データ構造

初めてのプログラミング

2020年度

只木進一（理工学部）

リスト以外の構造

- タプル (tuple)
  - 値の組
- 集合
  - 同じ要素は一つだけ
- 辞書
  - キーと値の組

タプル
tuple

- 値の組を柔軟に作る

```
data=(1,2)
data +=(3,)
print(data)
print(data[0])
```

要素取り出し

- 一旦作成したタプルの要素は変更できないことに注意

dataStructure/tuples.ipynb

タプル

- 括弧の省略
  - タプルへのデータ登録

```
data=1,2,3
```

  - タプルからのデータ取り出し

```
a,b,c = data
```

タプル
要素を順に操作

```
1. colors=('green','red','blue','yellow','orange')
2. for c in colors:
3.     print(c)

4. for i in range(len(colors)):
5.     print(f'{i} : {colors[i]}')
```

タプル
タプルを要素とするリスト

```
1.  results = [
2.      ('Bob',80),
3.      ('Sue',90),
4.      ('Tim',70),
5.      ('Beth',90)
6.  ]
7.  for (name,record) in results:
8.      print(f'{name}の成績は{record}点')
```

出力

```
Bobの成績は80点
Sueの成績は90点
Timの成績は70点
Bethの成績は90点
```

タプル
要素がリストの場合

```
colors2 = (
    ('red',[255,0,0]),
    ('green',[0,255,0]),
    ('blue',[0,0,255]),
    ('yellow',[255,255,0]),
    ('orange',[255,165,0])
)
for c in colors2:
    print(f'{c[0]}={c[1]}')
```

```
red=[255, 0, 0]
green=[0, 255, 0]
blue=[0, 0, 255]
yellow=[255, 255, 0]
orange=[255, 165, 0]
```

集合
Set

- リストと異なり、同じ要素は一つだけに制限される

```
1. colorSet = {'red',‘green','blue'}
2. colorSet.add('yellow')
3. print(colorSet)
4. colorSet.remove('blue')
5. for c in colorSet:
6.     print(c)
7. c = colorSet.pop()
8. print(c)
9. print(colorSet)
```

```
{‘green', 'red', 'yellow', 'blue'}
green
red
yellow
green
{'red', 'yellow'}
```

dataStructure/Sets.ipynb

集合の操作

- add(要素) : 要素を追加
- remove(要素) : 要素を削除
- pop() : 要素を削除
  - 削除する要素を指定できない
- clear() : 全ての要素を削除

集合
空の集合、変更できない集合

```
setB=set()#空の集合を生成
print(setB)
setB.add('orange')
print(setB)
setC = frozenset(['apple','orange'])#変更できない集合
setC.add('banana')
```

```
set()
{'orange'}

---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-10-5768d0feef7c> in <module>
      4 print(setB)
      5 setC = frozenset(['apple','orange'])#変更できない集合
----> 6 setC.add('banana')

AttributeError: 'frozenset' object has no attribute 'add'
```

集合
集合の演算

```
1.  set1 = {'red','green','blue'}
2.  set2 = {'red','yellow','orange'}
3.  set3 = set1.union(set2)#和集合
4.  print(set3)
5.  set3 = set1 | set2
6.  print(set3)
7.  set4 = set1.intersection(set2)#共通部分
8.  print(set4)
9.  set4 = set1 & set2
10. print(set4)
11. set5 = set4 | {'red'}#同じオブジェクトは一度しか入らない
12. print(set5)
```

```
{'red', 'yellow', 'green', 'orange', 'blue'}
{'red', 'yellow', 'green', 'orange', 'blue'}
{'red'}
{'red'}
{'red'}
```

辞書
Dictionary

- キーと値の組

```
1. airports = {'HSG':'佐賀有明空港','FUK':'福岡空港','CTS':'新千歳空港'}
2. airports['HND']='羽田空港'#新しい組の追加
3. for key in airports:
4.     print(f'{airports[key]}のコードは{key}')
```

```
佐賀有明空港のコードはHSG
福岡空港のコードはFUK
新千歳空港のコードはCTS
羽田空港のコードはHND
```

dataStructure/dictionaries.ipynb

- 辞書からキーと値を取り出すことができる

```
print(airports.keys())
print(airports.values())
```

- キーと値の組を順に辞書に登録する例

```
1. records = [
2.     ('Tim',80),('Ann',90),('Bob',70),('Ray',95),('Joe',85)
3. ]
4. rd = {}
5. for name,r in records:
6.     rd[name]=r
7. print(rd)
```

次回

- 10章「ユーザ定義関数」