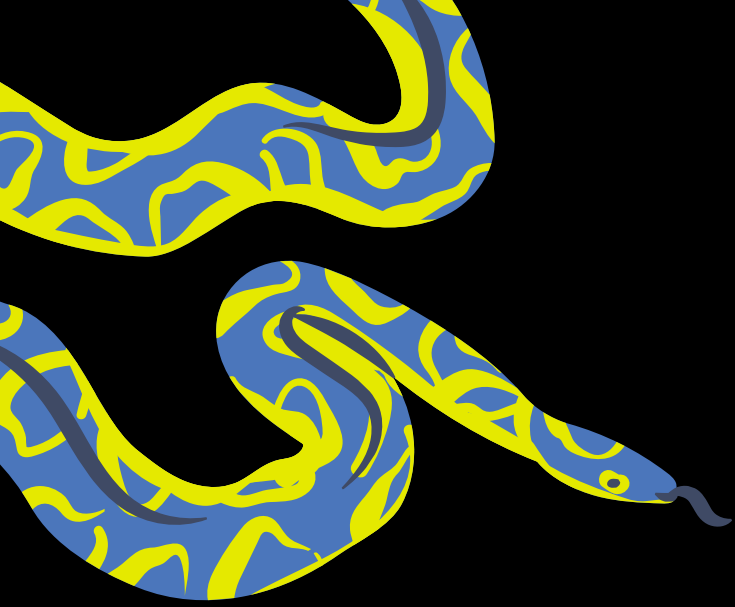




SciByteHub



Python

28.05



SciByteHub

Logowanie do komputerów

login: **nazimi1sbh** **(NAZwisko IMIe)**

hasło: **Edeibcabej024!**

RECAP - import

Importowanie bibliotek do kodu:

```
import biblioteka
```

Importowanie biblioteki jako alias:

```
import biblioteka as alias
```

API - Application Programming Interface

Czym jest API?

Zestaw reguł, które umożliwiają aplikacjom komunikować się ze sobą (Działa jak pośrednik lub tłumacz na linii użytkownik - program)

Konkretne API

REST (Representational State Transfer) - sposób komunikacji aplikacji. Wymiana informacji w formacie JSON. Aplikacje mogą pobierać, aktualizować lub usuwać dane



SciByteHub

Zastosowania API

Integracja systemów

Przykładowo - łączenie aplikacji i usług, takich jak połączenie sklepu internetowego z systemem płatności

Tworzenie aplikacji mobilnych

Aplikacje mobilne korzystają z API, żeby pobierać dane z serwerów, np. aplikacja, która pokazuje pogodę, pobiera dane z odpowiedniej strony

Powtarzalne czynności

Przykładowo - wysyłanie automatycznych powiadomień e-mail

API - pojęcia

Klient

Aplikacja, która wysyła żądania do API, aby uzyskać dostęp do danych lub funkcji

Serwer

Aplikacja, która odbiera żądania Klienta, przetwarza je i zwraca odpowiedź klientowi

Requests

Wiadomość wysyłana przez klienta do serwera z informacjami, co klient chce zrobić

Responses

Wiadomość zwrotna od serwera, zawierająca wynik żądania - dane o statusie operacji



SciByteHub

API - przykładowe statusy

200 OK

Żądanie obsłużone pomyślnie

404 Not Found

Żądany zasób nie został znaleziony

400 Bad Request

Żądanie niepoprawne lub niekompletne

401 Unauthorized

Wymagane uwierzytelnienie

500 Internal Server Error

Błąd po stronie serwera



SciByteHub

API - Metody

GET

Pobieranie danych z serwera

POST

Wysyłanie nowych danych na serwer

PUT

Aktualizacja istniejących danych na serwerze

DELETE

Usuwanie danych z serwera

API - Biblioteka requests

Biblioteka requests to narzędzie do interakcji z API w Pythonie. Umożliwia wysyłanie różnego rodzaju żądań i obsługę odpowiedzi

Większość API zwraca dane w formacie JSON, a biblioteka requests ułatwia obsługę tych odpowiedzi



API - Biblioteka requests

Biblioteka *requests*

instalacja - *pip install requests*

```
import requests

# GET request
response = requests.get('https://api.example.com/data')

# POST request
data = {'key': 'value'}
response = requests.post('https://api.example.com/data', data=data)

# PUT request
new_data = {'key': 'new_value'}
response = requests.put('https://api.example.com/data/1', data=new_data)

# DELETE request
response = requests.delete('https://api.example.com/data/1')
```



SciByteHub