

# Словарь: основы

**Роман Булгаков**

Спикер курса

Skillbox

# Задача

## Входные данные:

- список из пар «имя — номер»
- имя для поиска

## Выходные данные:

- телефонный справочник
- номер телефона



# Словарь

phonebook\_dict

{'Ваня': 88006663636, 'Петя': 88005553535, 'Лена': 88007773737}

Ключ

Значение

Ключи **уникальны**, т. е. они не повторяются.

Списки и словари **не могут** быть ключами, зато **могут** быть значениями.

{'a': [1, 2, 3], 'b': {'b': 10 'd': 20}}

# Задача «Студент»

**Входные данные:** информация о студенте.

Пример ввода: Илья Иванов Москва МГУ 5 4 4 4 5

**Выходные данные:** словарь.



# Итоги урока

- `phonebook_dict = {}`  
`phonebook_dict = dict()`
- `phonebook_dict = {`
  - `'Ваня': 88006663636,`
  - `'Петя': 88005553535,`
  - `'Лена': 88007773737`
  - `}`
- `student = dict()`  
`student['Имя'] = 'Алексей'`
- `print(student['Имя'])`

# Методы словарей

**Роман Булгаков**

Спикер курса

Skillbox

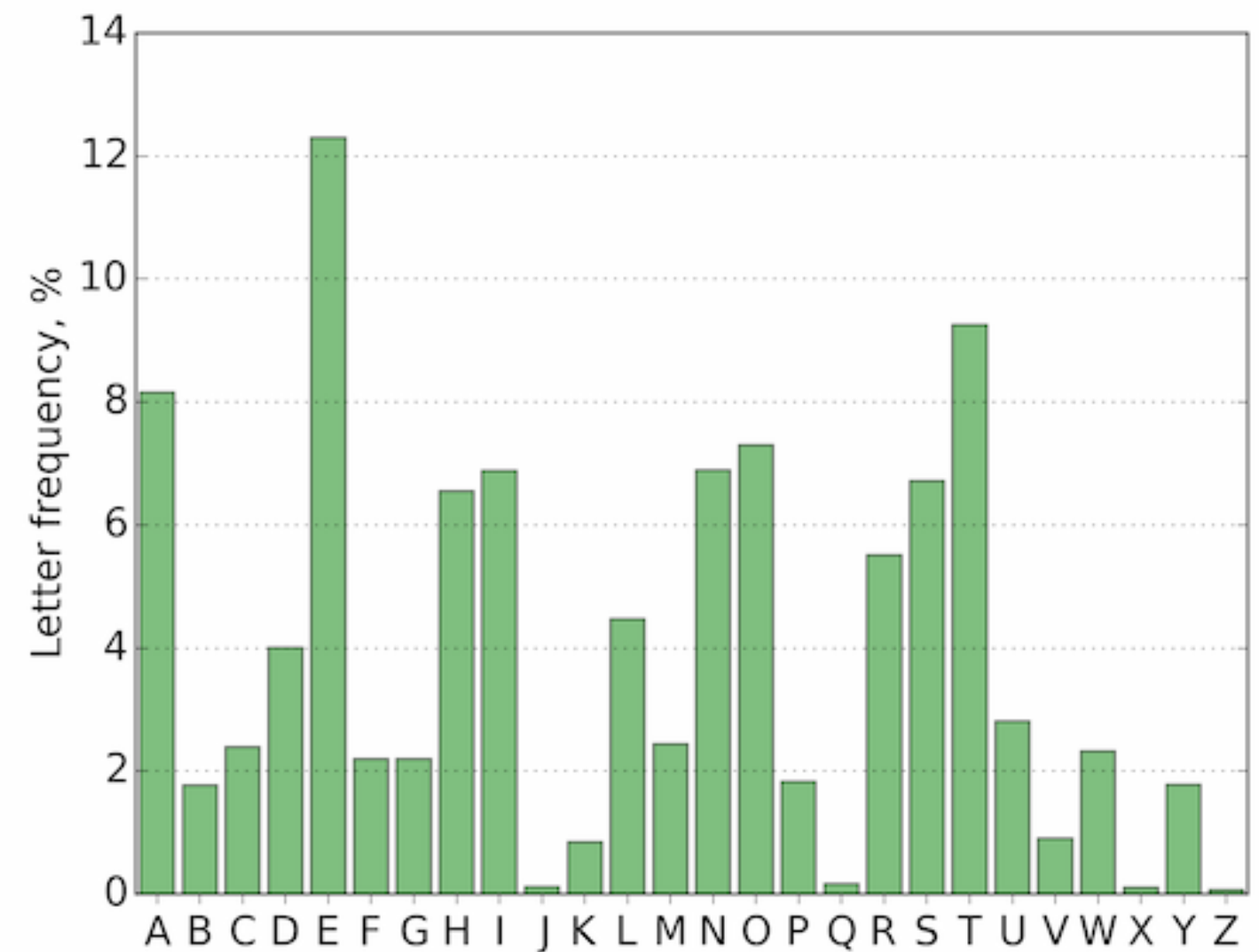
# Задача «Гистограмма частоты»

Входные данные:

- текст

Выходные данные:

- таблица вида «символ — частота»,  
отсортированная по алфавиту
- максимальная частота



# Задача «Контакты»

**Входные данные:** два словаря из пар «имя — номер».

**Выходные данные:** объединение двух словарей.





# Итоги урока

- `hist = {'a': 5, 'b': 10}`  
`hist.keys()` # `dict_keys(['a', 'b'])`
- `hist.values()` # `dict_values([5, 10])`
- `hist.update(other_hist)`
- `hist['c'] = hist.pop('b')` # `{'a': 5, 'c': 10}`
- `hist.get('a')` # `5`  
`hist.get('o')` # `None`

# Вложенные словари и значения по умолчанию в `get`

**Роман Булгаков**

Спикер курса

Skillbox

Вложенные словари и значения по умолчанию в get

# Задача «Структура данных»

```
{  
  "server": {  
    "host": "127.0.0.1",  
    "port": "10"  
  },  
  "configuration": {  
    "ssh": {  
      "access": "true",  
      "login": "Ivan",  
      "password": "qwerty"  
    }  
  }  
}
```

# Задача «Игроки»

## Условие задачи:

- словарь игроков
- игрок: имя, команда, статус

## Выходные данные:

- список игроков  
(team == 'A', status == 'Rest')

```
players_dict = {  
    1: {'name': 'Vanya', 'team': 'A', 'status': 'Rest'},  
    2: {'name': 'Lena', 'team': 'B', 'status': 'Training'},  
    3: {'name': 'Maxim', 'team': 'C', 'status': 'Travel'},  
    4: {'name': 'Egor', 'team': 'C', 'status': 'Rest'},  
    5: {'name': 'Andrei', 'team': 'A', 'status': 'Training'},  
    6: {'name': 'Sasha', 'team': 'A', 'status': 'Rest'},  
    7: {'name': 'Alina', 'team': 'B', 'status': 'Rest'},  
    8: {'name': 'Masha', 'team': 'C', 'status': 'Travel'}  
}
```

# Множества. Функция `set`

**Роман Булгаков**

Спикер курса

Skillbox

# Задача «Уникальные числа»

**Входные данные:** список  
из 10 случайных чисел.

**Выходные данные:** список  
из неповторяющихся чисел.



# Множества и словари

```
new_set = {1, 2, 3, 4} # множество
```

```
new_dict = {1: '66', 2: '77', 3: '88', 4: '99'} # словарь
```

## Сходства:

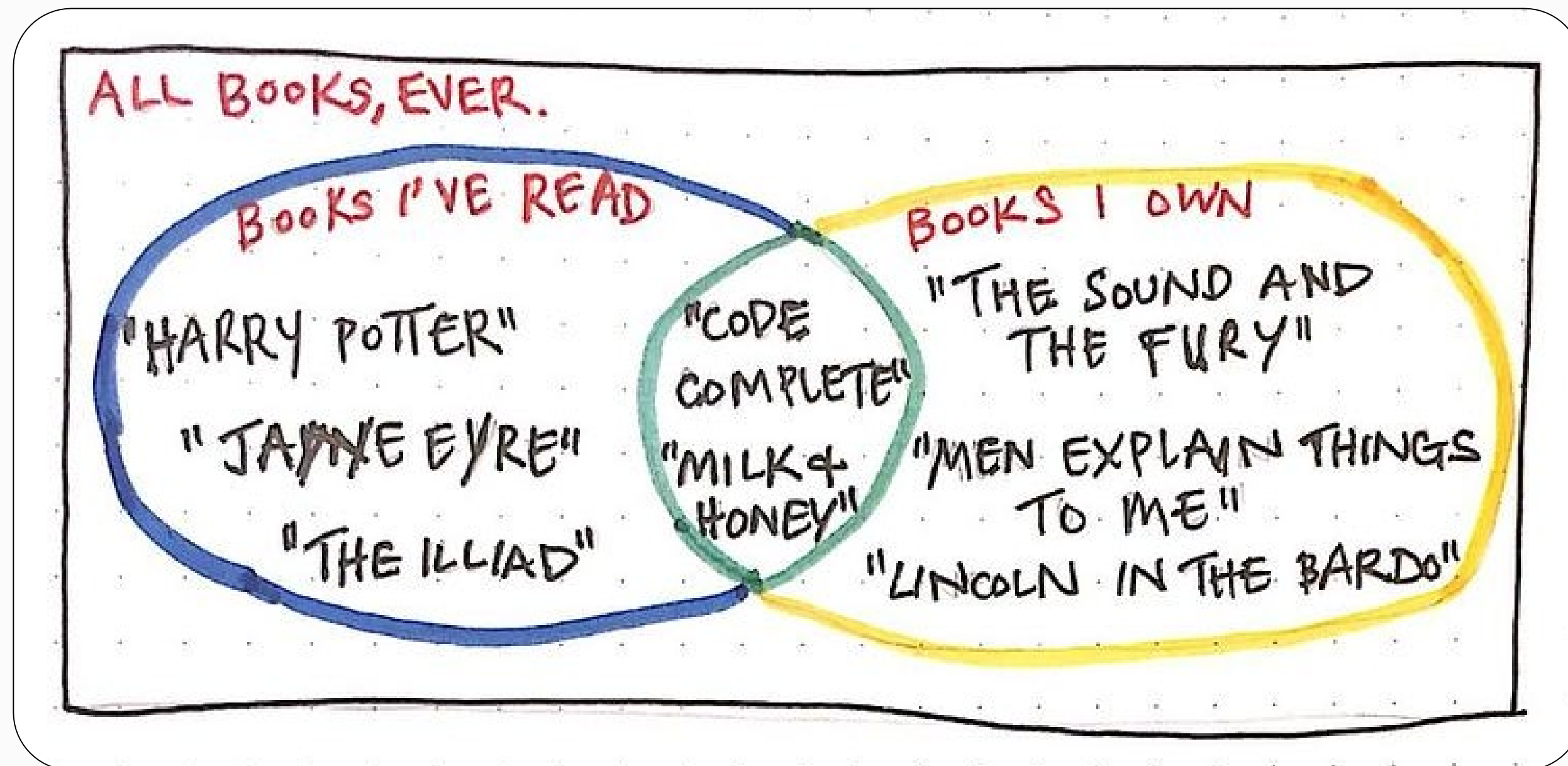
- ❶ Фигурные скобки
- ❷ Нет индексов
- ❸ Уникальные элементы

У множеств (как и у словарей) высокая скорость поиска, **гораздо выше, чем у списков.**

## Различия:

- ❶ Нет пар «ключ — значение»
- ❷ Другие **методы**

# Теория множеств





# Итоги урока

- `nums = [1, 1, 2, 2, 3]`  
`new_nums = set(nums) # {1, 2, 3}`
- `nums = {1, 2, 3, 4, 5}`  
`nums = {}` — словарь
- `nums_1 & nums_2` — пересечение (общие элементы)  
`nums_1 | nums_2` — объединение (все элементы)  
`nums_1 - nums_2` — разность (есть в 1-м, но нет во 2-м)
- `nums.add(10)`  
`nums.discard(3)`

# Генерация словарей

**Роман Булгаков**

Спикер курса

Skillbox

# Задача «Журнал посещений»

**Условие задачи:** список словарей  
(посетители — id, имя).

**Выходные данные:** список  
уникальных словарей.

```
data = [  
    {'id': 10, 'user': 'Bob'},  
    {'id': 11, 'user': 'Misha'},  
    {'id': 12, 'user': 'Anton'},  
    {'id': 10, 'user': 'Bob'},  
    {'id': 11, 'user': 'Misha'},  
]
```

# Итоги модуля

- `hist = dict()`  
`hist['a'] = 10`
- `hist.keys()` `hist.values()`
- `hist.update(new_hist)`  
`hist.pop('a')` `hist.get('a')` `hist.get('a', 0)`
- `nums = [1, 1, 2, 2, 3]`  
`new_nums = set(nums)`
- `{i_dict['id']: for i_dict in data}`

