条件分岐、繰り返し2

初めてのプログラミング

2020年度

只木進一（理工学部）

繰り返しの書き方

- whileは無限ループになる危険性がある
- 予め繰り返し回数が定まっていることが多い
  - for文を使う

for
範囲内を繰り返す

- 指定回数繰り返す
  - for 変数 in 範囲
  - for x in range (n)
  - I を0からn-1まで変化させながら繰り返す

```
sum = 0
for i in range(11):
    sum += i
print(sum)
```

Control/for0.ipynb

for
範囲内を繰り返す

- 指定回数繰り返す
  - range (開始、終了、ステップ)

```
for x in range(0,n,2) : #0からn-1まで、一つおきに繰り返す
    処理
```

Control/for0.ipynb

for
範囲内を繰り返す

- リストなどの要素で繰り返し範囲を指定

```
colors = [‘red’, ‘green’. ‘blue’]
for c in colors:
    print(colors)
```

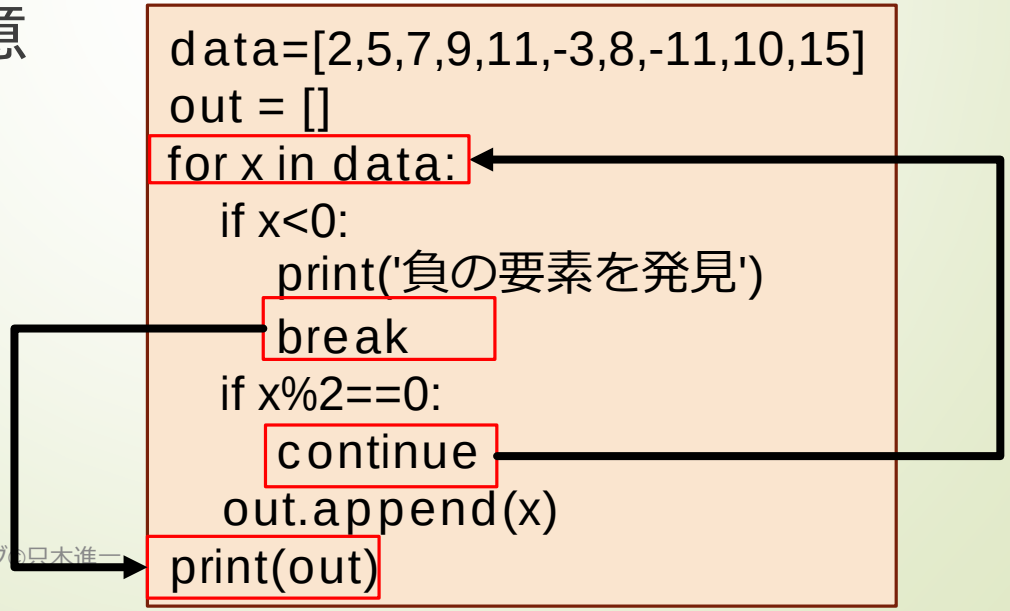
for文のネスティング(nesting)

- for文の処理の中にfor文を書くことで、多重ループを作ることができる。

```
for i in range(3):
    print (f'i={i}')
    for j in range(4):
        print(f'({i},{j})')
```

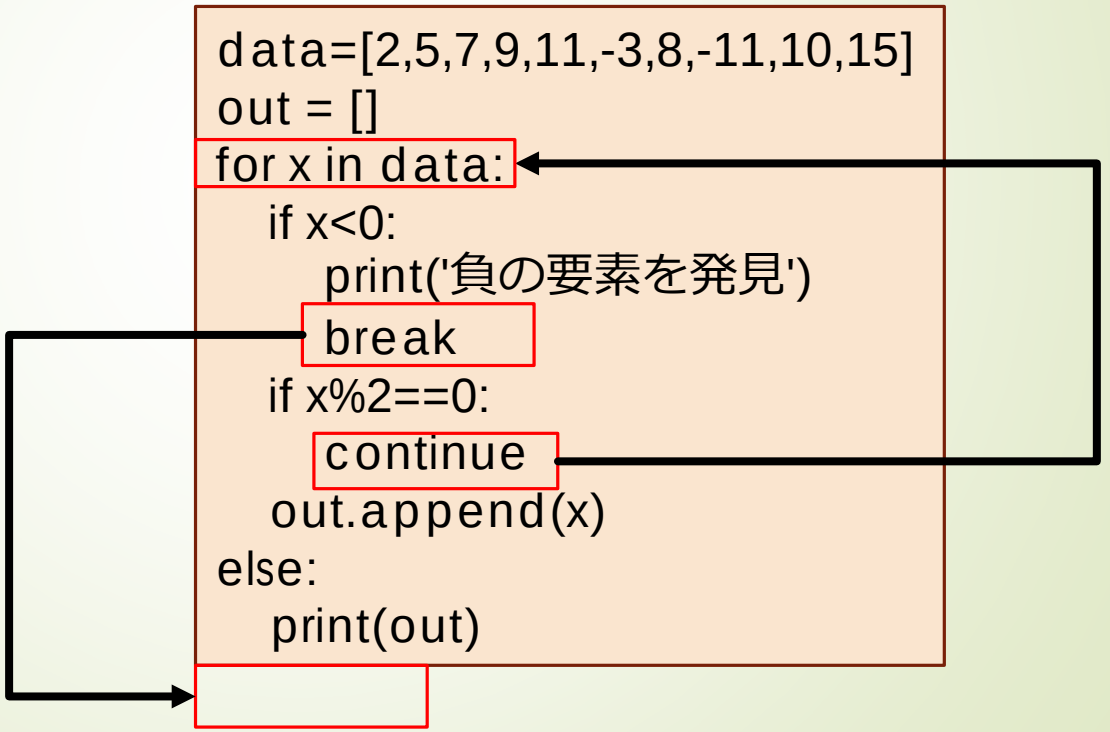
breakとcontinue

- break: forループから抜ける
- continue: forループの先頭へ
  - ループのインデクスが一つ増えることに注意



for-else構文

- forループが終了するとelse部分を実行



例外処理

- 実行中にエラーが発生すると、プログラムの実行は、そこで停止。
- 予期しないデータ、ディスクの状態、ネットワークの状態で、エラーが発生する可能性
- 予め、エラー発生を予測し、エラー発生時の対応を記述しておくことで、その後の処理を継続

例外処理

```
try:
    エラー発生可能性のある処理
except:
    エラー時の処理
```

- エラーの種類毎に振り分け

```
try:
    エラー発生可能性のある処理
except 例外1:
    例外1時の処理
except 例外2:
    例外2時の処理
```

次回

- 6章「リスト」