システムアーキテクト演習
7週目

吉田 則裕

スライスによるアクセス

- スライス
  - 要素番号を複数指定して，その範囲に対応する要素を一括して取得できる.
- 以下のように，開始位置と終了位置を整数値で指定する.

seq[開始位置:終了位置]

- 開始位置で指定した要素から終了位置で指定した一つ手前の要素からなるシーケンスを返す.
  - 存在しない要素番号を指定してもエラーは返さずに，指定された中で実際に存在する範囲の要素からなるシーケンスを返す.

```
S = 'Ritsumeikan'
print(S[-100:100])      Ritsumeikan
print(S[-3:-1])         ka
print(S[2:10])          tsumeika
```

スライスによるアクセス（続き）

- 以下のように，取得する要素番号の増分を指定できる.

  seq[開始位置：終了位置:増分値]

  - 例：S[0:100:2]
    - 先頭要素，3 番目，5 番目，...，99 番目の要素からなるシーケンスを取得できる.

```
S = 'Ritsumeikan'
print(S[0:100:2])
```

Rtuekn

存在しない要素番号を指定してもエラーは返さずに，
実際に存在する範囲の要素からなるシーケンスを返す.

バイト列シーケンス

- 画像や音楽データなど，バイナリデータのためのデータ型
- バイト列は0～255 の整数値のシーケンス
- 変更できない.
- シングルクォーテーションマークかダブルクォーテーションマークで囲った先頭にB またはb の1 文字を付加して記述する.

```
B = b'abcdef1234'
```

バイト列シーケンス（続き）

- バイト列から1 要素を参照すると，文字列の場合とは異なり，そのASCII コードでの値が返される.
  - 英字のa に対しては，そのASCII コードである97が返される.
  - 数字の4 に対しては，52が返される.

```
B = b'abcdef1234'
print(B[0])          97
print(B[-1])         52
print(B[1:6])        b'bcdef'
print(B[-100:-1])    b'abcdef123'
print(B[-100:-99])   b''
```

先頭のbはバイト列であることを示している

‘’は空であることを示している.

演習9，演習10

- 演習9
  - 演習8で作成したリスト型シーケンスLに対して，最後から4番目の要素と最後から2番目の要素までの3つの要素を負の整数で指定したスライスによって参照してみよう.

- 演習10
  - バイト列’1234abcde’ からなるシーケンスBを定義し，先頭要素を参照してみよう.
  - 5番目の要素から最終要素までをスライスによって参照してみよう.
  - 奇数番目の要素だけをスライスと増分によって参照してみよう.

演習9，10の解答例

```
L = [0,1,2,3,4,5,6,7,8,9]
print(L[-4:-1])
```

```
[6, 7, 8]
```

```
B = b'1234abcde'
print(B[0])
print(B[4:9])
print(B[1:9:2])
```

```
49
b'abcde'
b'24bd'
```

シーケンス要素への代入

- リスト型のような変更可能なシーケンスには，代入文により要素を設定できる.
  - 要素番号で指定する場合は，参照の場合と同じように負の値によって末尾からの位置指定もできる.
  - スライスで指定する場合は，代入文の右辺で，指定した範囲を置き換えることができる.
    - 増分も指定できる.
    - 置換前と置換後の要素数が異なっても構わない.
    - 右辺が長さ0 の場合，元のリストから指定された範囲の要素が削除される.

シーケンス要素への代入の例

```
L = [0,1,2,3,4,5,6,7,8,9]
L[2] = 'abc'
print(L)                    [0, 1, 'abc', 3, 4, 5, 6, 7, 8, 9]
L[-2] = 'def'
print(L)                    [0, 1, 'abc', 3, 4, 5, 6, 7, 'def', 9]
L[3:5] = 'ghi'
print(L)                    [0, 1, 'abc', 'g', 'h', 'i', 5, 6, 7, 'def', 9]
L[3:5] = ['ghi','jkl']
print(L)                    [0, 1, 'abc', 'ghi', 'jkl', 'i', 5, 6, 7, 'def', 9]
L[3:6] = ['mno','pqr']
print(L)                    [0, 1, 'abc', 'mno', 'pqr', 5, 6, 7, 'def', 9]
L[3:6] = []
print(L)                    [0, 1, 'abc', 6, 7, 'def', 9]
L[:] = []
print(L)                    []
```

スライスによる範囲指定を使った代入が可能
指定した範囲より右辺の要素数が多ければ，
余った分（この例では）は挿入される

指定した範囲より，右辺の要素数が少なければ，
不足分の要素が削除される

右辺が[]の場合は，指定範囲の要素が削除される

範囲指定を[:]とし，右辺を[]とすると，全要素が削除される．

演習11

- 演習8で定義したリスト型シーケンスLの3番目の要素を，代入文によって文字列’abc’ に設定してみよう.

演習11の解答例

- 演習8で定義したリスト型シーケンスLの3番目の要素を，代入文によって文字列’abc’ に設定してみよう.

```
L = [0,1,2,3,4,5,6,7,8,9]
L[2] = 'abc'
print(L)
```

シーケンス要素の削除

- del文を使っても削除できる.
  - スライスも指定
  - スライスの場合は，増分も指定可能

```
L = [0,1,2,3,4,5,6,7,8,9]
del L[1], L[3]
print(L)            [0, 2, 4, 5, 6, 7, 8, 9]
del L[2:4]
print(L)            [0, 2, 6, 7, 8, 9]
del L[-3:-1]
print(L)            [0, 2, 6, 9]
del L[:]
print(L)            []
```

範囲指定を[:]とすると，
全要素が削除される.

```
L = [0,1,2,3,4,5,6,7,8,9]
del L[0:11:3]
print(L)            [1, 2, 4, 5, 7, 8]
del L[0:100]
print(L)            []
```

演習12

- 演習11で用いたリスト型シーケンスL の4番目の要素を削除しなさい.
- 次に全要素を削除しなさい.

演習12の解答例

```
L = [0,1,2,3,4,5,6,7,8,9]
del L[3]                    [0, 1, 2, 4, 5, 6, 7, 8, 9]
print(L)
del L[:]                    []
print(L)
```

シーケンスの演算

演算	意味
+	シーケンスの連結
*	シーケンスを指定回数繰り返した新しいシーケンスの生成
比較演算 (<, >, <=, >=, ==, !=)	シーケンスの要素を先頭から順に比較し，TrueかFalseを返す
in	シーケンスに指定した要素が含まれるかどうかを判定
not in	シーケンスに指定した要素が含まれないかどうかを判定
len()	シーケンスの要素数（長さ）を返す

シーケンスの演算例

```
S1 = 'Ritsumeikan '
S2 = 'University'
print(S1 + S2)
print(S1 * 2)
print(2 * S2)
print(S1 == S2)
print('t' in S1)
print('r' in S1)
print('R' not in S2)
print(len(S1))
print(len(S1*3))
```

```
Ritsumeikan University
Ritsumeikan Ritsumeikan
UniversityUniversity
False
True
False
True
12
36
```

演習13

1. リスト型シーケンスL1, L2 をそれぞれ，[1, 2, 3] と[4, 5, 6] に設定し，2つを連結してみよ.
2. L1 の要素を10 回繰り返したリストL3 を作成せよ.
3. L1 とL2 が等しいかどうかを比較演算子! = によって確認せよ.
4. 数値1 がL1 に含まれることを演算子in によって確認せよ．また数値1 がL2 に含まれないことを演算子not in によって確認せよ.
5. L3 の長さを組み込み関数len() によって求めよ.

演習13の解答例

```
L1 = [1,2,3]
L2 = [4,5,6]
print(L1 + L2)
L3 = L1 * 10
print(L3)
print(L1 != L2)
print(1 in L1)
print(1 not in L2)
print(len(L3))
```

```
[1, 2, 3, 4, 5, 6]
[1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3]
True
True
True
30
```

for文

for 文を使うと，シーケンスの個々の要素に順にアクセスして，それぞれについて同じ処理を行うことができる.

- for 文の1 行目には，処理対象となるシーケンスとシーケンス要素を代入する変数を指定し，行末に”:”（コロン）を指定する.
- 2 行目はタブを使って字下げした後に各要素に対して繰り返して行う処理に対応した実行文を指定する.
- else ブロックがある場合は，ループ終了時に実行されるがbreak によってループを終了した場合はelse ブロックは実行されない.

```
for <target> in <sequence>:
    <statements>
    if <test>: break
else:
    <statements>
```

for文の例

```
for i in [1,2,3,4]:
    print(i)
    if i > 2: break
else:
    print(i*10)
```

```
1
2
3
```

breakが実行されるため，elseブロックは実行されない．

```
for i in [1,2,3,4]:
    print(i)
    if i > 10: break
else:
    print(i*10)
```

```
1
2
3
4
40
```

breakが実行されないため，elseブロックが実行される

```
for i1, i2 in [("Tokyo",1), ("Kyoto",2),("Hakata",3)]:
    print(i2,i1)
```

```
1 Tokyo
2 Kyoto
3 Hakata
```

- リストの要素が文字列と数値からなるタプルになっている
- タプルの各要素をi1とi2にそれぞれ代入している．

演習14

1 から5 までの5 個の整数値を要素とするリストに対して，for 文を用いて全要素の足し算と掛け算を実行するプログラムを記述してみよ.

演習14の解答例

```
a = 0
b = 1
for i in [1,2,3,4,5]:
    a = a + i
    b = b * i
else:
    print(a)
    print(b)
```

15
120

簡単なファイルの入出力

- ファイルの読み書きには組み込み関数open() を用いる.
  - open は第1 引数として，操作するファイル名を，第2 引数として，ファイルの操作モードを指定する.
    - 読み込み専用モードは’r’
    - 読み書きモードは’r+’
    - 書き込みモードは’w’
    - 追加で書き込むモードは’a’
    - モードを指定しない場合は’r’ と見なされる.

```
fout = open('example.txt','w')
```

- ’example.txt’ というファイル名を書き込み用として設定
- すでに同じファイル名のファイルが存在する場合は，内容は消えてしまう.

```
fout.write('Hello')
```

文字列’Hello’ を’example.txt’ というファイルに書き出す.

```
fout.close()
```

ファイル操作が終了した時点でclose関数により，ファイルをクローズする.

簡単なファイルの入出力（続き）

さきほどのファイルから書き出したデータを読み込んでみよう.

```
fin = open('example.txt')
read_str = fin.readline()
print(read_str)
fin.close()
```

- readline() メソッドはファイルから1 行を読み込む.
- これにより，変数read_str に書き込んだ文字列が設定される.

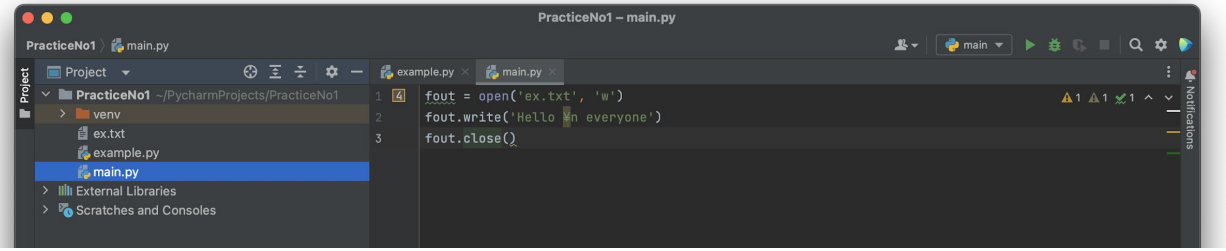
print() 関数によって変数の値を出力し，読み込んだ文字列が表示される

ファイル入出力の例

```
fout = open('ex.txt', 'w')
fout.write('Hello ¥n everyone')
fout.close()
```

改行文字である¥nを含めた2行を一度に書き込んでいる.

ex.txtがプロジェクトビューに追加されている




```
fin = open('ex.txt','r')
strs = fin.readlines()
print(strs)
print(strs[0],strs[1])
print(strs[0].strip(),strs[1].strip())
fin.close()
```

2行を一度に書き込んだため，複数行１度に読み込むreadline**s**関数を使う

strip関数は，文字列の前後にある余分な半角文字や制御文字を取り除く

演習15

- ‘ex15.txt’という名前のファイルを書き込み用として設定し，文字列’Ritsumeikan University’を書き込み，いったんclose しなさい.
- 書き込んだファイルから文字列を読み込み，print()関数によって読み込んだ文字列が表示されることを確認しなさい.

演習15の解答例

```
fout = open('ex15.txt', 'w')
fout.write('Ritsumeikan University')
fout.close()

fin = open('ex15.txt','r')
str = fin.readline()
print(str)
fin.close()
```

Ritsumeikan University

ライブラリ（モジュール）

- 数学用ライブラリmath
  - 三角関数，指数関数，対数関数，円周率など
- 行列演算Numpy
- 疑似乱数生成random
- OS の機能を扱うos

数学用ライブラリmathの使用例

```
import math

print(math.pi)
print(math.sin(math.pi/4))
print(math.sin(math.pi/4) ** 2)
print(math.log(2,10))
print(math.gcd(12,54))
```

mathライブラリの使用を宣言

円周率π

sin(π/4)

sin(π/4)の2乗

常用対数log2

12と54の最大公約数

その他

- Python には他にもいろいろなデータ型や多くの関数が標準で用意されている．また種々のライブラリもimport することで活用できる．
- この演習では時間の制約もあり，ごく一部しか紹介しなかったが，図書やネット上の資料で自学自習してほしい．
- 何か質問があれば学びラボ等を活用すること．

来週からの課題（予告）

来週から，グループでPython によるプログラムを開発する.

- 各グループでプログラムによって解決できる課題を見つけ，課題解決のためのPython プログラムを開発し，正しく動作することを確認すること.
- レポートを提出する必要がある.
- 課題解決の難易度や，プログラムで創意工夫した点を評価する.