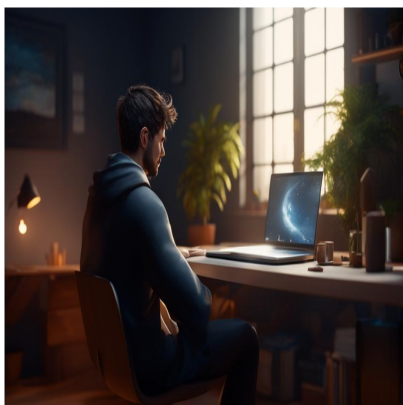




5 крутых вещей, которые можно сделать с помощью Python

Описание

Python – один из самых любимых языков программирования. И почему бы и нет, вы можете создавать от простых скриптов обслуживания до сложных приложений машинного обучения. С помощью **Python** можно делать множество крутых вещей, которые вам понравится изучать.

Введение

Python – очень популярный язык среди разработчиков. На нем легко и весело писать скрипты для автоматизации и создания различных вещей. К числу распространенных вариантов использования относятся:

- Создание ботов
- Скраппинг веб-сайтов
- Машинное обучение, визуализация и анализ данных
- Веб-разработка с использованием таких фреймворков, как Django и Flask
- Разработка игр с помощью Pygame
- Мобильные приложения с помощью фреймворков, таких как Kivy

В этой статье я постараюсь на примерах охватить несколько областей и показать вам некоторые интересные вещи, которые можно делать с помощью Python. Если вы не знаете Python, я рекомендую вам его выучить! **Давайте начнем!**

Для веб-разработки

Python очень хорошо поддерживает веб-разработку благодаря таким фреймворкам, как Django, Flask и другим. Он может использоваться для создания серверных веб-приложений и может быть интегрирован с любым фронтендом. Как правило, разработчики используют JavaScript во фронтенде и python для поддержки операций на стороне сервера. Python не используется непосредственно в браузерах. Django – один из самых популярных веб-фреймворков на языке python. Эти фреймворки предоставляют пакет, в котором есть определенная структура, поддерживается взаимодействие с базами данных; все это настраивается с помощью минимальной команды setup. Если вам нужно что-то минимальное для начала – я рекомендую Flask! Кроме них, в Python есть большое количество библиотек для веб-разработки. Некоторые популярные из них –

- Requests
- Pillow
- PyMongo

Некоторые **ресурсы** для начала веб-разработки на Python –

- Образовательный блог
- Веб-приложение с нуля

Пример - доступ к файловой системе компьютера с мобильного

Вы можете получить доступ к файловой системе, запустив на своей машине файловый сервер. Перейдите в нужный каталог, к которому вы хотите получить доступ, и выполните следующую команду –

```
# python version >= 3.X
python3 -m http.server
```

```
# If Python version >= 2.X and < 3.X
python -m SimpleHTTPServer
#default port: 8000
```

Это запускает файловый сервер, доступ к которому можно получить в той же сети. Чтобы получить доступ к файлам на мобильном телефоне, просто подключитесь к той же сети (Wi-Fi или используйте точку доступа телефона на ноутбуке). Теперь в браузере телефона откройте –

<your-computer-ip>:port

Проверьте свой IP, выполнив команду `ifconfig`. Проверьте свой локальный IP (должен начинаться с 192.168....) Предположим, ваш IP – 192.168.43.155, и вы используете порт по умолчанию. Тогда вам следует открыть –

192.168.43.155:8000 на мобильном. Вы увидите текущую директорию .

Автоматизация и сценарии

Если вы инженер, то, скорее всего, вам будет лень и вы захотите автоматизировать почти все, что только можно, верно? Не волнуйтесь, python поможет вам.

Существует масса вещей, которые можно автоматизировать всего 4-5 строками кода: от установки заданий cron и напоминаний до загрузки любимых видео на youtube – все это можно сделать с помощью пары строк на python.

Пример - Преобразование CSV в JSON

Вы можете преобразовать CSV-файл в JSON с помощью всего одной команды на языке python! Попробуйте прямо сейчас –

```
python -c "import csv,json;print json.dumps(list(csv.reader(open('your_csv_file.csv'))))"
```

Замените его на файл filename.csv, и вы получите результат в формате JSON! Легко, правда?

Строительные игры

Python поддерживает разработку игр. Его библиотека Pygame очень полезна. С ее помощью можно создавать художественные, музыкальные, звуковые, видео- и мультимедийные проекты. Вы даже можете создавать кроссплатформенные игры с помощью Kivy, которая работает на Windows, Mac, Linux, Android и iOS.

Пример - Виселица в терминале

Вот простая программа на python, которая позволяет играть в палач в терминале. Код можно значительно сократить, и я оставляю это на ваше усмотрение!

```
# hangman.py
# importing the time module
import time
import random

turns = 10

print "Hello, Let's play hangman! You will have " + str(turns) + " turns!"

print ""

# delay
time.sleep(0.5)

# set of words to guess from
wordList = ["geekflare", "awesome", "python", "magic"]
word = random.choice(wordList)

guesses = ''

# loop till no turns are remaining
while turns > 0:
    wrong = 0

    for char in word:
        if char in guesses:
            print char,
        else:
            print "_",
            wrong += 1

    print("\n")

    if wrong == 0:
        print "You won :)"

        break

    print

    guess = ''
    if len(guess) < 1:
        guess = raw_input("Guess a character or enter the correct word: ")[0]

    guesses += guess

    if guess not in word:
        turns -= 1
```

```
print "Wrong"
print "You have", + turns, ' turns left!'
if turns == 0:
    print "You Lose :("
```

Выходные данные будут выглядеть примерно так -

```
Hello, Let's play hangman! You will have 10 turns!

_ _ _ _ _

Guess a character or enter the correct word: a
_ a _ _ _

Guess a character or enter the correct word: s
Wrong
You have 9 turns left!
_ a _ _ _

Guess a character or enter the correct word: t
Wrong
You have 8 turns left!
_ a _ _ _

Guess a character or enter the correct word: m
m a _ _ _

Guess a character or enter the correct word: i
m a _ i _

Guess a character or enter the correct word: g
m a g i _

Guess a character or enter the correct word: c
m a g i c
```

Веб-скраппинг

Каждый день вы видите множество данных на разных сайтах. Подумайте, как было бы здорово, если бы вы могли легко получить доступ к этим данным. Это и есть веб-скреппинг, а python делает его еще проще благодаря своей удивительной поддержке и библиотекам. Данные в Интернете неструктурированы, и python обеспечивает простой способ анализа и потребления этих данных и даже дальнейшего анализа и операций. Позвольте мне показать вам пример того, как вы можете взять значения валют с сайта – x-rates.com.

Пример - Получение значения валюты в сравнении с USD

Давайте воспользуемся скраппингом в python, чтобы получить значения валют –

```
import requests
from bs4 import BeautifulSoup

URL = "https://www.x-rates.com/table/?from=USD&amount=1"
r = requests.get(URL)

soup = BeautifulSoup(r.content, 'html.parser')
ratelist = soup.findAll("table", {"class": "ratesTable"})[0].findAll("tbody")

for tableVal in ratelist:
    trList = tableVal.findAll('tr')
    for trVal in trList[:6]:
        print(trVal.text)
```

Это показывает, сколько стоит 1 доллар США в других валютах.

A terminal window with a dark background. The title bar shows a file icon, a folder icon, and the path ~/Desktop. The command 'python3 linky.py' is entered in green text. The output shows currency conversion rates for Euro, British Pound, Indian Rupee, Australian Dollar, Canadian Dollar, and Singapore Dollar, each with two numerical values.

Euro
0.861126
1.161270

British Pound
0.783327
1.276606

Indian Rupee
74.859254
0.013358

Australian Dollar
1.411427
0.708503

Canadian Dollar
1.342391
0.744939

Singapore Dollar
1.385453
0.721786

Наука о данных и машинное обучение

DS и ML – самые модные темы в наши дни. Эти технологии – будущее компьютерной науки. Python хорошо подходит для манипулирования данными, анализа и реализации сложных алгоритмов. Разбор и визуализация данных обычно представляют собой простые функции или несколько строк кода с помощью таких библиотек Python, как NumPy, scipy, scikit-learn и т. д. Python можно использовать в

приложениях для работы с большими объемами данных и машинного обучения, используя множество популярных библиотек, таких как...

- **NumPy**
- **Pandas**
- **Matplotlib**
- **Seaborn**

Существует множество инструментов глубокого обучения, поддерживающих python. Некоторые популярные библиотеки и фреймворки – это

- **TensorFlow**
- **PyTorch**
- **Keras**

Одна из других причин использования python заключается в том, что даже сложные модели машинного обучения могут быть реализованы с помощью 20-40 строк кода.

Заключение

В учебнике обсуждались различные области, в которых можно использовать Python. Здесь я привожу несколько простых примеров, но существует множество других замечательных приложений и инструментов, которые вы можете создать с помощью Python. Надеюсь, вы узнали что-то новое!

Дата Создания

19.04.2024