Урок 3. Функции. Словари

1. Написать функцию num_translate(), переводящую числительные от 0 до 10 с английского на русский язык. Например:

```
>>> num_translate("one")
"один"
>>> num_translate("eight")
"восемь"
```

Если перевод сделать невозможно, вернуть None. Подумайте, как и где лучше хранить информацию, необходимую для перевода: какой тип данных выбрать, в теле функции или снаружи.

2. * (вместо задачи 1) Доработать предыдущую функцию в num_translate_adv(): реализовать корректную работу с числительными, начинающимися с заглавной буквы — результат тоже должен быть с заглавной. Например:

```
>>> num_translate_adv("One")
"Один"
>>> num_translate_adv("two")
"два"
```

3. Написать функцию thesaurus(), принимающую в качестве аргументов имена сотрудников и возвращающую словарь, в котором ключи — первые буквы имён, а значения — списки, содержащие имена, начинающиеся с соответствующей буквы. Например:

```
>>>  thesaurus("Иван", "Мария", "Петр", "Илья")
{
    "И": ["Иван", "Илья"],
    "М": ["Мария"], "П": ["Петр"]
}
```

Подумайте: полезен ли будет вам оператор распаковки? Как поступить, если потребуется сортировка по ключам? Можно ли использовать словарь в этом случае?

4. * (вместо задачи 3) Написать функцию thesaurus_adv(), принимающую в качестве аргументов строки в формате «Имя Фамилия» и возвращающую словарь, в котором ключи — первые буквы фамилий, а значения — словари, реализованные по схеме предыдущего задания и содержащие записи, в которых фамилия начинается с соответствующей буквы. Например:

```
>>> thesaurus_adv("Иван Сергеев", "Инна Серова", "Петр Алексеев", "Илья Иванов", "Анна Савельева")
{
    "А": {
        "П": ["Петр Алексеев"]
    },
    "И": {
        "И": ["Илья Иванов"]
    },
```

```
    "С": {
        "И": ["Иван Сергеев", "Инна Серова"],
        "А": ["Анна Савельева"]
    }
}
```

Как поступить, если потребуется сортировка по ключам?

5. Реализовать функцию get_jokes(), возвращающую n шуток, сформированных из трех случайных слов, взятых из трёх списков (по одному из каждого):

```
nouns = ["автомобиль", "лес", "огонь", "город", "дом"]
adverbs = ["сегодня", "вчера", "завтра", "позавчера", "ночью"]
adjectives = ["веселый", "яркий", "зеленый", "утопичный", "мягкий"]
        Например:
>>> get_jokes(2)
["лес завтра зеленый", "город вчера веселый"]
```

Документировать код функции.

Сможете ли вы добавить еще один аргумент — флаг, разрешающий или запрещающий повторы слов в шутках (когда каждое слово можно использовать только в одной шутке)? Сможете ли вы сделать аргументы именованными?

Задачи со * предназначены для продвинутых учеников, которым мало сделать обычное ДЗ.