



1

データ構造

初めてのプログラミング

2019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

➡ 値の組を柔軟にする

```
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は変更できないことに注意

[dataStructure/tuples.ipynb](#)

タプル

➡ 括弧の省略

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

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

タプル

要素を順に操作

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

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

タプル

タプルを要素とするリスト

```
results = [  
    ('Bob',80),  
    ('Sue',90),  
    ('Tim',70),  
    ('Beth',90)  
]  
for (name,record) in results:  
    print(f'{name}の成績は{record}点')
```

```
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

```
colorSet = {'red','breen','blue'}  
colorSet.add('yellow')  
print(colorSet)  
colorSet.remove('blue')  
for c in colorSet:  
    print(c)  
c = colorSet.pop()  
print(c)  
print(colorSet)
```

```
{'breen', 'red', 'yellow', 'blue'}  
breen  
red  
yellow  
breen  
{'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

```
set1 = {'red','green','blue'}
set2 = {'red','yellow','orange'}
set3 = set1.union(set2)#和集合
print(set3)
set3 = set1 | set2
print(set3)
set4 = set1.intersection(set2)#共通部分
print(set4)
set4 = set1 & set2
print(set4)
set5 = set4 | {'red'}#同じオブジェクトは一度しか入らない
print(set5)
```

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

辞書 Dictionary

➡ キーと値の組

```
airports = {'HSG':'佐賀有明空港','FUK':'福岡空港','CTS':'新千歳空港'}  
airports['HND']='羽田空港'#新しい組の追加  
for key in airports:  
    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
```

- ➡ 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

```
records = [  
    ('Tim',80),('Ann',90),('Bob',70),('Ray',95),('Joe',85)  
]  
rd = {}  
for name,r in records:  
    rd[name]=r  
print(rd)
```

次回

➡ 10章「ユーザ定義関数」