

# Модуль secrets в Python, генерация паролей и токенов



avito.ru РЕКЛАМА · 16+

Больше информации на сайте рекламодателя

Подробнее

[Стандартная библиотека Python3.](#) / Модуль secrets в Python, генерация паролей и токенов

## Генерация паролей, токенов и криптографически сильных случайных чисел

[Модуль secrets](#) используется для генерации криптографически сильных случайных чисел, подходящих для управления такими данными, как пароли, аутентификация учетной записи, токены безопасности и связанные секреты.

В частности, [secrets](#) следует использовать по умолчанию вместо генератора псевдослучайных чисел [модуля random](#), который в свою очередь предназначен для моделирования и симуляции, а не для защиты или криптографии.

## Сколько байтов следует использовать токенам?

Чтобы обезопасить себя от атак методом "перебора", токены должны иметь достаточную случайность. С 2015 года считается, что 32 байта (256 бит) является достаточно сильным токеном для типичного варианта использования.

Что бы указать собственную длину токена, то можно явно передать [функциям token \\*](#) аргумент nbytes, который имеет [тип int](#). Этот аргумент принимается за количество байт, которое будет использовано при создании токена.

В противном случае, если аргумент nbytes не предоставлен или равен None, [функциям token \\*](#) будут использовать разумное значение по умолчанию.

### Примечания:

- Значение по умолчанию может быть изменено в любое время, в том числе во время выпусков обновлений.
- Приложения не должны хранить пароли в восстанавливаемом формате, будь то простой или зашифрованный текст. Они должны быть хешированы с использованием криптографически сильной односторонней (необратимой) хеш-функции.

### Примеры использования:

Создание восьмибуквенного буквенно-цифрового пароля:

```
>>> import string
>>> import secrets
>>> alphabet = string.ascii_letters + string.digits
>>> password = ''.join(secrets.choice(alphabet) for i in range(8))
>>> password
# '9vfMa4Wz'
```

Создание десятисимвольного буквенно-цифрового пароля, содержащего как минимум один символ нижнего регистра, как минимум один символ верхнего регистра и как минимум три цифры:

```
import string
import secrets
alphabet = string.ascii_letters + string.digits
while True:
    password = ''.join(secrets.choice(alphabet) for i in range(10))
    if (any(c.islower() for c in password)
```

```
        and any(c.isupper() for c in password)
        and sum(c.isdigit() for c in password) >= 3):
    break

print(password)
# 'oz2B4b3d09'
```

Создание кодовой фразы в стиле "XKCD":

```
import secrets
# В стандартных системах Linux используйте удобный файл словаря.
# Другие платформы, должны предоставить свой собственный список слов.
with open('/usr/share/dict/words') as f:
    words = [word.strip() for word in f]
    password = ' '.join(secrets.choice(words) for i in range(4))

print(password)
# "Np escape guidebook's outperforms"
```

Создание временного, трудно угадываемого URL-адреса, содержащего маркер безопасности и подходящего для приложений восстановления пароля:

```
>>> import secrets
>>> url = 'https://mydomain.com/reset=' + secrets.token_urlsafe()
>>> url
# 'https://mydomain.com/reset=l6nrqcxpqgoogE0TBjAQNHrrgm-VycP3gcbP4eqvFsM'
```

Содержание раздела:
<ul style="list-style-type: none"><li><a href="#">КРАТКИЙ ОБЗОР МАТЕРИАЛА.</a></li><li><a href="#">Случайные числа модуля secrets</a></li><li><a href="#">Генерация токенов модулем secrets</a></li></ul>

ХОЧУ ПОМОЧЬ  
ПРОЕКТУ

РЕКЛАМА



Вверх