Словари

Словарь — это структура данных, которая хранит пары «ключ — значение» (записи).
Пример: телефонная книга, где имя — это ключ, а номер — значение.

In [14]:
```
phonebook = {"Иван": "89160000000", "Ольга": "89251111111"}
print(phonebook)
```

```
{'Иван': '89160000000', 'Ольга': '89251111111'}
```

Если для получение элемента в массиве под известным индексом на требуется 1 операция (O(1)), то для получения индекса известного элемента нам требуется от n операций, где n-количество элементов в массиве (O(n) - поиск в массиве) .

В словаре поиск по ключу происходит очень быстро (O(1)), даже в большом объёме данных, за счёт того, что мы сразу знаем в какой участок памяти обращаться.

In [15]:
```
print(phonebook["Иван"])
```

```
89160000000
```

Другое название словарей - хеш-таблицы.
Словари работают по такому принципу: мы хешируем (преобразуем в число) ключ и вставляем в соответсвующее числу место в в памяти запись.

Хеширование

Хеш — это особая функция, которая преобразует данные (например, строку) в некоторое число фиксированной длины

В Python для получения хеша есть встроенная функция hash():

In [16]:
```
print(hash("Иван"))
print(hash(123))
```

```
1054698636696948047
123
```

А также библиотека hashlib, которая имеет функции с уже написанными популярными алгоритмами хеширования.

In [39]:
```
import hashlib
h=hashlib.sha256("Иван".encode('utf-8')).hexdigest()
print(h)
print(int(h,16))
```

```
cc0781950ffebec65a7f552e099f5111dc6526384615f13801e3f0d25b38e960
92285083384435061784370917080817754257140117280019516276158720982859724155232
```

Простую хеш функцию вы можете написать сами.
1.Выбрать параметры:

- Простое число prime — базовое число для вычислений (например, 31).
- Простое число mod — размер области хеш-значений (например, 101).

2. Инициализация:

- Задать начальное значение хеша равным 0.

3. Пройти по каждому символу по порядку.

- Получить числовой код символа (используется функция ord).
- Перемножить текущий хеш на prime и прибавить код символа.
- Взять остаток от деления на mod (операция % mod), чтобы ограничить размер числа и "перемешать" значения.

4. Результат:

- Полученное число — это итоговый хеш ключа.

In [43]:
```
def simple_hash(s, prime=31, mod=101):
    ...

print(simple_hash("Иван"))
print(simple_hash("Ира"))
print(simple_hash("Вася"))
```

```
22
12
83
```

Рассмотрим алгоритм использования хеширования в словарях:

1. Когда появляется новая запись, вычисляется хеш (число) ключа.

2. Это число превращается в индекс массива (например, делением с остатком):

`index=hash(key)%размер_массива`

3. В массиве по этому индексу проверяется, есть ли уже такой ключ.

4. Если нет — записывается новая пара «ключ—значение»; если да — значение обновляется.

НО, если хеш у двух ключей совпадет, то произойдет коллизия
to be continued

Задание

1. Написать свою хеш функцию

2. Написать свой словарь