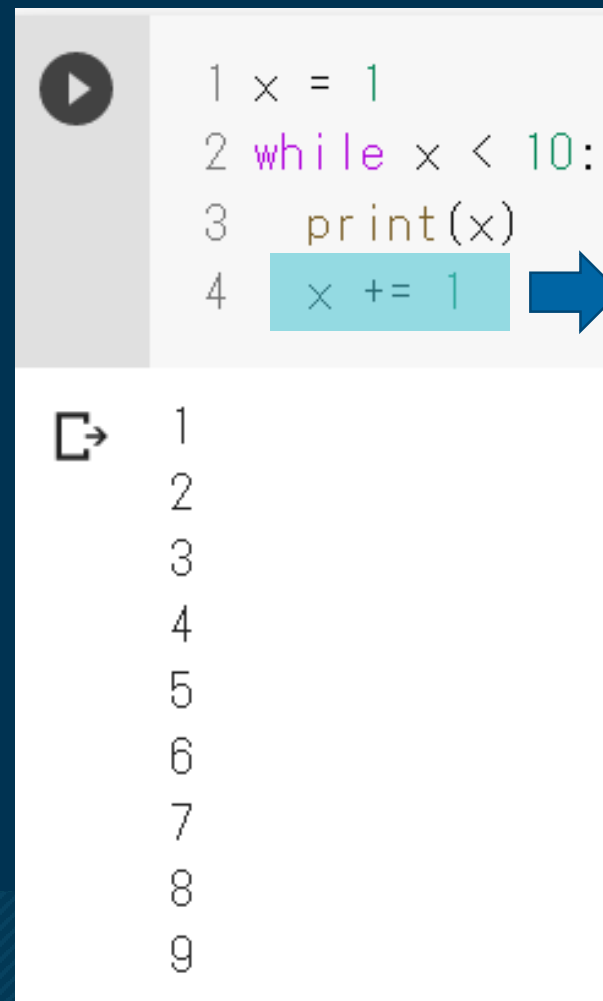
コードの書き方	意味	例
<code>a == b</code>	aとbは等しい	<code>a == 4, s == "ryo"</code>
<code>a < b</code>	aはbより小さい	<code>a < 0.5</code>
<code>a > b</code>	aはbより大きい	<code>a > 0.5</code>
<code>a != b</code>	aはbと等しくない	<code>a != 0</code>
<code>a <= b</code>	aはb以下	<code>a <= 0.5</code>
<code>a >= b</code>	aはb以上	<code>a >= 0.5</code>

例1

$x < 10$ を満たす限り, 3-4行目が繰り返される

条件を満たすたびに $x += 1$ が実行

⇒ 1 ~ 9までが出力される



```
1 x = 1
2 while x < 10:
3     print(x)
4     x += 1
```

The code is executed, and the output is displayed as a list of numbers from 1 to 9. A blue arrow points from the `x += 1` line in the code to the output, indicating the increment operation.

1
2
3
4
5
6
7
8
9

point (p.71):
複合代入演算子
 $x = x + 1$ と同じ

例2

- 2の倍数を出力するプログラム
 - xが20になったら, (x < 20)を満たさないので終了
 - x += 2は, 2ずつ増やしていく命令 (p.71)

```
1 x = 2
2
3 while x < 20:
4     print(x)
5     x += 2
```

2
4
6
8
10
12
14
16
18

例3 while文とif文の組み合わせ

while文の中にif文を組み合わせて使うことができる

3回 0-1の間でランダムな数を振って、0.7以上が出れば「チャレンジ成功」と出力するプログラム

tは繰り返しの回数を表す

if 文, else文で実行したい命令は, if, elseの列よりも1回分インデントする.



```
1 import random
2
3 t = 0
4 while (t < 3):
5     x = random.random()
6     if (x < 0.7):
7         print("チャレンジ失敗")
8     else:
9         print("チャレンジ成功")
10
11     t += 1
```



```
チャレンジ失敗
チャレンジ失敗
チャレンジ成功
```

for文

書き方

for 変数 in リスト:

イン
デ
ン
ト

命令1

命令2

... ..


... ..

リストの長さだけ黄色の四角
の中を繰り返す

リストの中の数が、先頭から順に
変数の中に入る

イメージ

①




```
for x in [0, 1, 2, 3, 4]:  
    x *= 2  
    print(x)
```

xの中にリストの先頭を入れる

for下のインデント部分を実行する

②




```
for x in [0, 1, 2, 3, 4]:  
    x *= 2  
    print(x)
```

xの中にリストの次の要素を入れる

for下のインデント部分を実行する


イメージ

③




```
for x in [0, 1, 2, 3, 4]:  
    x *= 2  
    print(x)
```

④



```
for x in [0, 1, 2, 3, 4]:  
    x *= 2  
    print(x)
```

⑤



```
for x in [0, 1, 2, 3, 4]:  
    x *= 2  
    print(x)
```

すべてのリストの要素をたどったので
終了する.

実行例

リストを用いる例



```
1 for x in [0, 1, 2, 3, 4]:  
2     x *= 2  
3     print(x)
```



```
0  
2  
4  
6  
8
```

等差数列を使う場合の例



```
1 for x in range(5):  
2     x *= 2  
3     print(x)
```



```
0  
2  
4  
6  
8
```

for文の書き方の工夫

```
for x in [0, 1, 2, ... .. ,100]: # 書くのが大変
```



```
for x in range(100): # range(100): = [0, 1, 2, ... .. ,99]
```

真理値の利用

True (1): 真
条件が真であることを示す値

False (0): 偽
条件が偽であることを示す値



```
1 while True:  
2     print("Hello")  
3  
4 while 1:  
5     print("Hello")
```

無限ループ



```
1 while False:  
2     print("Hello")  
3  
4 while 0:  
5     print("Hello")
```

実行されない

break / continue 文

- 繰り返し文の中で用いられる特殊な命令
 - break 文
 - この文にたどり着いたら、繰り返しを中断して、繰り返しから抜ける。
 - continue 文
 - この文にたどり着いたら、今の繰り返しを中断して、次の繰り返しにうつる。

break 文の例

本来なら、 t が10になるまで繰り返される。
($t < 10$)

チャレンジ成功した場合、
break文にたどり着くので、そこで繰り返しが強制終了される。

- 他の事例
 - 成功したらゲームを次に進める
 - パスワードがあっていたらログインさせる (3回も入力させない)
 - 体力がなくなったらGame Overと表示して繰り返しを終わらせる

```
1 import random
2
3 t = 0
4 while (t < 10):
5     x = random.random()
6     if (x < 0.7):
7         print("チャレンジ失敗")
8     else:
9         print("チャレンジ成功")
10        break
11
12 t += 1
13
14 print("Finish.")
```

→ チャレンジ失敗
チャレンジ失敗
チャレンジ失敗
チャレンジ失敗
チャレンジ成功
Finish.

continue 文の例

```
1 import random
2 t = 0
3 HP = 1
4 while (t < 5):
5     t += 1
6     print("現在のHP:", HP)
7     x = random.random()
8     if (x < 0.3):
9         print("チャレンジ失敗")
10        continue
11
12    print("チャレンジ成功, HPを3回復します.")
13    HP += 3
```

現在のHP: 1

チャレンジ成功, HPを3回復します.
現在のHP: 4

チャレンジ失敗
現在のHP: 4

<= HPも3増えていない

チャレンジ成功, HPを3回復します.
現在のHP: 7

チャレンジ成功, HPを3回復します.
現在のHP: 10

チャレンジ失敗

チャレンジ失敗したら, continue文にたどり着き,
これ以上先のwhile文内の命令(12, 13行目)
を実行せずに, 条件文(4行目)へ戻る.

【やや難&重要】繰り返し文のネスト, ネストされた繰り返し文

- 繰り返し文の中に, さらに繰り返しを書くこと



```
1 for i in range(3):  
2     print("i =", i)  
3     for j in range(2):  
4         print("    j =", j)  
5
```

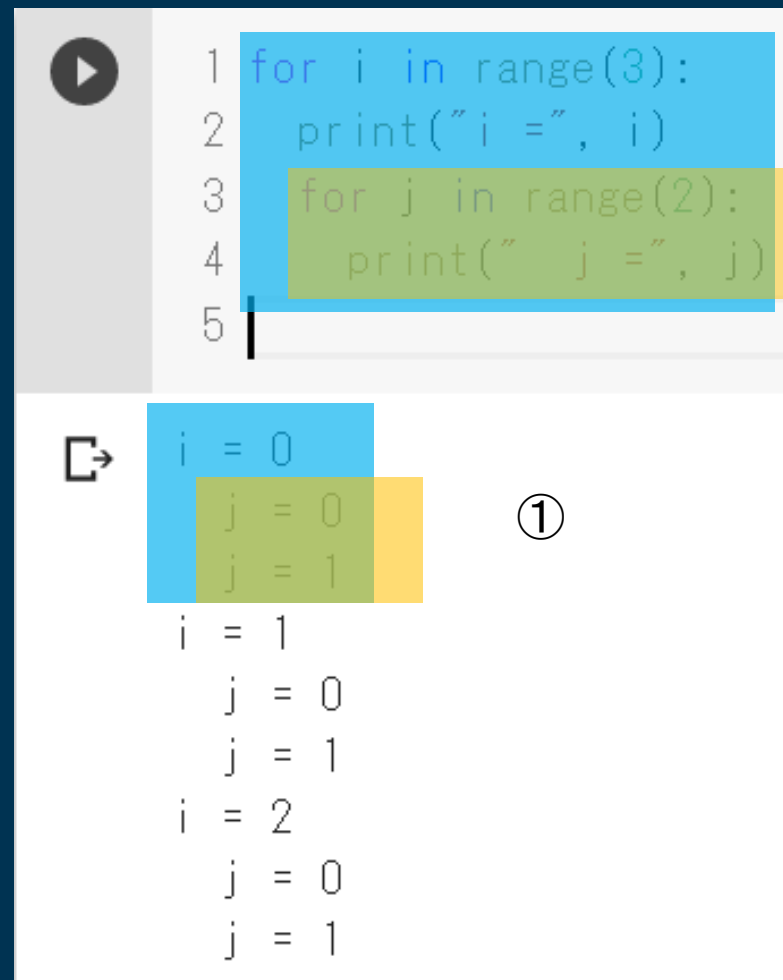


```
i = 0  
    j = 0  
    j = 1  
i = 1  
    j = 0  
    j = 1  
i = 2  
    j = 0  
    j = 1
```

【やや難&重要】繰り返し文のネスト, ネストされた繰り返し文

- 繰り返し文の中に, さらに繰り返しを書くこと

①: $i=0$ が代入された状態で,
 $j=0, j=1$ の繰り返しが行われる



The image shows a Python code editor with a play button icon on the left. The code is as follows:

```
1 for i in range(3):
2     print("i =", i)
3     for j in range(2):
4         print("    j =", j)
```

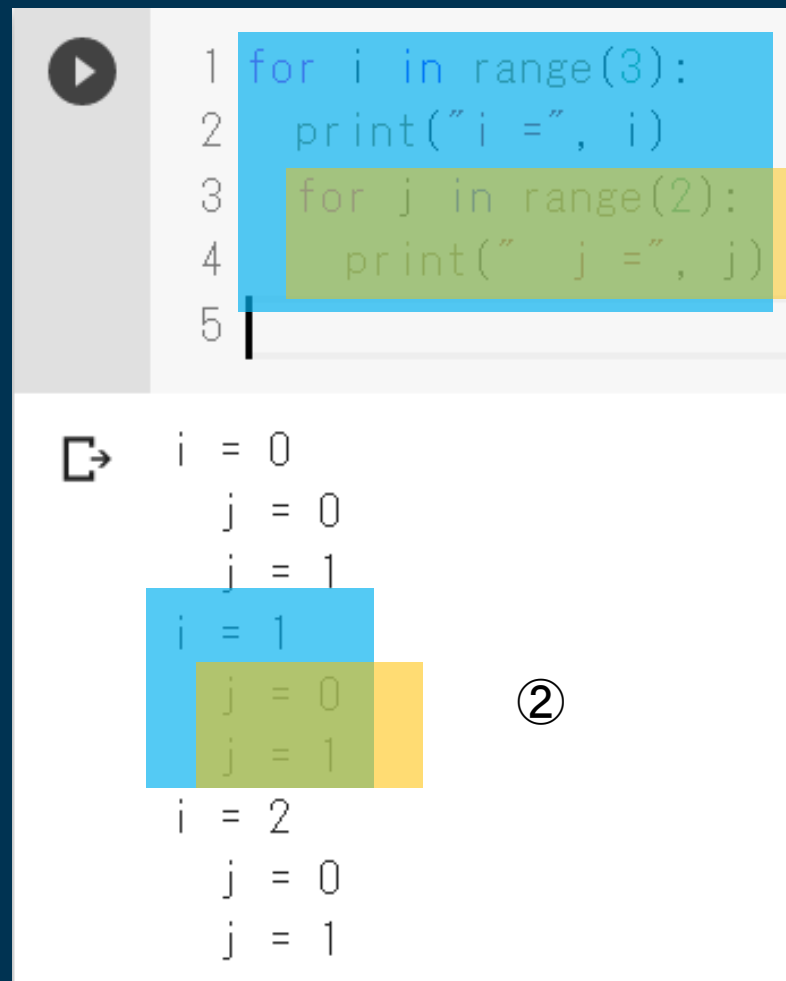
Below the code, the execution output is shown. It starts with a copy icon and a circled 1. The output is:

```
i = 0
    j = 0
    j = 1
i = 1
    j = 0
    j = 1
i = 2
    j = 0
    j = 1
```


【やや難&重要】繰り返し文のネスト, ネストされた繰り返し文

- 繰り返し文の中に, さらに繰り返しを書くこと

②: $i=1$ が代入された状態で,
 $j=0, j=1$ の繰り返しが行われる



The image shows a Python code editor with a play button icon on the left. The code is as follows:

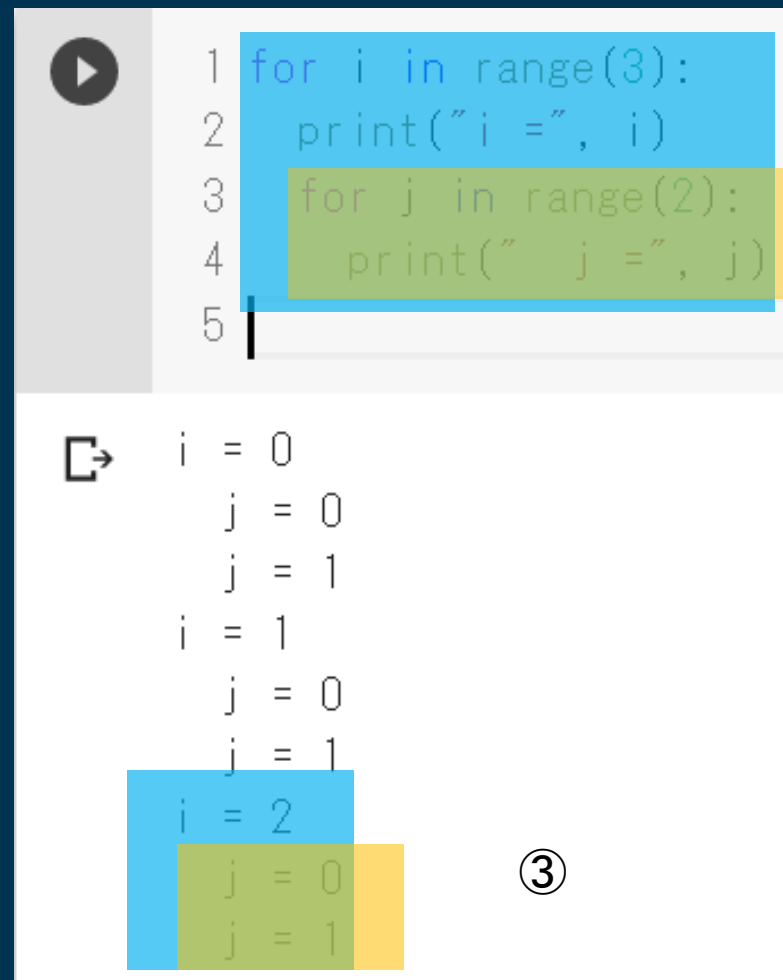
```
1 for i in range(3):  
2     print("i =", i)  
3     for j in range(2):  
4         print("    j =", j)  
5
```

Below the code, a diagram illustrates the execution state at the second iteration of the outer loop (when $i=1$). A blue box highlights the state where $i=1$ is assigned. Inside this box, a green box highlights the state where $j=0$ and $j=1$ are assigned. To the right of the green box is a circled number 2, indicating the second iteration of the inner loop. Below the green box, the state for the third iteration of the outer loop is shown: $i=2$, $j=0$, and $j=1$.

【やや難&重要】繰り返し文のネスト, ネストされた繰り返し文

- 繰り返し文の中に, さらに繰り返しを書くこと

③: $i=2$ が代入された状態で,
 $j=0, j=1$ の繰り返しが行われる



The image shows a Python code editor with a play button icon on the left. The code is as follows:

```
1 for i in range(3):
2     print("i =", i)
3     for j in range(2):
4         print("    j =", j)
```

Below the code, the execution state is shown. It lists the values of i and j for each iteration. The first two iterations of the outer loop are shown in grey, and the third iteration is highlighted in blue. The inner loop iterations are shown in green. The current state is labeled with a circled 3.

```
↳ i = 0
   j = 0
   j = 1
i = 1
   j = 0
   j = 1
i = 2
   j = 0
   j = 1
```

③

自由課題 1

3回コイントスを行って、各回で、表(H)、裏(T) を出力するプログラムを書いてください。

条件

- 表と裏の出る確率は等しい
- `random.random()` を使用してください
- 先週書いたプログラムをそのまま使ってもOK

自由課題 2

- TRPGというゲームでは、N回分、1～Mまでの整数値を出力してプレイヤーのダメージを計算したり、魔法の効果を判定することがあります。
- N, Mの値を受け取って、N回分、1～Mまでの整数値を出力するプログラムを作ってください (効果などは実装しなくてOK).
 - 整数値のランダムな数字は、`random.randint(最小, 最大)` で出力します。



```
1 N = input("さいころを振る回数 (N): ")
2 M = input("さいころの最大値 (M): ")
3
4 # 何らかの繰り返しの処理
```

自由課題 3

p.23-26の原理を利用して, 九九を表示するプログラムを書いてください.

$$1 \times 1 = 1$$

$$1 \times 2 = 2$$

...

...

$$9 \times 8 = 72$$

$$9 \times 9 = 81$$

簡単すぎる場合は, 九九九(3つの数の積を $9 \times 9 \times 9$ まで表示するプログラム)を書いてください.

練習したい方のための課題

- **ABC 089 B - Hina Arare**

- https://atcoder.jp/contests/abc089/tasks/abc089_b

- **ABC 073 B – Theater**

- https://atcoder.jp/contests/abc073/tasks/abc073_b

- **ABC 076 B - Addition and Multiplication**

- https://atcoder.jp/contests/abc076/tasks/abc076_b

本日の内容

- 繰り返し文
 - while文: 条件を満たす限り繰り返す
 - for 文: リストの長さだけ繰り返す
 - 中断
 - break 文: 中断して繰り返しから抜ける
 - continue 文: 中断して繰り返しの先頭に戻る
 - 繰り返し文のネスト, ネストされた繰り返し文
 - くりかえしの中にさらにくりかえしを書くこと

次回の内容

- リストの応用
 - リスト内包表記
 - split()機能の利用
 - input()で受け取った文字列のリスト化
- 辞書
 - 辞書とは
 - 辞書を利用したプログラム
- (余力があれば) 集合