

Примеры решения задач

1. Ввод и обработка элементов словаря.

```
# словарь - это структура, хранящая пары ключ:значение
# ключ и значение могут быть любого типа, не только строчные
# например, численность населения России с 2000г. по годам (ключ - год,
значение - численность населения)
population = {2000: 146890128, 2001: 146303611, 2002: 145166731,
              2003: 144963650, 2004: 144168205, 2005: 143474219,
              2006: 142753551, 2007: 142220968, 2008: 142008838,
              2009: 141903979, 2010: 142856536, 2011: 142865433,
              2012: 143056983, 2013: 143347059, 2014: 143666931,
              2015: 146267288, 2016: 146544710, 2017: 145804372,
              2018: 146880432}
```

```
# однако, в отличие от строк, списков и кортежей, словари не упорядочены:
print('Численность населения России, раз:', population)
```

```
# хотя порядок вывода постоянный, что зависит от реализации интерпретатора:
print('Численность населения России, два:', population)
print('Численность населения России, три:', population)
```

```
# добавлять элементы в словарь можно без дополнительных функций:
population[1999] = 147539426
print('Численность населения России:', population)
```

```
# извлечь с удалением можно по ключу:
year = 2016
year_population = population.pop(year)
print('Численность населения России в', year, 'г.:', year_population)
print('Численность населения России:', population)
```

```
# или первый элемент в виде кортежа:
year_population = population.popitem()
print('Численность населения России:', year_population)
print('Численность населения России:', population)
```

```
# посмотреть значение в словаре, не удаляя элемент:
year = 2015
print('Численность населения России в', year, 'г.:', population[year])
```

```
# но если элемента нет, будет выведена ошибка:
# year = 2017
# print('Численность населения России в', year, 'г.:', population[year])
# чтобы ее избежать, необходимо проверять, есть ли такой ключ:
year = 2017
if year in population:
    print('Численность населения России в', year, 'г.:', population[year])
else:
    print('Информации о численности населения России за', year, 'нет.')

# либо используя метод get(key), который вместо ошибки выдаст значение None
print('Численность населения России в', year, 'г.:', population.get(year))
```

```
# для прохода по ключам в цикле можно использовать:
```

```
print('Годы, за которые есть записи (1 способ)')
for years in population:
    print(years)
```

```
# или, что то же самое:
```

```
print('Годы, за которые есть записи (2 способ)')
for years in population.keys():
    print(years)
```

```
# для прохода по всем значениям нужно использовать метод values:
```

```
print('Известные годовые значения численности населения России')
for year_population in population.values():
    print(year_population)
```

2. Словарь словарей.

```
# зададим несколько словарей:
```

```
ru_en = {'красный': 'red', 'зеленый': 'green', 'синий': 'blue', 'желтый': 'yellow'}
en_ru = {'red': 'красный', 'green': 'зеленый', 'blue': 'синий', 'yellow': 'желтый'}
fr_ru = {'rouge': 'красный', 'vert': 'зеленый', 'bleu': 'синий', 'jaune': 'желтый'}
ru_fr = {'красный': 'rouge', 'зеленый': 'vert', 'синий': 'bleu', 'желтый': 'jaune'}
en_de = {'red': 'rot', 'green': 'grün', 'blue': 'blau', 'yellow': 'gelb'}
de_en = {'rot': 'red', 'grün': 'green', 'blau': 'blue', 'gelb': 'yellow'}
de_fr = {'rot': 'rouge', 'grün': 'vert', 'blau': 'bleu', 'gelb': 'jaune'}
fr_de = {'rouge': 'rot', 'vert': 'grün', 'bleu': 'blau', 'jaune': 'gelb'}
```

```
# получить перевод слова теперь можно так:
```

```
print('На английском \'желтый\': ', ru_en['желтый'])
print('На немецком \'blue\': ', en_de['blue'])
```

```
# можно объединить все словари в словарь словарей:
```

```
lang_dictionary = {'Русско-английский': ru_en, 'Англо-русский': en_ru,
                   'Франко-русский': fr_ru, 'Русско-французский': ru_fr,
                   'Англо-немецкий': en_de, 'Немецко-английский': de_en,
                   'Немецко-французский': de_fr, 'Франко-немецкий': fr_de}
```

```
# и теперь обращаться к словарям через общий словарь:
```

```
print('На английском \'желтый\': ', lang_dictionary['Русско-английский']['желтый'])
```

```
# и можно в цикле:
```

```
word = 'синий'
langs = ['Русско-английский', 'Русско-французский']
for lang in langs:
    print(lang, 'перевод для слова', word, '-', lang_dictionary[lang][word])
```

3. Генерация словаря.

```
# как и список, словарь можно создать при помощи
# comprehension, используя фигурные скобки.
# например, создадим кортеж ключей:
keys = ('Имя', 'Фамилия', 'Группа', 'Возраст')

# затем запросим у пользователя ответы на них и создадим словарь:
personal_card = {key: input('{}: '.format(key)) for key in keys}
print(personal_card)
print(type(personal_card))

# comprehension можно комбинировать, например, для создания
# списка словарей:
file_cabinet = [{key: input('{}.{}: '.format(number, key)) for key in keys}
                for number in range(3)]
print(file_cabinet)
```