Concepts Behind Python News App Using APIs

This file explains all core Python concepts used in the News Fetching Project, including logic and code examples. It helps in revision and better recall for GitHub documentation.

✅API (Application Programming Interface)

Meaning: API allows two software systems to communicate. We send a request and the server returns data.

In this project: We use `NewsAPI` to fetch real-time news.

```
# Example API request base URL
https://newsapi.org/v2/everything
```

✅API Key

Meaning: A secret unique key given by API providers to authenticate users.

```
api = "your_api_key_here"
```

Used like a password to prevent misuse of API.

✅`requests` Module

Purpose: To send HTTP requests and get data from internet APIs.

```
import requests
```

🔲Sending GET Request

```
r = requests.get(url)
```

This retrieves information from the given API URL.

✅User Input in Python

Takes data from user through keyboard.

```
query = input("Enter topic: ")
```

✅String Formatting (f-strings)

Used to insert variables inside strings.

```
url = f"https://newsapi.org/v2/everything?q={query}&apiKey={api}"
```

✅JSON Data Handling

API sends data in **JSON format**. We convert it to Python dictionary using:

```
data = r.json()
```

Accessing JSON keys

```
articles = data.get("articles", [])
```

`.get()` is safer because it avoids errors if key is missing.

✅Conditional Statements

To check API status and handle errors:

```
if data.get("status") != "ok":
    print("Error fetching news")
```

✅Looping Through Articles

We use `for` loop to display each article.

```
for article in articles[:10]:
    print(article.get("title"))
```

`[:10]` → Shows only first 10 results.

✅Printing Separator

For clean UI

```
print("-" * 60)
```

Prints 60 dashes.

✅Error Handling Logic

Case when no results found:

```
if not articles:
    print("No articles found.")
```

✅Final Working Code (Reference)

```
import requests
query = input("Enter topic for news: ")
api = "YOUR_API_KEY"
url = f"https://newsapi.org/v2/everything?q={query}
&sortBy=publishedAt&apiKey={api}"
print(f"\nFetching latest news for '{query}'...\n")
r = requests.get(url)
data = r.json()
if data.get("status") != "ok":
    print("❌ Error:", data.get("message"))
else:
    articles = data.get("articles", [])
    if not articles:
        print("No articles found.")
    else:
        for article in articles[:10]:
            print("💬", article.get("title"))
            print("☪️", article.get("description"))
            print("✅", article.get("url"))
            print("-" * 60)
```

✅Summary Table

Concept	Purpose
API	Get online data
API Key	Authentication
requests library	Sends request to server
JSON	Data format
input()	Take user input
f-string	Format URL
if-else	Error handling
for loop	Display results
slice `[:10]`	Limit output

🔝This project teaches

- Python Basics
- API Handling
- JSON Processing
- CLI Application Design
- Real-world Programming

End of notes ✅

> Use these notes for GitHub README and learning revision.