Структуры данных

№ урока:	6	**Курс:**	Python Базовый

Средства обучения:	Персональный компьютер/ноутбук стандартной производительности

Обзор, цель и назначение урока

Познакомиться с усложненными вариациями комбинаций структур данных, рассмотреть смысл их применения и тонкости работы с ними. Так же будет рассмотрены так называемые comprehensions. После данного урока вы сможете работать со сложными структурами данных и использовать comprehensions в ваших задачах.

Изучив материал данного занятия, учащийся сможет:

- Понимать сложные структуры данных и работать с ними.
- Знать, что такое comprehensions и уметь их применять на практике.

Содержание урока

1. Примеры комбинаций структур данных
2. Что такое comprehensions в Python
3. Решение задач

Резюме

- Комбинировать можно почти любые структуры данных между собой, создавая тем самым сложные и комплексные объекты. Например, вы можете сделать список словарей или сделать словарь со списками словарей, в которых будут списки строк и т.д. Фантазия тут может быть безгранична.
- Однако для каждой задачи существует своя идеальная структура данных. Например, если вам нужно хранить данные о пользователях, то удобно будет сделать список словарей. Таким образом вам будет легко посчитать их количество (len), взять, удалить, добавить новый элемент. А самое главное, это то, что вы сможете очень легко работать с такой структурой даже без кода. В каждом словаре будет наглядно видно пары ключ-значение, где вы сможете записать любые данные о пользователях.
- Но если вы решите сделать эту задачу через список списков, то создадите себе лишние сложности. Например, вы банально можете забыть какая позиция во внутренних списках за что отвечает (25 это возраст или зарплата или что?). В случае же со словарями - у вас будет {'age': 25} и все понятно.
- Бывает такое, что структуры бывают еще сложнее. Это зависит от сложности самих данных, с которыми вы работаете. Если у пользователей есть данные о приобретенной ими недвижимости, то у каждого пользователя внутри его словаря по ключу 'real_estate' будет свой список словарей с данными о недвижимости. А если у каждого купленного дома еще будут данные о его мебели, то это еще новые списки словарей. Таким образом структуры данных могут быть очень огромными и глубокими.

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- Comprehensions — это один из one-liners языка Python. Специальный синтаксис, который позволит вам работать с итерируемыми структурами с помощью цикла for одной строчкой. Помимо полезного свойства сокращения пространства кода, еще стоит отметить, что comprehensions работают быстрее обычных циклов.
- Синтаксис comprehensions следующий [object for object in iterable]. Данная запись просто продублирует итерируемый объект. В случае если вы хотите произвести дополнительные преобразования с ним, например возвести в квадрат каждый элемент списка, то это будет так: sqaures_list = [element**2 for element in my_list]

Закрепление материала

- Какие структуры данных можно комбинировать между собой?
- Можно ли использовать вложенные comprehensions?
- Является ли всегда разумным использовать comprehensions вместо циклов так как они быстрее?

Дополнительное задание

Пофантазируйте со структурами данных. Пробуйте объединять разные структуры и посмотреть, как с ними работать (списки словарей списков, словари словарей словарей, списки списков со словарями списков и т.д.) Напишите циклы для итерации таких структур.

Самостоятельная деятельность учащегося

1. Напишите комбинацию циклов чтобы проитерировать все элементы сложной структуры данных вида d = {"a": {"1": [...], "2": [...], ...}, ...} (словарь словарей в которых лежат списки)
2. Напишите comprehension, который итерирует строку "1_2,40_5,5_32" (разбить по запятым) и каждый элемент разбивает по символу "_" и полученные элементы приводит к типу int, и складывает их, тем самым образуя список целых чисел. (Вы можете использовать lambda функцию или написать обычную, которая принимает строку вида "1_2" на вход и возвращает число как сумму значений из строки; используйте метод split()).

Рекомендуемые ресурсы

- Nested dicts in Python
- Рекурсивная работа с вложенными словарями
- Вложенные структуры
- Comprehensions
- List comprehensions
- Double comprehension

CyberBionic systematics

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com