

СЪЗДАВАНЕ НА СЪДЪРЖАНИЕ

27.

БИБЛИОТЕКИ ЗА РЕАЛИЗИРАНЕ НА ПРОГРАМНИ ВЪЗМОЖНОСТИ

В ТОЗИ УРОК

■ Ще научите какво е:

- библиотека.
- вградена и външна библиотека.

■ Ще можете да:

- откривате нови модули/пакети/библиотеки за реализиране на програмни възможности съобразно идентифицирани потребности
- познавате начини за инсталиране на намерените модули/пакети/библиотеки.

■ Интересно

Идеята за библиотеки в програмирането възниква през 50-те години на XX век. Тогава учените започват да събират често използвани функции в общи „каталози“, за да не се пише един и същи код многократно.

Преглед: Като част от научен екип трябва да изследвате температурите във вашия град за последната седмица и да направите кратък доклад. За задачата трябва резултатите да бъдат визуализирани с диаграма.



Можем ли да направим всичко това само с командите `print`, `input`, `if`?

БИБЛИОТЕКИ

За да реализираме програма, която анализира данни и създава диаграма, се нуждаем от инструменти, които не са част от основния език за програмиране. Библиотеките са готови колекции от функции, класове и модули. Те са създадени специално за определени задачи.

Библиотеките разширяват възможностите на езика за програмиране. С тях можем да изчисляваме сложни математически операции, да работим с таблици, да визуализираме данни, да създадем изображения. Те са съвкупност от готови инструменти, които програмистът може да импортира и използва в своя код. Това спестява време, намалява грешките, позволява да се създават по-сложни проекти с много по-малко усилия. Например:

- `math` – математически операции
- `random` – генериране на произволни числа
- `csv` – работа с таблични данни
- `matplotlib` – визуализации
- `pillow` – обработка на изображения.

ВИДОВЕ БИБЛИОТЕКИ

Вградени библиотеки – част от езика и могат да се използват без допълнителна инсталация, веднага чрез импортиране. Например: `math`, `random`, `datetime`, `csv`, `turtle`.

Външни библиотеки – инсталират се допълнително за нуждите на проекта. Например: `matplotlib` (диаграми), `pillow` (изображения).

Задача 1. Създайте програма, която генерира случайно число от 1 до 10 и намира квадратния му корен.

За да се използва дадена библиотека, трябва първо да се **импортира**. Това означава да се включат и използват инструментите от тази библиотека.

■ Искам да покажа:

1. как в библиотеката `matplotlib` могат да се променят:

- цвят на линията
- стил на линията
- дебелина на линията
- маркери за отбелязване на точките.

■ Задачи за самостоятелна работа:

Създайте бар диаграма с данните от задача 2 (`plt.bar`) и я сравнете с линейната диаграма. Коя представя информацията по-ясно?

Импортиране на цяла библиотека

```
1 import random
2 import math
3
4 x = random.randint(a: 1, b: 10)
5 print("Числото е:", x)
6 print("Коренът му е:", math.sqrt(x))
```

Импортиране с псевдоним

```
1 import random as rnd
2 import math as m
3
4 x = rnd.randint(a: 1, b: 10)
5 print("Числото е:", x)
6 print("Коренът му е:", m.sqrt(x))
```

Недостатък: Ако псевдонимът (`rnd`, `m`) е кратък, решението на задачата може да не е ясно.

Импортиране на конкретна функция

```
1 from random import randint
2 from math import sqrt
3
4 x = randint(a: 1, b: 10)
5 print("Числото е:", x)
6 print("Коренът му е:", sqrt(x))
7
```

Недостатък: Ако има две функции с едно и също име от различни библиотеки, може да се получи конфликт.

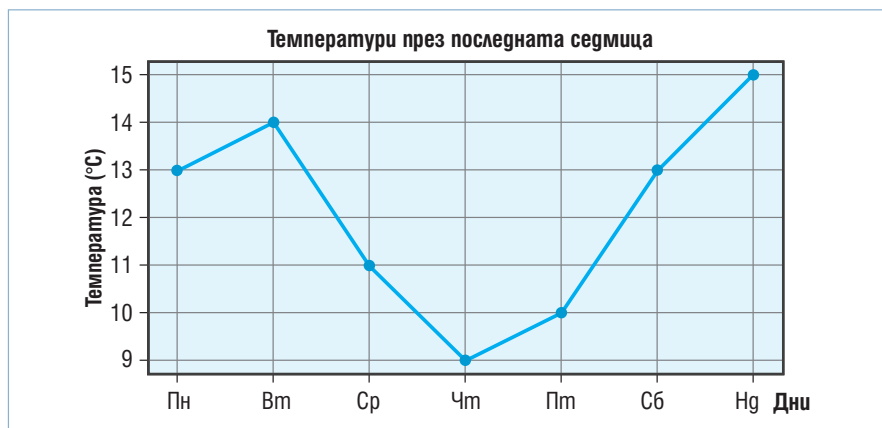
ИНСТАЛИРАНЕ НА БИБЛИОТЕКА

Задача 2. Създайте програма, която визуализира температурите във вашия град за последната седмица, като използва външната библиотека `matplotlib`. Можете да потърсите информация в интернет или да използвате данните от примерната таблица.

Ден от седмицата	Температура (°C)
понеделник	12
вторник	14
сряда	11
четвъртък	9
петък	10
събота	13
неделя	15

Програмата трябва да:

- съдържа списък с температурите за 7 дни
- използва външната библиотека `matplotlib` за създаване на диаграмата
- визуализира температурите чрез линейна графика
- добави подходящо заглавие на диаграмата, означения на осите и мрежа
- запише получената диаграма като изображение `temps_week.png`.

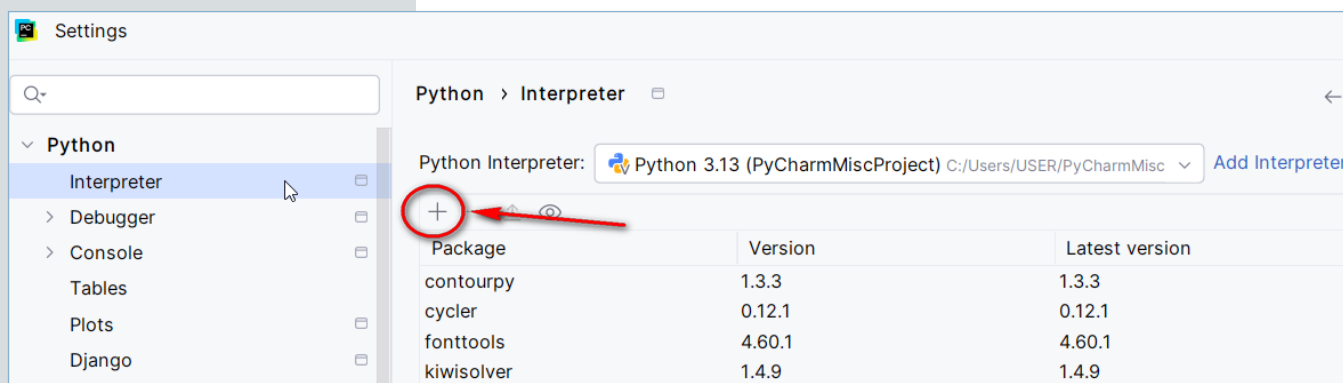


Среда за програмиране PyCharm:

За да се инсталира библиотека в терминалния прозорец, се изписва командата `pip install` име на библиотеката.



От *File* → *Settings* могат да се видят всички инсталирани библиотеки. Тук е мястото, откъдето също могат да се добавят нови библиотеки, да се изтриват и да се актуализират.



Среда за програмиране Google Colab:

В Google Colab са налични множество популярни библиотеки като: `matplotlib`, `pillow`, `csv`. Това означава, че могат да се използ-

ват директно. Ако библиотеката не е предварително налична, позволява добавяне по следния начин:

 **!pip install** име на библиотеката.

В Google Colab командите, които започват с `! pip instal`, не са Python команди, а са системни команди. Добра практика е те да са в самостоятелна клетка, отделени от останалия код.



Задача 3. Създайте програма, която използва външната библиотека `emoji`, за да показва времето чрез емоджита.

Пн:	12 °C	
Вт:	14 °C	
Ср:	11 °C	
Чт:	9 °C	
Пт:	10 °C	
Сб:	13 °C	
Нд:	15 °C	

СЪВЕТ: инсталирайте библиотеката:
`pip install emoji==2.8.0`

Решение:

```
pip install emoji==2.8.0
```

```
Collecting emoji==2.8.0
```

```
  Downloading emoji-2.8.0-py2.py3-none-any.whl.metadata (5.3 kB)
```

```
  Downloading emoji-2.8.0-py2.py3-none-any.whl (358 kB)
```

```
358.9/358.9 kB 4.1 MB/s eta 0:00:00
```

```
Installing collected packages: emoji
```

```
Successfully installed emoji-2.8.0
```

```
import emoji
```

```
temperatures = [12, 14, 11, 9, 10, 13, 15]
```

```
days = ["Пн", "Вт", "Ср", "Чт", "Пт", "Сб", "Нд"]
```

```
for day, temp in zip(days, temperatures):
```

```
    if temp > 15:
```

```
        symbol = emoji.emojize(":sun_with_face:")
```

```
    elif temp >= 10:
```

```
        symbol = emoji.emojize(":sun_behind_cloud:")
```

```
    else:
```

```
        symbol = emoji.emojize(":snowflake:")
```

```
    print(f"{day}: {temp}°C {symbol}")
```

Въпроси и упражнения:

1. Каква е разликата между вградена и външна библиотека?
2. Коя библиотека бихте избрали за създаване и редактиране на изображения – зареждане, преоразмеряване, добавяне на текст върху изображение?

28.

ПРИЛОЖЕНИЕ НА БИБЛИОТЕКИ
ЗА ПРОГРАМИРАНЕ

В ТОЗИ УРОК

■ Ще се упражните да:

- прилагате налични технологични средства за използване на намерените модули, пакети или библиотеки.

■ Интересно

CSV (Comma-separated values) е прост текстов формат за таблични данни, при който всяка редица е на нов ред, а клетките в реда са разделени със запетая. На практика разделителят понякога е друг символ (например точка и запетая или таб), но името CSV е останало исторически. Създаден е от нуждата за прост, четим текстов обмен на таблични данни. Започва да се използва още от 70-те години на XX век.



Предизвикателство: Храненето е важна част от нашето всекидневие. Със сигурност се интересувате от здравословно хранене и обсъждате с приятели. Направете табло „Любими закуски в нашия клас“, като преди това сте събрали отговори на съучениците си в таблица.

За да направим таблото, може да използваме езика *Python* и неговите библиотеки, които да ни помогнат да обработим резултатите от анкетата (таблица CSV), да създадем проста диаграма (коя закуска е най-предпочитана) и да направим красиво изображение плакат (фон, диаграма, заглавие).

Задача 1. Потърсете подходящи библиотеки за езика за програмиране *Python*, с които може да се обработят данни от таблица във формат *.csv*; да се създаде диаграма от подадени данни; да се създаде изображение. Инсталирайте ги.

Подходящи библиотеки за изпълнението на предизвикателството са:

- вградена библиотека: *csv* (четене на данни от CSV файл – електронна таблица в текстов вид)
- външни библиотеки:
 - **matplotlib** (рисуване на диаграма)
 - **Pillow** (обработка/сглобяване на изображение).

В среда на *Python* в нов файл изпълнете веднъж: `!pip install pillow matplotlib`, за да инсталирате двете външни библиотеки.

Задача 2. Създайте код, който чрез библиотеката *csv* прочита данните от файла *breakfast.csv* и преброява по колко пъти се среща всяка любима закуска.

Решение:

```
import csv # вмъкване на библиотека за четене на CSV файл

labels = [] # списък за уникалните закуски (етикети)
values = [] # списък от цели числа за броя на всяка закуска

# отваряме файла
with open("breakfast.csv", encoding="utf-8-sig") as f:
    # цикъл за обхождане на файла ред по ред
    for row in csv.reader(f):
        snack = row[0] # променлива със стойността на първия ред
        if snack in labels: # ако вече се е срещала тази закуска
            i = labels.index(snack) # намира на коя позиция е
            values[i] += 1 # увеличава позицията с 1
```

■ Искам да покажа:

1. как може да се съберат отговорите на съучениците чрез електронен формуляр
2. как може да се реши задача 2 чрез клас от езика за програмиране Python за броене на елементи.

■ Задачи за самостоятелна работа:

1. Напишете код в Python с библиотеките *csv*, *pillow* *matplotlib* за създаване на плакат за любим спорт на момчетата и любим спорт на момчетата от класа.
2. Добавете дата под заглавието на плакатите.

```
else: # ако е нова закуска
    labels.append(snack) # добавя я в списъка с етикети
    values.append(1) # продължава броенето

# показва резултата като двойки (закуска, брой)
print(list(zip(labels, values)))
```

Резултат:

```
[('банан', 7), ('баница', 4), ('мюсли', 4)]
```

Решение на задачата чрез клас от езика на програмиране Python за броене на елементи:

```
import csv
from collections import Counter

with open("breakfast.csv", encoding="utf-8-sig") as f:
    counts = Counter(row[0] for row in csv.reader(f))

print(dict(counts))
```

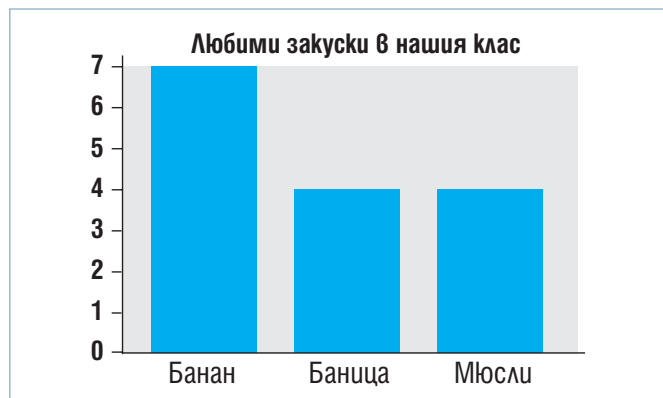
Задача 3. Създайте код, който чрез библиотеката *matplotlib* създава диаграма, визуализираща данните от таблицата.

Решение:

```
import matplotlib.pyplot as plt

plt.bar(labels, values)
plt.title("Любими закуски в нашия клас")
plt.savefig("chart.png"); plt.show()
```

Резултат:



Задача 4. Напишете код чрез библиотеката *pillow* в *Python*, който създава изображение, подходящо за плакат с данните от проучването коя е любимата закуска на класа. Плакатът да съдържа заглавие, син фон и диаграмата от предишната задача.

Решение:

```
# импортираме нужните класове от Pillow
from PIL import Image, ImageDraw, ImageFont
# импортираме мениджър на шрифта
from matplotlib import font_manager
```

```
# намираме шрифт, който поддържа на кирилица
font_path = font_manager.findfont("DejaVu Sans")

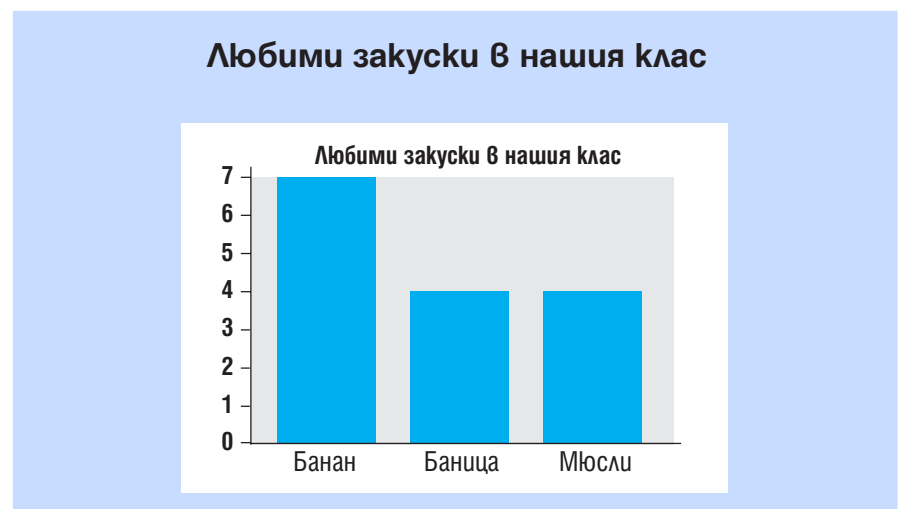
# създаваме празно изображение 1200x800 със син фон
img = Image.new("RGB", (1200, 800), (200, 220, 255))

# създаваме „четка“ за рисуване върху изображението
draw = ImageDraw.Draw(img)
# зареждаме Truetype шрифт с размер 64
font = ImageFont.truetype(font_path, 64)

# рисуваме заглавието горе вляво
draw.text((40, 30), "Любими закуски в нашия клас", fill=(0, 0, 0), font=font)

# зареждаме PNG диаграмата от задача 3
chart = Image.open("chart.png")
# преоразмеряваме до ширина 1000 px и намалена височина
chart = chart.resize((1000, int(chart.height * 1000 / chart.width/2)))
# поставяме диаграмата върху плаката на позиция (100, 240)
img.paste(chart, (100, 240))
# записваме готовия плакат като PNG файл
img.save("poster_lubimi_zakuski.png")
```

Резултат:



Въпроси и упражнения:

1. По какво ще познаете дали в използваната от вас среда библиотеката, която искате да инсталирате, е вградена?
2. Сменете цвета на фона на плаката по желание, удебелете заглавието, намалете размера на диаграмата.