Алгоритмы и структуры данных на Python

Урок 3

Хеш-функции

Хеш-функции, хеши, хеш-таблицы

План урока

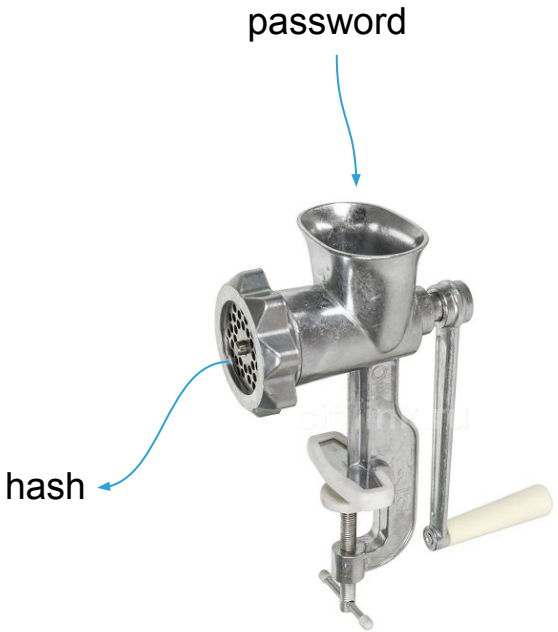
- Что такое хеши и где они используются
- Виды хеш-функций
- Хеш-таблицы

Что такое хеш?

- Трансформация данных любого типа и длины в битовую строку.
- Американское блюдо из мяса, картофеля и лука (to hash - рубить).

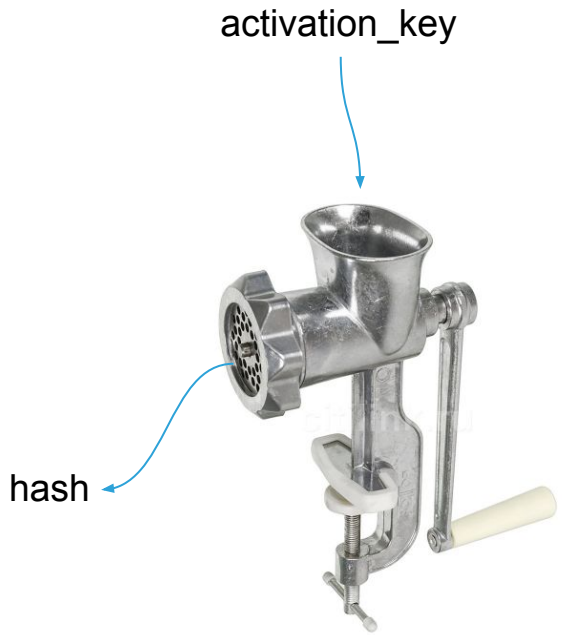
Пример хеша

1) Вы создали аккаунт в приложении.
2) Ваш пароль хешируется за счет хеш-функции и сохраняется в БД.
3) Если вы потом пытаетесь залогиниться, то введенный вами пароль пропускается через хеш-функцию и сравнивается с хешем правильного пароля, сохраненным в БД.
4) При совпадении хешей пользователь получает доступ к ресурсу, иначе пароль будет запрошен повторно.
5) Шаги 3 и 4 проходят каждый раз, когда вы будете проходить авторизацию.



Еще пример хеша

Все то же самое, только вместо пароля код активации

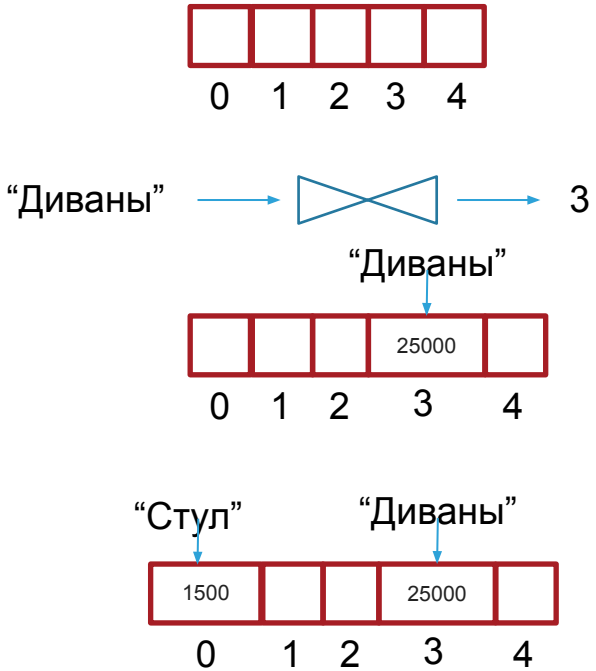


Хеш-функции

Название хеш-функции	Длина строки (дайджеста) в битах
md5	128
sha1	160
sha224	224
sha256	256
sha384	384
sha512	512

«Соленые» хеши

Хеш-таблицы


0 1 2 3 4
“Диваны”
3
“Диваны”
25000
0 1 2 3 4
“Стул”
“Диваны”
1500
25000
0 1 2 3 4

Коллизии хешей

Одинаковые хеши для объектов!

Например, коллизии хешей выявлены для md5 и sha1!

Что делать?

«Солите» хеши!

Подведем итоги

- Словарь в Python – фундаментальный тип данных, реализованный в виде хеш-таблицы, с открытой адресацией и встроенным методом разрешения коллизий.
- Ключ – обязательно хешируемый объект, т.е. у него должен существовать метод __hash__.
- Словари, благодаря константной сложности, обеспечивает быстрый поиск по ключу.
- Словари требуют больше памяти, т.к. хеш-таблица должна быть достаточно большой для эффективного ее использования.