


ХОЧУ ПОМОЧЬ ПРОЕКТУ



skillbox.ru

Станьте «DevOps-инженером» всего за 7 месяцев!
Обучим DevOps с гарантией трудоустройства. Трудоустроим – или вернем деньги.
[Узнать больше](#)

gb.ru

Бесплатный практикум для детей: Python и анимация

5,0 ★ Рейтинг организации ⓘ

Живой практикум для детей по 2D-анимации и Python. Количество мест ограничено.

Бесплатный мастер-класс >

3 подарка участникам >

[Узнать больше](#)

Модуль chardet в Python, определение кодировки

1 символ Python

представляем слова и буквы, которые видим на экране компьютера. Но компьютеры не работают с целло с битами и байтами. Каждый фрагмент текста, который выводится на экране, на самом деле состоит из символов. Существует множество различных кодировок, некоторые из которых относятся к разным языкам, таких как русский, китайский или английский, а другие могут использоваться для кодировки символов обеспечивает соответствие между тем, что мы видим на экране, и тем, что хранится в памяти и на диске.

Модуль chardet является детектором кодировки текста и является портом кода автоопределения в Mozilla. Этот модуль определяет кодировку символов, если вдруг на экране появятся "кракозябры".

Модуль chardet поддерживает и определяет русские кодировки: KOI8-R, MacCyrillic, IBM855, IBM866, ISO-8859-5, и другие.

установка модуля snardet в виртуальное окружение.

```
# создаем виртуальное окружение, если нет
$ python3 -m venv .venv --prompt VirtualEnv
# активируем виртуальное окружение
$ source .venv/bin/activate
# ставим модуль chardet
(VirtualEnv):~$ python -m pip install -U chardet
```

Примеры автоматического определения кодировки символов:

Самый простой способ автоматически определить кодировку - это использовать функцию обнаружения detect() модуля chardet.

```
>>> import urllib.request, chardet
>>> rawdata = urllib.request.urlopen('http://yandex.ru/').read()
>>> chardet.detect(rawdata)
# {'encoding': 'utf-8', 'confidence': 0.99, 'language': ''}
>>> rawdata = urllib.request.urlopen('https://www.zeit.de/index').read()
>>> chardet.detect(rawdata)
# {'encoding': 'utf-8', 'confidence': 0.99, 'language': ''}
```

Расширенное использование модуля chardet.

Если имеется большой объем текста/данных, то можно вызывать обнаружение кодировки постепенно. Как только модуль будет достаточно уверен в своих результатах, он остановится.

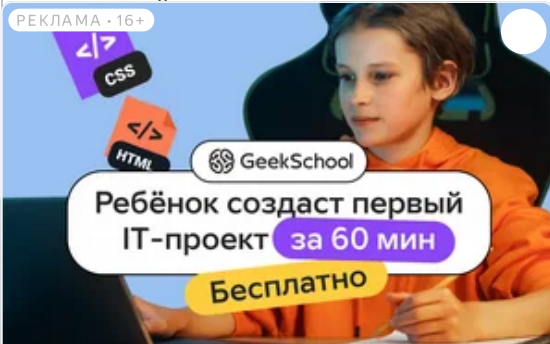
Для такого поведения необходимо создать объект UniversalDetector(), затем повторно вызывать его метод подачи .feed() с каждым блоком текста. Если созданный детектор достигнет минимального порога достоверности, он установит для Detector.done значение True.

В конце работы детектора необходимо вызвать Detector.close(), который выполнит некоторые окончательные вычисления в случае, если детектор не достиг минимального порога достоверности.

```
import urllib.request
from chardet.universaldetector import UniversalDetector

usock = urllib.request.urlopen('https://www.zeit.de/index')
# Вверх
detector = UniversalDetector()
```

```
for line in usock.readlines():  
    # скармливаем детектору строки  
    detector.feed(line)  
    if detector.done:
```



gb.ru

Бесплатный практикум для детей: Python и анимация

5,0 ★ Рейтинг организации ⓘ

Живой практикум для детей по 2D-анимации и Python. Количество мест ограничено.

Бесплатный мастер-класс >

3 подарка участникам >

Узнать больше

```
        if detector.done: break  
    detector.close()  
    print(detector.result)
```

...
... цикл

```
...ence': 0.99, 'language': ''}
```

ировки нескольких файлов.

ых файлов, их необходимо [открывать в режиме чтения](#) байтов: more='rb'

```
import UniversalDetector
```

```
...):  
...id='')
```

...едного

```
...rb'):
```