みんなの Python
- 2. 変数と組み込み型 -

新納浩幸

変数とリテラル

変数に関する説明は略、、、
型の宣言が必要ないことだけがポイント

変数ではなく、データそのもののことをリテラルという

Python ではオブジェクト

組み込みデータ型

数値： 1, 0.5, 0xFF, ・・・
文字列：”dog”, ‘Ibaraki Univ.’, ”日立市”
リスト： [1, ‘2’, [3,4], [1, ”cat”]]
タプル：(1, 2, 3)
辞書：{“dog”:”犬”, ”cat”:”猫” }

数値の演算

＋、－、＊、／　は通常通り

** 累乗　2 ** 3 → 8

% 剰余　11 % 4 → 3

注意）

整数どうしの割り算は整数

5 / 2 → 2

5.0 / 2 → 2.5

小数点以下の切り捨ては //

5.0 // 2 → 2.0

文字列の定義

S1 = “I love Hitachi”

文字列内に二重引用符がある場合

S2 = ‘You said “I love Hitachi”’

文字列内に改行がある場合

S3 = ”””<body>
My name is Shinnou.
</body>
”””

文字列の連結

S1 = “1234”
S2 = “5678”
S1 + S2 → “12345678”

S1 = “1234”
S1 * 3 → “123412341234”

文字列の数値への変換

int (“1234”) → 1234

float (“12.34”) → 12.34

文字列の長さ

len (“Hitachi”) → 7

len (“日立”) → 4

上記を2にするためには unicode を利用する

len (u“日立”) → 2

文字列内の文字列

（文字列） in （文字列）→ True/False

‘ac’ in “Hitachi” → True
‘b’ in “Hitachi” → False

日本語は Unicode にした方が無難

u‘日立’ in u“茨城県日立市” → True

文字列の置換

文字列 . replace（検索文字列，置換文字列）

str = “We lives in Hitachi”
str.replace(“Hitachi”, “Mito”) → ’We lives in Mito’
str は変更されないことに注意

str = u“私たちは日立に住む”
str.replace(u“日立”, u“水戸”)
→ u'¥u79c1¥u305f¥u3061¥u306f¥u6c34¥u6238¥u306b¥u4f4f¥u3080‘
str2 = str.replace(u“日立”, u“水戸”)
print(str2.encode(‘cp932’))

文字列の Index

S = “12345”

S[2] = ‘3’

* Index は 0 から
* 取り出されるのは文字ではなく文字列

リスト

使い勝手がよい、何でも入れられる、
文字列と扱い方は似ている
Index は 0 から

a = [1, 2, 3, 4, 5]
a[2] → 3

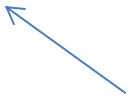
a[-1] → 5

Index の -1 は最後の要素を指す

スライスによるリスト要素の取り出し

a = [1, 2, 3, 4, 5]

a[1 : 3] → [2,3]



1 以上 3 未満

a[2 :] → [3,4,5]　　2 以上最後まで

a[: 3] → [1,2,3]　　最初から 3 未満

リストの足し算、かけ算

a = [1, 2, 3, 4, 5]
b = [10, 20, 30]
a + b → [1,2,3,4,5,10,20,30]

b * 3 → [10,20,30,10,20,30,10,20,30]

リストの Tips

リストの要素の追加は足し算を利用

a = [1, 2, 3, 4, 5]
a + [6] → [1,2,3,4,5,6]

長さを指定したリストの作成はかけ算を利用

a = [0] * 6
a → [0,0,0,0,0,0]

要素の置き換えと削除

a = [1, 2, 3, 4, 5]
a[2] = “200”
a → [1,2, “200”, 4,5]

a = [1, 2, 3, 4, 5]
del a[2]
a → [1,2, 4,5]

要素の検索

names = [‘Gray’, ‘Chick’, ‘Pat’, ‘Roy’, ‘Dave’]
‘Gray’ in names → True
‘Peter’ in names → False

names.index(‘Chick’) → 1

names.index(‘Hiro’) → エラーになる

in で最初に要素があることを確認して使う

リスト要素の最大値と最小値

a = [15, -2, 23, 14, 5]

max(a) → 23

min(a) → -2

リストの要素は数値でないと無意味

リストで与える必要はない

max(15, -2, 23, 14, 5) → 23

min(15, -2, 23, 14, 5) → -2

リストのソートと反転

a = [15, -2, 23, 14, 5]

a.sort()

a → [-2, 5, 14, 15, 23]

小さい順、もとのリストが書き換わる

a = [15, -2, 23, 14, 5]

a.reverse()

a → [5, 14, 23, -2, 15]

反転、もとのリストが書き換わる

append メソッド

a = [1,2,3,4,5]

a.append(10)

a → [1,2,3,4,5,10]

最後に追加、もとのリストが書き換わる

以下の形に注意

a.append([6,7])

a → [1,2,3,4,5, [6, 7]]

expand メソッド

a = [1,2,3,4,5]

a.expand([10,20,30])

a → [1,2,3,4,5,10, 20, 30]

足し算と同じ感じ
expand の引数はリストでないとダメ
もとのリストが書き換わる

多次元配列

リストのリストで表現する

```
a = [ [  1,  2,   3,  4,  5],
      [  6,  7,   8,  9,10],
      [11,12,13,14,15] ]
```

3 X 5 の行列

a[1][2] → 8

a[2] → [11,12,13,14,15]

辞書

連想配列（ハッシュ）のこと

dic = { “dog”:”犬”, “cat”:”猫”, “caw”: “牛 ” }
dic[‘dog’] → ‘犬’

辞書の要素の追加

dic = { “dog”:”犬”, “cat”:”猫”, “caw”: “牛 ” }
dic[‘horse’] = “馬”
dic → { “dog”:”犬”, “cat”:”猫”,
“caw”: “牛 ”, “horse”:”馬” }

既存のキーで追加すると、
上書きされる

要素の削除

dic = { “dog”:”犬”, “cat”:”猫”, “caw”: “牛 ” }

del dic[‘cat’]

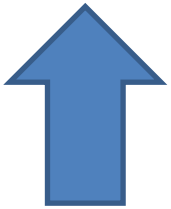
dic → { “dog”:”犬”, “caw”: “牛 ” }

キーの一覧確認

dic = { “dog”:”犬”, “cat”:”猫”, “caw”: “牛 ” }

dic.keys()

→ dict_keys([‘dog’, ‘cat’, ‘caw’])



リストではない、イテレータとして使う

キーの存在の確認

dic = { “dog”:”犬”, “cat”:”猫”, “caw”: “牛 ” }

‘cat’ in dic → True

‘horse’ in dic → False

タプル

リストと似ている
1回作ると書き換えができない点が違う

a = (1, 2, 3, 4, 5)
a[1] → 2
a[2:4] → (3, 4)　スライスで取り出してもタプル

a[2] = 10 ← エラーになる
タプルの要素は変更できない

タプルの足し算

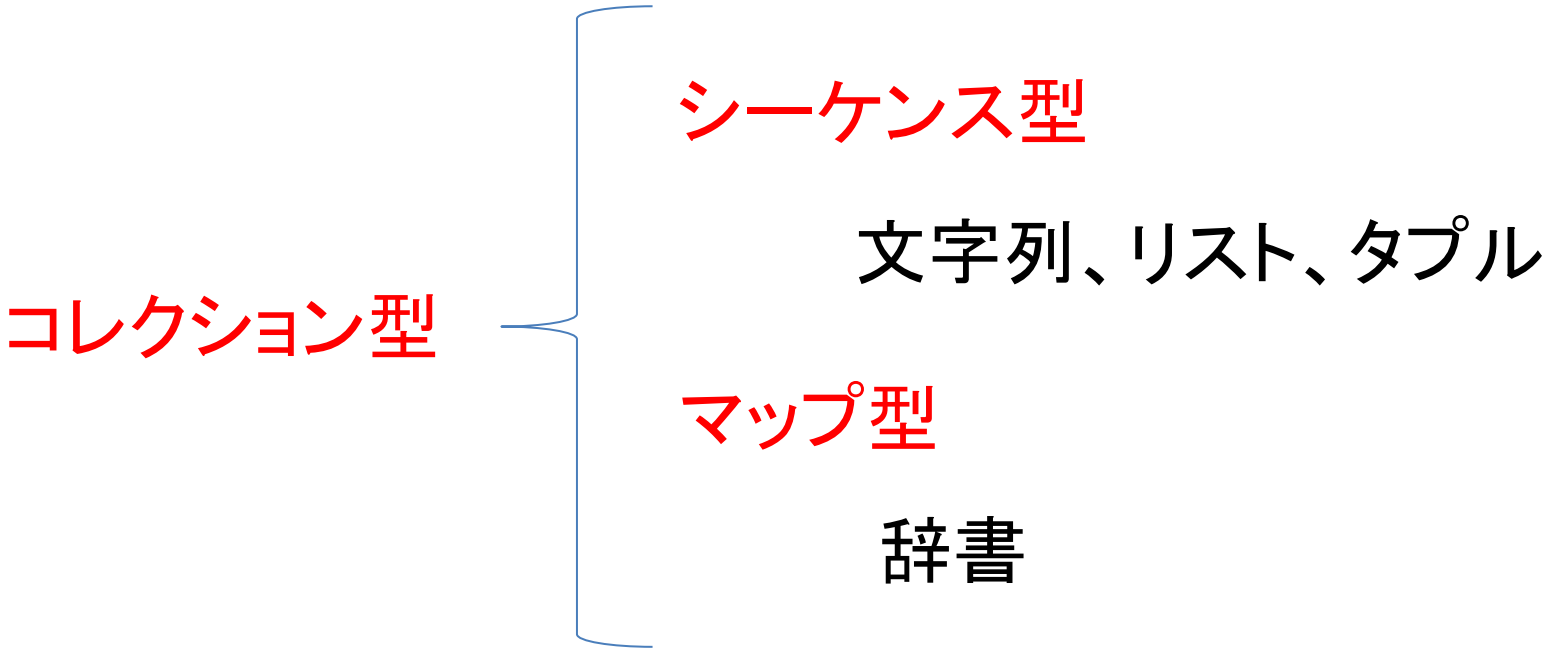
a = (1, 2, 3, 4, 5)

a + (10, 20) → (1, 2, 3, 4, 5, 10, 20)

タプルの足し算はタプル

コレクション型

リストのように複数の要素をデータに持てるデータ型



変更可能と変更不可能

コレクション型

- 変更可能
  - リスト、辞書
- 変更不可能
  - 文字列、タプル

組み込み型の相互変換

list ([シーケンス])

リストに変換

tuple ([シーケンス])

タプルに変換

str ([オブジェクト])

ふさわしい表記の文字列に変換

課題

以下は 1.txt から 395.txt のテキストファイル、utf-8 の日本語全角文字のみ

http://nlp.dse.ibaraki.ac.jp/~shinnou/netnews-utf8.zip

これらのファイルから平仮名 bi-gram を作成する
bi-gram の頻度が大きい順に 100 個表示せよ

```
[~/UNIV/2015/python-zemi 10] python kadai1.py netnews-utf8/*
した 1531
てい 1315
った 1200
して 1138
する 971
いる 804
など 696
から 658
ない 616
され 607
いた 547
って 520
こと 510
れた 459
ると 424
わそ 347
```