

Python Базовый

Структуры данных


Python Базовый

Introduction



Бондаренко Кирилл

Senior Data scientist, CreatorIQ

 [profile.php?id=100011447245832](https://www.facebook.com/profile.php?id=100011447245832)

 [kirill-bond/](https://www.linkedin.com/in/kirill-bond/)

 [@bond.kirill.alexandrovich](https://www.telegram.com/@bond.kirill.alexandrovich)



Python Базовый

Тема урока

Структуры данных

Python Базовый

План урока

1. Примеры комбинации структур данных и работа с ними
2. Comprehensions в Python
3. Решение задач

Python Базовый

Сложные объекты

```
1 user = {
2     "name": "JohnDoe",
3     "info": {
4         "basic": {
5             "age": 25,
6             "salary": 5000
7         },
8         "additional": {
9             "study": "mathematics",
10            "family": "married"
11        },
12        "special": {
13            "projects": [
14                {"name": "quantum_computer", "stage": "in progress"},
15                {"name": "laser gun", "stage": "in production"}
16            ]
17        }
18    }
19 }
```

```
22 def get_data(data_dict, keys):
23     data = data_dict
24     for key in keys:
25         data = data[key]
26     return data
27
28
29 def get_data_rec(data_dict, keys, index=0):
30     if index < len(keys):
31         return get_data_rec(data_dict[keys[index]], keys, index + 1)
32     return data_dict
33
34
35 print(get_data(user, ['info', 'special', 'projects']))
36
37 print(get_data_rec(user, ['info', 'special', 'projects']))
38
39
```

Python Базовый

Comprehensions

List comprehension

```
1 chars = [char for char in "abcde"] # ['a', 'b', 'c', 'd', 'e']
2
3 names = ['mike', 'john', 'sally', 'mary', 'kate', 'robert']
4 names = [name.capitalize() for name in names] # ['Mike', 'John', ...]
5
6 even_nums = [num for num in [1, 2, 3, 4, 5, 6, 7, 8] if num % 2 == 0] # [2, 4, 6, 8]
7
8 unique_values = {value**2 for value in [1, 3, 5, 5, 2, 1, 7, 3]} # {1, 4, 9, 49, 25}
9
10
```

Set comprehension

Dict comprehension

```
1 data = ["John_25", "Sally_19", "Susan_35", "Jack_16"]
2 name_age_dict = {v.split('_')[0]: v.split('_')[1] for v in data}
3 print(name_age_dict) # {'John': '25', 'Sally': '19', ...}
4
5 values_squares = {value: value ** 2 for value in range(10)}
6 print(values_squares) # {0: 0, 1: 1, 2: 4, 3: 9, ...}
7
8 months = (num for num in range(12))
9 print(months) # (0, 1, 2, 3, ...)
10
```

Tuple comprehension

Python Базовый

“Double Comprehension” и “flattening”

```
1 matrix = [[j for j in range(2)] for i in range(2)]
2 print(matrix) # [[0, 1], [0, 1]]
3
4 flatten_matrix = [val for sublist in matrix for val in sublist]
5 print(flatten_matrix) # [0, 1, 0, 1]
6
7
```

Python Базовый

Задачи

1. Написать функцию, которая принимает аргументы `index`, `count` и создает словарь с ключом "ID": `index`, полем "values" как список строк, сгенерированных в диапазоне от 0 до `count` вида "i_index", где `i` - значения от 0 до `count`, например: `index=1, count = 3: {"ID": 1, "values": ["0_1", "1_1", "2_1"]}`.
2. Сделать функцию, которая принимает аргумент `count` и вызывает функцию из первого задания и передает туда `i, j` где `i` - значения от 0 до `count`, а `j` - значения от `count` до 0. Полученный словарь она добавляет в список, делая список словарей.
3. Написать `flatten list comprehension` для полученного списка словарей, где финальный результат будет список из "распакованных" списков "values" в каждом словаре.

PS: Используйте только `comprehension`

Чтобы получить список вида `[(0,9) (1,8) (2,7) (3,6) ...]` для выполнения 2 задачи с генерацией значений `i`, `j` используйте `zip(range(count), list(range(count))[::-1])` и итерируйте сразу `i, j`.

Python Базовый

Решение

```
1 def func(index, count):
2     return {"ID": index, "values": ["{}_{}".format(index, value) for value in range(count)]}
3
4
5 def generate(count):
6     return [func(i, j) for i, j in zip(range(count), list(range(count))[:-1])]
7
8
9 r = generate(10)
10 f = [value for sublist in r for value in sublist['values']]
11 print(f)
12 # ['0_0', '0_1', '0_2', '0_3', '0_4', '0_5', '0_6', '0_7', '0_8', '1_0', '1_1', '1_2', '1_3', '1_4', '1_5', '1_6',
13 #  '1_7', '2_0', ... ]
14
15
```

Информационный видеосервис для разработчиков программного обеспечения



Проверка знаний

TestProvider.com



Проверьте как Вы усвоили данный материал на TestProvider.com

TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.

Python Базовый

Спасибо за внимание! До новых встреч!



Бондаренко Кирилл
Senior Data scientist, CreatorIQ

