- **created by**: Sudip Ghimire
- **URL**: https://www.sudipghimire.com.np
- **GitHub**: https://github.com/ghimiresdp

go to course contents

Chapter 3.5. Nesting of Iterables

Table of Contents

Introduction to nesting

Nesting is the process of creating an iterable inside another iterable. For example If we add List inside another List, then we call it as a nested list. We also call nested list as n-dimensional list.

Creating a nested iterable

A nested List is a list that contains a list. For example:

Example 1: Nested Lists and Tuples

```
nested_list = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
nested_tuple = (('a', 'apple'), ('b', 'ball'), ('c', 'cat'))

# we can also represent them multiline to make it more readable.

nested_list = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9],
]
nested_tuple = (
    ('a', 'apple'),
    ('b', 'ball'),
    ('c', 'cat'),
)
```

Example 2: Nested dictionary

```
person = {
    'name': 'John',
    'age': 20,
    'company': {
        'name': 'Microsoft',
        'established': 1974,
        'location': 'Albuquerque, New Mexico'
    }
}
```

Example 3: *Nesting between different data types* The example below shows a `list` of students(`dict`) in which the value of the key `'majors'` is a `tuple` of `strings`

```
students = [
    {
        'name': 'John Doe',
        'age': 20,
        'majors': ('Mathematics', 'Physics')
    },
    {
        'name': 'Jane Doe',
        'age': 21,
        'majors': ('Biology', 'Neurosurgery')
    },
]
```

Accessing items from the nested iterable

The example below represents a detail of a fictional `Star Wars` Movie. The example below shows statements for getting different elements from the data shown below:

```
person = {
    'name': 'Yoda',
    'age': 900,
    'species': "Yoda's",
    'language': 'Galactic Basic',
    'affiliation': {
        'name': 'Jedi',
        'member_size': 12,
        'weapons': ['Force', 'Lightsaber', 'swords', 'batons']
    },
    'weapon': 'lightsaber',
}

print(person['affiliation']['name'])            # 'Jedi'
print(person['affiliation']['weapons'][1])      # 'Lightsaber'
```

Modifying items from the nested iterable

Modifying items inside of the nested iterable is similar to that of regular iterables.

```
students = [
    {
        'name': 'John Doe',
        'age': 20,
        'majors': ('Mathematics', 'Physics')
    },
    {
        'name': 'Jane Doe',
        'age': 21,
        'majors': ('Biology', 'Neurosurgery')
    },
]

# Adding a new student to the list of students
new_student = {
    'name': 'John Lennon',
    'age': 50,
    'Majors': ('Music', 'Vocals')
}
students.append(new_student)

# changing the age of `john doe` from 20 to 30
students[0]['age'] = 30

print(students)
```

Output:(Reformatted)

```
[
    {
        'name': 'John Doe'
        'age': 30, 'majors': ('Mathematics', 'Physics'),
    },
    {
        'name': 'Jane Doe',
        'age': 21,
        'majors': ('Biology', 'Neurosurgery')
    },
    {
        'name': 'John Lennon'
        'age': 50,
        'Majors': ('Music', 'Vocals')
    }
]
```