

14 библиотек и модулей Python, которые должен знать каждый разработчик

27.04.2023

Библиотеки и модули облегчают жизнь программиста. При работе над проектами вы можете столкнуться со сценариями, которые вы не сможете решить стандартным кодированием на языке программирования. Для решения этих проблем нам понадобятся библиотеки и модули. К счастью, Python поддерживает множество модулей и библиотек. В Python есть встроенные модули, а также сторонние библиотеки и модули для разработки. Мы увидим как встроенные, так и сторонние модули, которые очень полезны для проектов Python. Давайте сначала рассмотрим встроенные модули. Python поставляется с большим количеством встроенных модулей для различных случаев использования. Мы будем изучать модули по очереди в зависимости от их использования.

Collections

В Python есть различные типы коллекций для хранения коллекции данных. Например, `tuple`, `list`, `dict` и т.д. являются одними из встроенных коллекций Python. Модуль `collections` предоставляет дополнительные возможности для встроенных коллекций. Если взять коллекцию данных `deque` из модуля коллекций, то она больше похожа на список Python. Но мы можем выталкивать и разворачивать элементы с обеих сторон. Это быстрее, чем список. Вы можете использовать `deque` в зависимости от ваших потребностей. Давайте посмотрим на реальное кодирование с использованием коллекции данных `collections.deque`.

```
import collections
nums = [1, 2, 3]
# creating deque collection from the list
deque = collections.deque(nums)
```

```

print(deque)

# adding an element at the end
deque.append(4)

print(deque)

# adding element at the starting
deque.appendleft(0)

print(deque)

# removing the element at the end
deque.pop()

print(deque)

# removing element at the starting
deque.popleft()
print(deque)

```

Запустите приведенный выше код и посмотрите результаты. В модуле коллекций у нас есть и другие коллекции данных.

Вот некоторые из них:

Counter	Возвращает dict, содержащий частоту элементов из списка. Это подкласс класса dict.
UserList	Используется для быстрого подкласса списка.
UserDict	Используется для быстрого подкласса dict.
UserString	Используется для быстрого подкласса класса str.

Перейдите к документации модуля collections, чтобы изучить все коллекции и методы данных. Краткое примечание:- Используйте встроенный метод dir(object) в Python, чтобы увидеть все методы объекта.

CSV

Для хранения табличных данных мы можем использовать файлы CSV (значения, разделенные запятыми). Наиболее часто используемый формат для импорта и экспорта данных из электронных таблиц и баз данных. Python поставляется с модулем CSV для работы с файлами CSV. Рассмотрим один пример чтения данных из CSV-файла. Создайте файл с именем `sample.csv` на вашем ноутбуке и вставьте в него следующие данные.

```
Name,Age,Graduation Year
```

```
Hafeez,21,2021
```

```
Aslan,23,2019
```

```
Rambabu,21,2021
```

В модуле CSV у нас есть методы для чтения и записи. Мы рассмотрим, как читать данные из CSV-файлов с помощью модуля CSV.

```
import csv

with open('sample.csv') as file:
    # creating the reader
    reader = csv.reader(file)
    # reading line by line using loop
    for row in reader:
        # row is a list containing elements from the CSV file
        # joining the list using join(list) method
        print(','.join(row))
```

Запустите приведенный выше код, чтобы увидеть результаты. У нас также будет объект `csv.writer()` для записи данных в CSV-файл. Поиграйте с другими методами самостоятельно, используя встроенные методы `dir()` и `help()`. У нас есть еще один модуль под названием JSON, который используется для работы с файлами JSON. Это также встроенный модуль.

Random

В Python есть модуль `random`, который позволяет генерировать данные случайным образом. Мы можем произвести что угодно случайным образом, используя различные способы модуля `random`. Вы можете использовать этот модуль в таких приложениях, как крестики-нолики, игра в кости и т.д., Рассмотрим простую программу для генерации случайных целых чисел из заданного диапазона.

```
import random

# generating a random number from the range 1-100
print(random.randint(1, 100))
```

Другие методы модуля `random` можно проверить с помощью методов `dir()` и `help()`. Давайте напишем небольшую и простую игру с использованием модуля `random`. Мы можем назвать ее “Игра в угадывание чисел”.

Что такое игра “Угадывание чисел”?

Программа генерирует случайное число в диапазоне от 1 до 100. Пользователь будет угадывать число до тех пор, пока оно не совпадет со случайным числом, сгенерированным программой. Каждый раз будет выводиться информация о том, меньше ли число пользователя, чем случайное число, или больше, чем случайное число. Затем в исходном коде будет выводиться количество угаданных чисел.

Смотрите приведенный ниже код для вышеуказанной программы.

```
# importing random module
import random

# generating random number
random_number = random.randint(1, 100)

# initializing no. of guess to 0
guess_count = 0
```

```
# running loop until user guess the random number
while True:
    # getting user input

    user_guessed_number = int(input("Enter a number in the
range of 1-100:- "))

    # checking for the equality
    if user_guessed_number == random_number:
        print(f"You have guessed the number in {guess_count}
guesses")
        # breaking the loop
        break
    elif user_guessed_number < random_number:
        print("Your number is low")
    elif user_guessed_number > random_number:
        print("Your number is high")

# incrementing the guess count
guess_count += 1
```

Tkinter

Tkinter – это встроенный модуль для разработки приложений с графическим интерфейсом пользователя (GUI). Он удобен для начинающих. Мы можем разрабатывать GUI-приложения, такие как калькуляторы, системы входа в систему, текстовые редакторы и т.д.; существует множество ресурсов, чтобы научиться разрабатывать GUI с помощью Tkinter. Лучшая поддержка – следовать официальной документации. Чтобы начать работу с Tkinter, перейдите к документации и начните создавать красивые GUI-приложения.

Модули сторонних производителей

Requests

Модуль Requests используется для отправки всех видов HTTP-запросов на сервер. Он позволяет отправлять запросы HTTP/1.1.

Мы также можем добавлять заголовки, данные и другие вещи, используя словари Python. Поскольку это сторонний модуль, нам необходимо его установить. Выполните следующую команду в терминале или командной строке, чтобы установить модуль requests.

```
pip install requests
```

Работать с модулем запросов очень просто. Мы можем начать работать с запросами без каких-либо предварительных знаний. Давайте посмотрим, как отправить запрос get и что он возвращает.

```
import requests
```

```
# sending a get request
```

```
request = requests.get("https://www.google.com/")
```

```
#
```

```
print(request.status_code)
```

```
print(request.url)
```

```
print(request.request)
```

Приведенный выше код выведет код_состояния, URL и метод запроса (GET, POST). Вы также получите источник URL. Вы можете получить доступ к нему с помощью байта request.content. Перейдите к документации модуля requests и узнайте больше.

BeautifulSoup4

Библиотека BeautifulSoup используется для парсинга. Это удобный модуль для работы. Даже новички могут начать работать с ним, используя документацию. Посмотрите пример кода для парсинга информации об отчетах клиентов. Вы можете установить BeautifulSoup, набрав следующую команду в терминале/командной строке.

```
pip install beautifulsoup4
```

И, простая программа для вашего первого парсинга.

```
## Scrping the ConsumerReport products list using
BeautifulSoup

## importing bs4, requests modules
import bs4
import requests

## initializing url
url =
"https://www.consumerreports.org/cro/a-to-z-index/products/index.htm"

## getting the reponse from the page using get method of
requests module
page = requests.get(url)

## storing the content of the page in a variable
html = page.content

## creating BeautifulSoup object
soup = bs4.BeautifulSoup(html, "lxml")

## see the class or id of the tag which contains names and
links
div_class = "crux-body-copy"

## getting all the divs using find_all method
div_tags = soup.find_all("div", class_=div_class) ## finding
divs whichs has mentioned class

## we will see all the tags with a tags which has name and
link inside the div
for tag in div_tags:
    print(tag)
```

Запустите приведенный выше код, чтобы увидеть магию парсинга.
Вы можете попробовать и другие фреймворки для парсинга.

Наука о данных и машинное обучение

Есть несколько библиотек, специально созданных для науки о данных и машинного обучения. Все они разработаны на языке С. Они работают молниеносно.

Numpy

Numpy используется для научных вычислений. Он позволяет работать с многомерными массивами. В Python реализация массивов отсутствует. В основном разработчики используют numpy в своих проектах машинного обучения. Это простая в освоении библиотека с открытым исходным кодом. Почти каждый инженер машинного обучения или специалист по исследованию данных использует этот модуль для сложных математических вычислений. Выполните следующую команду для установки модуля numpy.

```
pip install numpy
```

Pandas

Pandas – это модуль анализа данных. Мы можем наиболее эффективно фильтровать данные, используя библиотеку pandas. Она предлагает различные типы структур данных, с которыми удобно работать. Она также обеспечивает работу с файлами различных форматов. Установите модуль с помощью следующей команды.

```
pip install pandas
```

Matplotlib

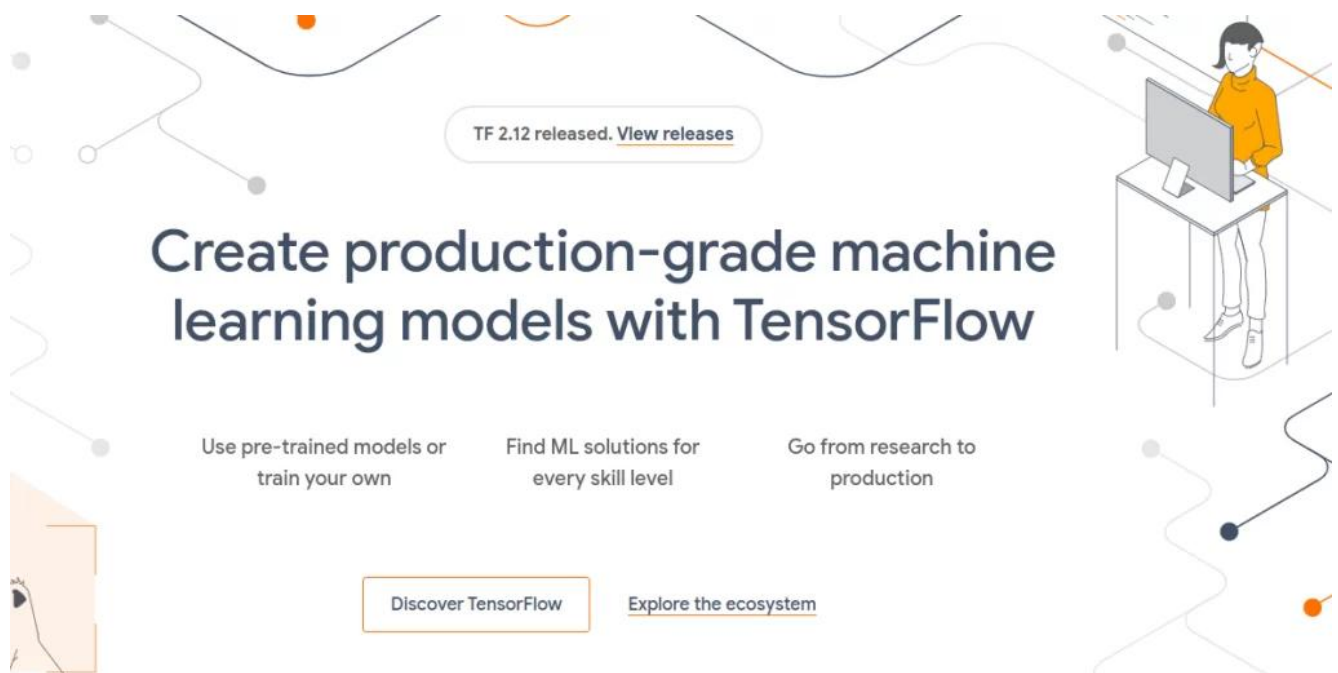
Matplotlib – это библиотека для построения двумерных графиков. С помощью Matplotlib можно визуализировать данные. Мы можем генерировать изображения рисунков в различных форматах. Мы строим различные типы диаграмм, такие как гистограммы, диаграммы ошибок, гистограммы, диаграммы рассеяния и т.д. Вы можете установить matplotlib с помощью следующей команды.


```
pip install matplotlib
```

Краткое примечание:– Вы можете установить Anaconda, чтобы получить все библиотеки и модули, необходимые для Data Science.

TensorFlow

TensorFlow – это сквозная библиотека машинного обучения на языке Python, разработанная компанией Google. С помощью TensorFlow можно запускать стандартные алгоритмы машинного обучения, создавать нейронные сети и выполнять численные вычисления. Однако для работы с TensorFlow необходимо глубокое понимание векторов и алгебры.



Как использовать TensorFlow

Создайте проект, перейдите в него, а затем установите TensorFlow с помощью этой команды;

```
pip install tensorflow-cpu
```

TensorFlow довольно популярен по этим причинам;

- **Гибкость:** TensorFlow может выполнять различные задачи машинного обучения, такие как распознавание речи,

распознавание изображений, обработка естественного языка и т.д. и т.п.

- **Масштабируемость:** Вы можете использовать TensorFlow для обучения и развертывания моделей Machine Learning на различных аппаратных системах, таких как TPU, GPU и CPU.
- **Большое сообщество:** TensorFlow привлек большую пользовательскую базу с более чем 174 тыс. звезд и 88 тыс. форков на GitHub.
- **Расширяемость:** Вы можете настроить TensorFlow в соответствии с конкретными потребностями вашего проекта.

TensorFlow – это бесплатный проект с открытым исходным кодом, исходный код которого находится на GitHub.

Keras

Keras – это интерфейс прикладного программирования глубокого обучения (API), построенный на платформе машинного обучения TensorFlow. Глубокое обучение является подразделом машинного обучения. Keras был создан для быстрого проведения экспериментов, позволяя пользователям быстро перейти от идеи к продукту.



Simple. Flexible. Powerful.

В то время как TensorFlow выступает в качестве инфраструктурного слоя для дифференцируемого программирования, Keras предлагает основные строительные блоки и абстракции для создания и внедрения решений машинного обучения. С помощью Keras инженеры получают преимущества кросс-платформенных возможностей и масштабируемости TensorFlow.

Чтобы начать использовать Keras, установите его с помощью pip:

```
pip install keras
```

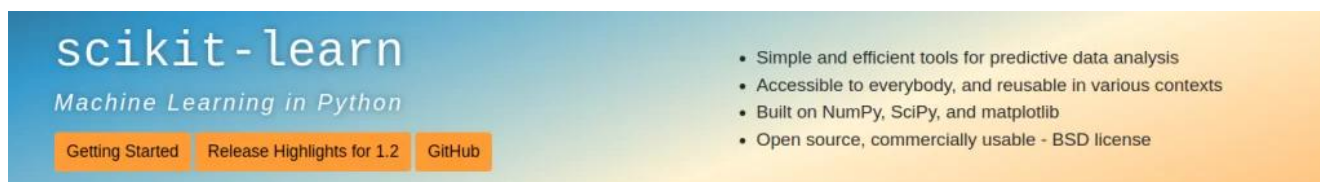
Почему именно Keras?

- **Гибкость:** Keras был создан по принципу “постепенного раскрытия сложности”. Таким образом, вы можете создавать различные нейронные сети, поскольку он поддерживает различные оптимизаторы, функции активации и типы слоев.
- **Простота использования:** Keras создан, чтобы быть простым, но не упрощенным. Как инженеру, Keras позволит вам сосредоточиться на важных решениях, освободив вас от когнитивной нагрузки.
- **Большое сообщество:** Keras имеет большое начинающее сообщество с более чем 57,9 тыс. звезд на GitHub.
- **Масштабируемость:** Вы можете использовать Keras для экспериментов. Однако вы также можете продолжать использовать этот API по мере роста вашего приложения.

Keras – это бесплатная Python-библиотека глубокого обучения с открытым исходным кодом.

Scikitt-Learn

Scikitt-Learn – это модуль машинного обучения Python, построенный на SciPy. Модуль предлагает эффективные и простые инструменты, которые инженеры могут использовать для анализа прогностических данных.



Scikitt-Learn имеет множество вариантов использования, таких как классификация, кластеризация, регрессия, предварительная обработка и другие.

В функции Классификация вы можете использовать этот модуль для определения категории объекта. Эта функция полезна при распознавании изображений и обнаружении спама.

Вы можете установить Scikit-Learn в папку вашего проекта с помощью pip следующим образом:

```
pip install scikit-learn
```

После установки вы можете проверить, хорошо ли он настроен, импортировав его в ваши файлы следующим образом:

```
import sklearn
```

Почему именно Scikit-Learn?

- **Различные алгоритмы машинного обучения:** Вы можете использовать различные алгоритмы, от деревьев решений, нейронных сетей и случайных лесов до машин опорных векторов.
- **Визуализация моделей:** Модуль имеет различные инструменты для визуализации моделей машинного обучения.
- **Оценка моделей:** С помощью этого модуля вы можете проводить перекрестную валидацию и оценивать метрики.
- **Предварительная обработка данных:** Вы можете выполнять уменьшение размерности, нормализацию и выбор признаков с помощью инструмента предварительной обработки данных этого модуля.

Scikit-Learn имеет открытый исходный код и может использоваться в коммерческих целях по лицензии BSD.

Веб-фреймворки

В Python можно найти множество веб-фреймворков. Мы рассмотрим два фреймворка, которые широко используются разработчиками. Эти два фреймворка – Django и Flask.

Django

Django – это веб-фреймворк с открытым исходным кодом, разработанный на языке Python. С помощью Django удобно создавать сайты. С помощью этого фреймворка можно создать любой сайт. Одними из самых популярных сайтов, созданных с помощью Django, являются Instagram, bitbucket, Disqus, Mozilla Firefox и т.д.,

- Мы можем быстро создавать сложные сайты, используя возможности Django.
- Django уже выполняет множество задач, необходимых для веб-разработки.
- Он также обеспечивает защиту от атак SQL Injection, межсайтового скриптинга, подделки межсайтовых запросов и clickjacking.
- Мы можем создать любой сайт, от системы управления контентом до социальных сайтов.

Документация Django однозначна. Вы должны быть знакомы с Python для Django. Но не волнуйтесь, если это не так. Изучить Django очень просто.

Flask

Flask – это микро веб-фреймворк, разработанный на Python. Он более питоничен, чем Django. По нему есть отличная документация. Он использует шаблонизатор Jinja. Создавать большие сайты на Flask очень сложно. Большинство функций, таких как маршрутизация URL, диспетчеризация запросов, безопасные cookies, сессии и т.д., присутствуют как в Django, так и во Flask. Выбирайте фреймворк в зависимости от сложности вашего сайта. Django набирает популярность среди разработчиков. Это самый используемый фреймворк для веб-разработки на Python.

Заключение

Надеюсь, вы узнали о различных модулях, библиотеках и фреймворках для Python. Каждый человек когда-то становится новичком. Что бы вы ни хотели начать, сначала обратитесь к документации и начните изучать ее. Если вы не можете понять документацию, найдите курсы на этих образовательных сайтах.