

# СЪЗДАВАНЕ НА СЪДЪРЖАНИЕ

## Използване на модули и библиотеки

Днес живеем в свят, в който информацията е един от най-ценните ресурси. Огромни обеми данни се създават всяка секунда – от социални мрежи, сензори, мобилни устройства, научни изследвания и бизнес процеси. За да можем да вземаме информирани решения, е необходимо тези данни да бъдат **събрани, обработвани и представяни по разбираем начин**.

Един от най-популярните инструменти за обработка на големи масиви от данни е **скриптовият език Python**. Той е лесен за научаване, четим. Използва се от учени, инженери, програмисти и анализатори по целия свят за автоматизация, обработка на информация, изкуствен интелект и визуализация на резултатите.



Основната сила на Python е в огромния брой **библиотеки** (или **модули**), които разширяват неговите възможности. Те съдържат предварително написан код, който можем да използваме за решаване на всякакви задачи – от анализ на данни до разработка на уебсайтове и приложения.

За да е лесно търсенето и използването на тези библиотеки, те се съхраняват общодостъпни **хранилища**. Най-популярното такова хранилище е **PyPI (Python Package Index)**, където можем да търсим, изтегляме и инсталираме хиляди пакети. Други популярни платформи са **Anaconda**, която предлага библиотеки за наука и анализ на данни, и **GitHub**, където програмистите споделят и развиват свои проекти заедно.

**Задача 1.** Посетете уеб адреса на PyPI ([www.pypi.org](http://www.pypi.org)) и разгледайте платформата. Тъй като повечето модули/библиотеки са описани на английски език, търсенето по български думи е почти невъзможно. Потърсете какви модули има за генериране на шеги (jokes). Намерете информация за модула **pyjokes**.

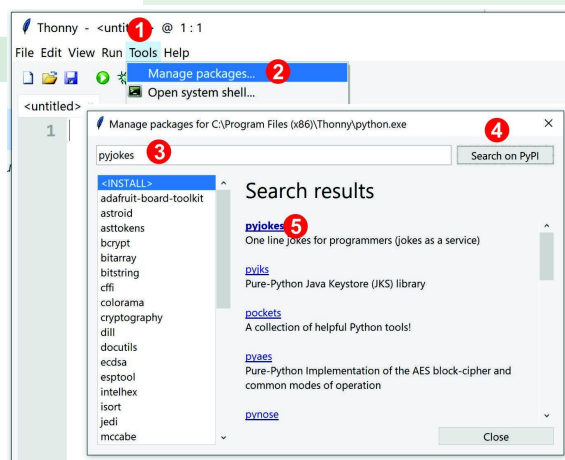
Преди да бъде използвана, всяка външна библиотека трябва да бъде инсталирана в средата за програмиране. Инсталирането става или с помощта на вградени в средата инструменти и менюта, или с команда в терминален прозорец.



## Инсталиране на модули и библиотеки

### Среда за програмиране Thonny

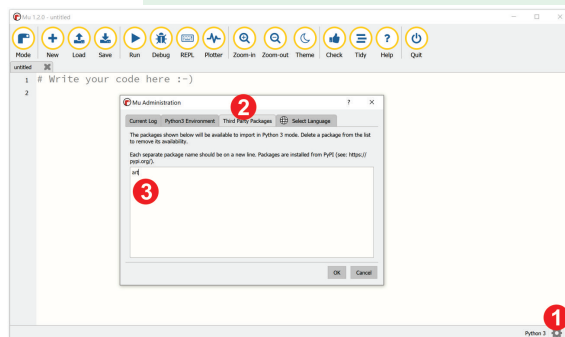
За да инсталираме библиотека в средата, е нужно да изберем опцията **Manage packages** (2) от менюто **Tools** (1). В отворения се прозорец трябва да потърсим името на библиотеката (3) и след това да кликнем върху намерения линк (5). Преди да натиснем бутона **Install**, можем да проверим дали това е точно библиотеката, която желаем да инсталираме.



Инсталиране на библиотека в Thonny

### Среда за програмиране Mu Editor

За да инсталираме библиотека в средата, е нужно да влезем в настройките на редактора. Това става от зъбното колело (1) в долния десен ъгъл. След това избираме секцията **Third Party Packages** (2) и пишем в прозореца името на библиотеката (3), която искаме да се инсталира. Редакторът се опитва да я намери в хранилището **PyPI.org** и да я инсталира.



Инсталиране на библиотека в Mu Editor

### Среда за програмиране PyCharm

В тази среда можем да инсталираме библиотеки по няколко начина. Едн от тях е да отворим терминалния прозорец (1) и там да напишем командата „**pip install ...**“, като на мястото на точките напишем името на библиотеката.

За да използваме инсталирана библиотека, е нужно да я импортираме в кода. Това може да стане по два начина:

1. Като напишем `import` и след това името на библиотеката (напр. `import pyjokes`).

2. Като напишем следната конструкция:

```
from ... import *
```

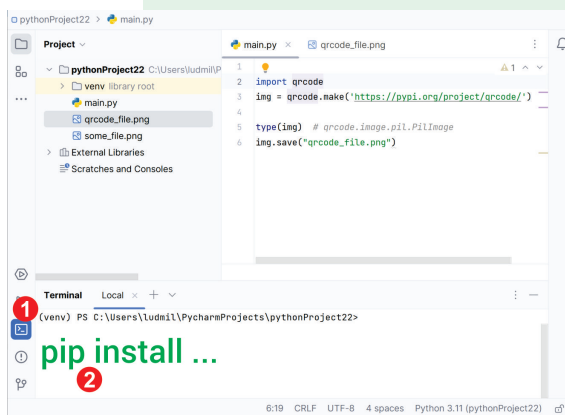
На мястото на точките трябва да сложим името на библиотеката.

При първия начин, за да използваме функция от библиотека-та трябва да напишем нейното име и точка след нея. След това пишем името на библиотеката. Например:

```
import pyjokes
joke = pyjokes.get_joke()
print(joke)
```

При втория начин използваме директно името на функцията. Например:

```
from pyjokes import *
joke = get_joke()
print(joke)
```



Инсталиране на библиотека в PyCharm

**Задача 2.** Последвайте линковете “Инсталиране на модули” от сайта [www.kmit.bg](http://www.kmit.bg) за съответния урок и разгледайте процеса на инсталиране на модули и библиотеки в различните среди за програмиране.

**Задача 3.** Стартирайте среда за програмиране на Python и инсталирайте модула `pyjokes`.

За да използваме библиотеката, след като вече е инсталирана, е нужно само да я вмъкнем в кода с командата `import`. Например:

```
import pyjokes
```

**Задача 4.** Изпробвайте инсталираната библиотека с помощта на следния код:

```
import pyjokes

joke = pyjokes.get_joke()
print(joke)
```

При правилна работа в конзолата ще видите шега, изписана на английски език. Библиотеката поддържа множество езици, но не и български. Нека да видим как можем да превеждаме шегите автоматично.

**Задача 5.** Потърсете и инсталирайте библиотеката за превод `translate`.

Нека да видим как можем да преведем на български език създадената в предната задача шега.

```
import pyjokes
from translate import Translator

joke = pyjokes.get_joke()
translator = Translator(to_lang="bg")
translation = translator.translate(joke)
print(translation)
```

Стартирайте програмата няколко пъти и разгледайте резултатите.

**Задача 1.** Като използвате кода на задачата от предходния урок, съставете програма, с помощта на която да извеждате само кратки шеги. Броят на символите на шегата на български език не трябва да надвишава 90.

Библиотека: **art**



ASCII Art Library for Python

**art** е библиотека от PyPI, която може да генерира ASCII банери, фигури и дори емотикони, направени от символи. Може да се използва, за да направите по-интересни и ефектни програми с изход в конзолата.

**Задача 2.** Потърсете информация за библиотеката в хранилището PyPI. Разгледайте как се използва. Инсталирайте библиотеката в проекта от предходната задача. Променете кода така, че да извежда само първата дума на шегата с тази библиотека като заглавие.

Тъй като не всички шрифтове в библиотеката поддържат кирилица, е добре да се вземе първата дума от текста на английски.

`words = joke.split()` – прави списък с всички думи в joke. Първата от тях е `word[0]`.

Проверете действието на кода и го коригирайте, ако има нужда.

Вижте какво е действието на командите:

Готови рисунки по име

```
aprint("butterfly")
```

```
aprint("coffee")
```

Случайно арт изображение

```
randart()
```

Примерен код:

```
import pyjokes
```

```
from translate import Translator
```

```
from art import *
```

```
# Генериране на шег
```

```
joke = pyjokes.get_joke()
```

```
# Превеждане на български
```

```
translator = Translator(to_lang="bg")
```

```
translation = translator.translate(joke)
```

```
# Отпечатване на заглавие
```

```
words = joke.split()
```

```
print(text2art(words[0]))
```

```
# Отпечатване на шегата
```

```
print(translation)
```

**Задача 3.** Потърсете библиотека за генериране на QR код. Инсталирайте я и направете QR код на сайт по ваш избор. За целта можете да използвате ИИ или кода, даден по-долу.

Библиотеката **qrcode**

Инсталация: `pip install qrcode[pil]`

Примерен код на програмата

```
import qrcode
```

```
data = input("Въведи линк (например https://google.com): ")
img = qrcode.make(data)
img.save("my_qr.png")
print("Готово! QR кодът е записан като 'my_qr.png'")
print("Можете да го откриете в папката на проекта.")
```



**Задача 4.** Създайте програма, която изчислява разстоянието между две географски точки (например два града). Използвайте библиотеката **geopy**. За генериране на програмата използвайте ИИ или примерния код по-долу.

```
from geopy.distance import geodesic
from geopy.geocoders import Nominatim

geolocator = Nominatim(user_agent="myGeocoder")

# Географски координати на два града
city_1 = input('Въведете първия град:')
city_2 = input('Въведете втория град:')
location1 = geolocator.geocode(city_1)
location2 = geolocator.geocode(city_2)

# Изчисляване на разстоянието
coords1 = (location1.latitude, location1.longitude)
coords2 = (location2.latitude, location2.longitude)

# 5. Изчисляване на разстоянието
distance = geodesic(coords1, coords2).kilometers
print(f"Разстоянието е {distance:.2f} км.")
```

