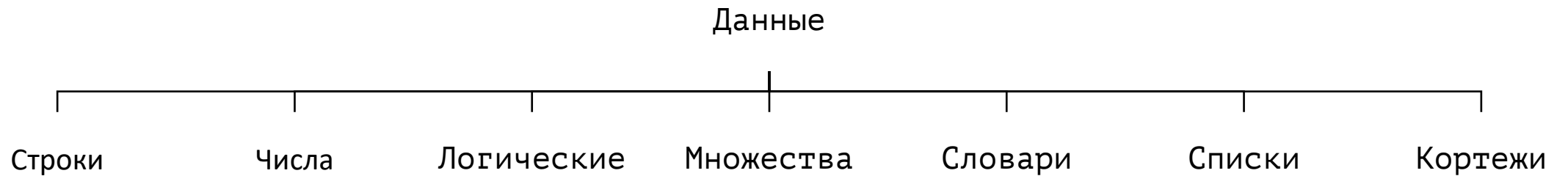


Решение задач по словарям, циклам, условиям

Типы данных



Узнать тип данных

```
>>> print(type(x))
```

Словари

неупорядоченный набор данных произвольного типа с доступом по ключу

Синтаксис

создание

```
имя = {'ключ 1': значение 1, ...,  
      'ключ N': значение N}
```

```
имя = dict(ключ 1 = значение  
1, ..., ключ N = значение N)
```

добавление элемента

```
имя[новый ключ] = значение
```

удаление элемента

```
del имя[ключ]
```

вызов словаря

```
имя
```

Зачем они нужны?

- Для подсчета числа каких-то объектов. В этом случае нужно завести словарь, в котором ключами являются объекты, а значениями – их количество.
- Для хранения каких-либо данных, связанных с объектом. Ключи – объекты, значения – связанные с ними данные.
- Установка соответствия между объектами (например, “родитель–потомок”). Ключ – объект, значение – соответствующий ему объект.
- Если нужен обычный массив, но максимальное значение индекса элемента очень велико, и при этом будут использоваться не все возможные индексы (так называемый “разреженный массив”), то можно использовать ассоциативный массив для экономии памяти.

Методы словарей

<code>имя.clear()</code>	Очищает словарь
<code>имя.copy()</code>	Возвращает копию словаря
<code>имя.items()</code>	Возвращает пары (ключ, значение)
<code>имя.keys()</code>	Возвращает ключи
<code>имя.values()</code>	Возвращает значения в словаре
<code>имя.update({ключ: значение})</code>	Обновляет словарь, добавляя пары.
<code>имя.get('ключ')</code>	Возвращает значение по указанному ключу

Перебор словаря

```
>>> a = {1: 'one', 2: 'two', 3: 'three'}
>>> for key, value in a.items():
print(key, ': ', value)
1 : one
2 : two
3 : three
```

```
>>> a = {1: 'one', 2: 'two', 3: 'three'}
>>> for key in a.keys():
print(key)
1
2
3
```

```
>>> a = {1: 'one', 2: 'two', 3: 'three'}
>>> for val in a.values():
print(val)
one
two
three
```

Создайте словарь, где ключ – это римская цифра, а значение арабская. Для цифр от 1 до 4.

Задание 1

Выведите цифру 2 из словаря

Задание 2

Добавьте в словарь число 11, с ключом «XI»

Задание 3

Измените значение под ключом 4 на «IV, четыре, four»

Задание 4

Выведите все ключи словаря

Задание 5

Выведите значение, которое хранится под ключом 3

Задание 6

Замените значение под ключом «I» на слово «один»

Логические

Операторы сравнения

<	Меньше
>	Больше
<=	Меньше или равно
>=	Больше или равно
==	Равенство
!=	Неравенство

Операторы сравнения возвращают значение специального логического типа **bool()**, принимающего либо значение **True** (истина), либо **False** (ложь).

```
>>> x = 5
>>> y = 4
>>> print(bool(x == y))
False
```

```
>>> x = 5
>>> y = 4
>>> print(bool(x > y))
True
```

```
>>> x = 'privet'
>>> print(bool(x))
True
#не пустая строка
```

Условная инструкция

Синтаксис

```
if Условие:  
    Блок инструкций 1  
elif Условие:  
    Блок инструкций 2  
else:  
    Блок инструкций 3
```

Условные операторы

or ()	или
and (&)	и
not	инверсия

```
>>> x = int(input())  
>>> if x < 0:  
        x = -x  
>>> print(x)
```

```
>>> x = int(input())  
>>> if x < -5:  
        print('low')  
>>> elif -5 <= x <= 5:  
        print('mid')  
>>> else:  
        print('high')
```

```
>>> x = -12  
>>> if x < 0:  
        print('negative')  
>>> else:  
        print('positive')  
negative
```

```
>>> x = 7  
>>> if x > 0 and x % 10  
    != 0:  
        x = x // 2  
>>> else:  
        print('NO')
```

```
x = 3
```

Задание 1

Написать функцию, которая будет проверять со скольких лет человек может стать президентом России?

-через **bool()**

-через **if, else**

Задание 2

Нужно проверить, является ли целая часть от выражение $x^y + 2 * y + 17,5$ кратной 3 **или** «большой или равной» 10. Если это так вывести True, иначе False. x и y вводится с клавиатуры. Реализовать программу 2 способами. Первый с помощью **bool()**, второй с помощью **if** и **else**.

Ввод

1

0.5

Вывод

True

Условные операторы

or ()	или
and (&)	и
not	инверсия

$x + y$	сложение
$x - y$	вычитание
$x * y$	умножение
x / y	деление
$x // y$	целая часть от деления
$x \% y$	остаток от деления
$x ** y$	возведение в степень
$abs(x)$	модуль числа

Циклы range(), for

Синтаксис

`range`(начало, конец, шаг)

`for` переменная `in`
последовательность:
Блок инструкций

`for` переменная `in`
`range`(начало, конец, шаг):
Блок инструкций

Диапазон range

```
>>> b = list(range(5, 10, 2))
>>> print(b)
[5, 7, 9]
```

```
>>> a = range(5, 10, 2)
>>> print(a)
range(5, 10, 2)
```

Цикл for

```
>>> for i in 10, 14, 'первый', 'второй':
>>>     print(i)
10
14
первый
второй
```

Вывод в обратном порядке

```
>>> x = list(reversed(range(5, 10)))
>>> print(x)
[9, 8, 7, 6, 5]
```

Цикл for + range()

```
>>> sum = 0
>>> for i in range(11, 13):
>>>     a = i + 5
>>>     sum = sum + a
>>> print(sum)
33
```

Даны числа от 5 до 26. Необходимо проверить является ли сумма остатков от деления этих чисел на N (вводится с клавиатуры) чётной и больше 1. Если да вывести +, иначе -. Из остатков от деления нужно создать список и вывести его в конце.

1. Создаем пустой список, куда будем записывать остатки от деления и 0 переменную куда будем суммировать остатки
2. Создаем цикл с **for/range** по диапазону
3. Ищем остаток от деления с помощью %. Не забываем, что N может быть вещественным числом
4. С помощью **append()**, добавляем значение остатка в наш пустой список
5. Добавляем остаток в сумму
6. Проверяем с помощью сложного условия является ли наша сумма четной и больше 1

Условные операторы

or (|) или

and (&) и

not инверсия

Задание 2

Евгению предоставили строку (мПЛОПВД0едиПР0ВЛРа0ВА0ПКкоПм), состоящую из русских букв разных регистров, и попросили очистить ее от заглавных литер.

1. Создаем переменную в которую нужно записать строку для очистки
2. Создаем пустую строку, куда запишем очищенный текст
3. Создаем цикл с **for/in**, по строке
4. Внутри цикла проверяем с помощью условия **if not** и функции **.isupper()** является ли буква заглавной, если **НЕТ**, то записываем ее в пустую строку (созданную в пункте 2)
5. Выводим очищенную строку (после цикла)

Задание 3

С помощью цикла заполнить словарь из 4 элементов, где ключ это овощ, а значение, это его цвет.

1. Создаем пустой словарь
2. Цикл **for/range()**
3. **input()** для значения и ключа
4. Добавляем

С помощью цикла вывести:

- Все ключи словаря (**имя.keys()**)
- Все значения словаря (**имя.items()**)
- Все пары словаря (**имя.values()**)

Цикл while

Синтаксис

`while` условие:
 Блок инструкций

Прерываем цикл

`while` условие:
 Блок инструкций
 `if` условие:
 `break`

Прерываем итерацию

`while` условие:
 Блок инструкций
 `if` условие:
 `continue`

Используется для неоднократного исполнения определенной инструкции до тех пор, пока заданное условие остается истинным.

```
>>> a = 0
>>> while a < 5:
>>>     a += 1
>>>     print(a)
1
2
3
4
5
```

```
>>> a = 0
>>> while a < 5:
>>>     a += 1
>>>     if a == 3:
>>>         break
>>>     print(a)
1
2
```

```
>>> a = 0
>>> while a < 5:
>>>     a += 1
>>>     if a == 3:
>>>         continue
>>>     print(a)
1
2
4
5
```

Синтаксис

`while` условие:
 Блок инструкций

Прерываем цикл

`while` условие:
 Блок инструкций
 if условие:
 break

Прерываем итерацию

`while` условие:
 Блок инструкций
 if условие:
 continue

`While` также можно использовать инструкцию `else`. В этом случае блок в `else` выполняется, когда условие цикла становится ложным

```
>>> a = 0
>>> while a < 5:
>>>     print('Условие выполняется')
>>>     a += 1
>>> else:
>>>     print('Условие не выполняется')
Условие выполняется
Условие выполняется
Условие выполняется
Условие выполняется
Условие выполняется
Условие не выполняется
```

Важно

Операторы `break` и `continue` аналогично применяются внутри цикла `for`

Задание 1

Задан список $A=[5, -3, 1, 8, 3, 10]$. С помощью `while` вычислить сумму элементов списка

Задание 2

Задано число a ($1 < a < 1.5$). Из чисел $1+1/2$, $1+1/3$, $1+1/4$..., (заметим, что каждое следующее число в последовательности меньше предыдущего) найдите первое, меньшее a .

Задание 3

Посчитать количество вхождений заданной строки в список `['abc', 'bcd', 'xvm', 'abc', 'abd', 'bcd', 'abc']`

*Подсчет количества элементов
в списке*

```
>>> len(A)  
6
```

Задание 4

Написать программу, которая для дня (число от 1 до 365) в 2021 году определяет номер дня недели. Первый день 2021 года – пятница. **Бонусное задание: определить дату**

1. Создаем переменную, которая будет запрашивать номер дня
2. Инициализируем переменную, хранящую день недели 01.01.2021 и счетчик – номер первого дня в году
3. Используем цикл **while**, который перебирает цифры, пока счетчик меньше проверяемого дня
4. Внутри цикла пишем условие, что нужно сделать, если номер дня недели стал больше 7
5. Вывод номера дня недели

Задание 5

Минюст опубликовал список СМИ, которых признали Иностранцами агентами: Дождь, МБХ, Медуза, The Insider, Голос Америки, ТК Настоящее время, Радио Свобода. Ваша задача написать код, который будет проверять есть ли интересующее вас СМИ (вводится с клавиатуры) в списке ИО.

1. Создаем переменную, которая будет запрашивать название СМИ
2. Создаем список с названиями ИО
3. Используем конструкцию **if** с условием **not in**, чтобы проверить находится ли СМИ в списке ИО (подсказка: `СМИ not in ИО`)
4. Сообщаем, что нужно вывести, если СМИ нет в списке
5. Пишем условие, что нужно вывести, если СМИ есть в списке (через **else**)

Практикум по словарям

Задание 1

Каким будет результат выполнения следующих операций?

```
>>> playlist = {'Imagine Dragons': 'Lonely',  
'The Kooks': 'If only' }  
>>> new_playlist = {'Harry Styles':  
'Falling', 'The Kooks': 'Naive'}  
>>> playlist.update(new_playlist)  
>>> print(playlist)
```

Задание 2

Дана информация по количеству каждого типа животных в зоомагазине на текущий момент. Нужно написать код, который будет считать количество животных (данные на вход могут меняться, нужно это учитывать и создать универсальное решение).

```
animals_data = dict(guinea_pigs = 6, dogs = 4, cats = 13, parrots  
= 3, fish = 7, snails = 8)
```

Задание 3.

Под конец учебного года преподаватель решил провести анализ оценок. У него есть список группы, из которой он хочет понять сколько людей получило оценку «отлично» (8-10), «хорошо» (6-7), «удовлетворительно» (4-5) и «неудовлетворительно» (0-3). Более того, он хочет получить список имен тех людей, которые отправились на пересдачу. После пересдач, все студенты получили оценку 4. Как тогда будет выглядеть окончательные список группы с оценками (нужно изменить список данный в начале)

Данные для 3 задания

```
group_marks = {'Олег Кардашьян': 9, 'Вячеслав Русиано': 3, 'Максим  
Максим': 10,  
               'Владислав Донтхертми': 7, 'Павел Хинкальный': 8,  
               'Ярослав Мутный': 2,  
               'Чак Нормис': 5, 'Александра Петрова': 7, 'Степан  
Кружок': 10,  
               'Мария Великая': 1, 'Джейсон Стэтхэм': 10, 'Светлана  
Ыыыыы': 8,  
               'Николай Колбасков': 2, 'Максим Солидный': 6, 'Валерий  
Мармеладзе': 5,  
               'Михаил Быкян': 8, 'Суэта Вечная': 4, 'Стас Борецкий': 0,  
               'Кеннедикт Мембербетч': 7,  
               'Артур Пирожков': 8, 'Элианора Андреева': 5, 'Михаил  
Круг': 3}
```