# 2018 Fall Python Study

7. Data Structure

# Data

자료

# Structure

구조

# List

[ ]

$$[1, 2, 3, \cdots, a, b]$$

```
# This is my shopping list
shoplist = ['apple', 'mango', 'carrot', 'banana']
print(shoplist)
```

# len( )

```
print('I have', len(shoplist), 'items to purchase.')
```

```
shoplist = ['apple', 'mango', 'carrot', 'banana']
```

I have 4 items to purchase.

```
print('These items are:', end=' ')
for item in shoplist:
        print(item, end=' ')
```

```
shoplist = ['apple', 'mango', 'carrot', 'banana']
```

These items are: apple mango carrot banana

```
print('\nI also have to buy rice.')
shoplist.append('rice')
print('My shopping list is now', shoplist)
```

```
shoplist = ['apple', 'mango', 'carrot', 'banana']
```

```
I also have to buy rice.
My shopping list is now ['apple', 'mango', 'carrot', 'banana', 'rice']
```

```
print('I will sort my list now')
shoplist.sort()
print('Sorted shopping list is', shoplist)
```

```
shoplist = ['apple', 'mango', 'carrot', 'banana']
```

```
I will sort my list now
Sorted shopping list is ['apple', 'banana', 'carrot', 'mango', 'rice']
```

```
print('The first item I will buy is', shoplist[0])
olditem = shoplist[0]
del shoplist[0]
print('I bought the', olditem)
print('My shopping list is now', shoplist)
```

```
shoplist = ['apple', 'mango', 'carrot', 'banana']
```

```
The first item I will buy is apple
I bought the apple
My shopping list is now ['banana', 'carrot', 'mango', 'rice']
```

# Dictionary

{ }

{ “a” : 1, “b” : 2, … }

# Key & Value

# Why Dictionary?

```
# 'ab' is short for 'a'ddress'b'ook

ab = {
    'Swaroop': 'swaroop@swaroopch.com',
    'Larry': 'larry@wall.org',
    'Matsumoto': 'matz@ruby-lang.org',
    'Spammer': 'spammer@hotmail.com'
}
```

```
print("Swaroop's address is", ab['Swaroop'])
```

```
ab = {
    'Swaroop': 'swaroop@swaroopch.com',
    'Larry': 'larry@wall.org',
    'Matsumoto': 'matz@ruby-lang.org',
    'Spammer': 'spammer@hotmail.com'
}
```

Swaroop's address is swaroop@swaroopch.com

```
# Deleting a key-value pair
del ab['Spammer']
```

```
ab = {
    'Swaroop': 'swaroop@swaroopch.com',
    'Larry': 'larry@wall.org',
    'Matsumoto': 'matz@ruby-lang.org',
    'Spammer': 'spammer@hotmail.com'
}
```

```
print('\nThere are {} contacts in the address-book\n'.format(len(ab)))
```

```
ab = {
    'Swaroop': 'swaroop@swaroopch.com',
    'Larry': 'larry@wall.org',
    'Matsumoto': 'matz@ruby-lang.org'
}
```

There are 3 contacts in the address-book

```
for name, address in ab.items():
        print('Contact {} at {}'.format(name, address))
```

```
ab = {
    'Swaroop': 'swaroop@swaroopch.com',
    'Larry': 'larry@wall.org',
    'Matsumoto': 'matz@ruby-lang.org'
}
```

Contact Swaroop at swaroop@swaroopch.com
Contact Larry at larry@wall.org
Contact Matsumoto at matz@ruby-lang.org

```
# Adding a key-value pair
ab['Guido'] = 'guido@python.org'
```

```
ab = {
    'Swaroop': 'swaroop@swaroopch.com',
    'Larry': 'larry@wall.org',
    'Matsumoto': 'matz@ruby-lang.org’,
    'Guido': 'guido@python.org'
}
```

```
if 'Guido' in ab:
        print("\nGuido's address is", ab['Guido'])
```

```
ab = {
    'Swaroop': 'swaroop@swaroopch.com',
    'Larry': 'larry@wall.org',
    'Matsumoto': 'matz@ruby-lang.org’,
    'Guido': 'guido@python.org'
}
```

Guido's address is guido@python.org

# Tuple

( )

$$(1, 2, 3, \cdots, a, b)$$

```
# I would recommend always using parentheses
# to indicate start and end of tuple
# even though parentheses are optional.
# Explicit is better than implicit.
zoo = ('python', 'elephant', 'penguin')
print('Number of animals in the zoo is', len(zoo))
```

Number of animals in the zoo is 3

```
new_zoo = 'monkey', 'camel', zoo # parentheses not required but are a good idea
print('Number of cages in the new zoo is', len(new_zoo))
print('All animals in new zoo are', new_zoo)
print('Animals brought from old zoo are', new_zoo[2])
print('Last animal brought from old zoo is', new_zoo[2][2])
print('Number of animals in the new zoo is’,len(new_zoo)-1+len(new_zoo[2]))
```

```
zoo = ('python', 'elephant', 'penguin')
```

```
Number of cages in the new zoo is 3
All animals in new zoo are ('monkey', 'camel', ('python', 'elephant', 'penguin'))
Animals brought from old zoo are ('python', 'elephant', 'penguin')
Last animal brought from old zoo is penguin
Number of animals in the new zoo is 5
```

# List

List

Tuple

Mutable

Immutable

# Sequence

```
shoplist = ['apple', 'mango', 'carrot', 'banana']
name = 'swaroop'
```

```
# Indexing or 'Subscription' operation #
print('Item 0 is', shoplist[0])
print('Item 1 is', shoplist[1])
print('Item 2 is', shoplist[2])
print('Item 3 is', shoplist[3])
print('Item -1 is', shoplist[-1])
print('Item -2 is', shoplist[-2])
print('Character 0 is', name[0])
```

```
shoplist = ['apple', 'mango', 'carrot', 'banana']
name = 'swaroop'
```

```
Item 0 is apple
Item 1 is mango
Item 2 is carrot
Item 3 is banana
Item -1 is banana
Item -2 is carrot
Character 0 is s
```

```
# Slicing on a list #
print('Item 1 to 3 is', shoplist[1:3])
print('Item 2 to end is', shoplist[2:])
print('Item 1 to -1 is', shoplist[1:-1])
print('Item start to end is', shoplist[:])
```

```
shoplist = ['apple', 'mango', 'carrot', 'banana']
name = 'swaroop'
```

```
Item 1 to 3 is ['mango', 'carrot']
Item 2 to end is ['carrot', 'banana']
Item 1 to -1 is ['mango', 'carrot']
Item start to end is ['apple', 'mango', 'carrot', 'banana']
```

```
# Slicing on a string #
print('characters 1 to 3 is', name[1:3])
print('characters 2 to end is', name[2:])
print('characters 1 to -1 is', name[1:-1])
print('characters start to end is', name[:])
```

```
shoplist = ['apple', 'mango', 'carrot', 'banana']
name = 'swaroop'
```

```
characters 1 to 3 is wa
characters 2 to end is aroop
characters 1 to -1 is waroo
characters start to end is swaroop
```