🎨Command-Line Utilities, cURL, argparse & requests (Colorful Cheat Sheet)

🏐What is a Command-Line Utility?

- A **program/tool** that runs in a **text-based interface** (like CMD, PowerShell, Bash, Terminal).
- Takes **commands and arguments** as input and produces **text output**.
- Lightweight, efficient, and commonly used by developers and system admins.

Examples

- **File Management** → `ls`, `dir`, `cp`, `rm`
- **Networking** → `ping`, `curl`, `ipconfig`, `ifconfig`
- **System Utilities** → `top`, `tasklist`, `df`

👉Example: `curl -o file.jpg https://example.com/image.jpg`

🏐What is cURL?

- **cURL = Client URL**
- A command-line tool & library to **transfer data** with URLs.
- Supports many protocols: HTTP, HTTPS, FTP, FTPS, SCP, SFTP, LDAP, POP3, IMAP, SMTP, etc.
- Built into **Linux, macOS, Windows 10+**.

👉Think of cURL as a **Swiss army knife for network requests**.

🏐Basic Structure (cURL)

```
curl [options] [URL]
```

- `curl` → the command
- `[options]` → flags to control behavior
- `[URL]` → the resource you want to interact with

🏐Common Use Cases (cURL)

1. Download a File

```
curl -o filename.jpg https://example.com/pic.jpg
```

- `-o` → save file with given name

2. Follow Redirects

```
curl -L https://short.url/abc
```

- `-L` → follow redirects

3. Get Headers Only

```
curl -I https://www.google.com
```

- `-I` → fetch only HTTP headers

4. POST Request (Form Data)

```
curl -X POST -d "username=zahid&password=123" https://example.com/login
```

- `-X POST` → HTTP method
- `-d` → send form data

5. Send JSON Data (API)

```
curl -X POST -H "Content-Type: application/json" -d '{"name":"Zahid","age":24}'
https://api.example.com/users
```

- `-H` → set request header
- `-d` → send JSON body

6. Authentication

```
curl -u user:password https://example.com/secure
```

- `-u` → username\:password

🏐General Pattern (cURL)

```
curl [METHOD/OPTIONS] [HEADERS] [DATA] [URL]
```

- **METHOD** → `-X GET`, `-X POST`, `-X PUT` ...
- **OPTIONS** → `-o`, `-L`, `-I` ...
- **HEADERS** → `-H "Header: Value"`
- **DATA** → `-d "key=value"` or JSON
- **URL** → target resource

🏐Real Life Uses of cURL

- 👐Downloading files
- 👉Testing APIs
- 👇Uploading files
- ☹️Sending POST requests
- 👹Debugging websites
- 🤖Automating scripts (DevOps)

🏐argparse in Python

- A **built-in module** for creating **command-line interfaces**.
- Lets you define **arguments and options** users can pass into your Python scripts.

Example

```python
import argparse

parser = argparse.ArgumentParser(description="Simple CLI example")
parser.add_argument("name", help="Your name")
parser.add_argument("-g", "--greet", action="store_true", help="Add greeting")

args = parser.parse_args()

if args.greet:
    print(f"Hello, {args.name}!")
else:
    print(args.name)
```

Run:

```
python script.py Zahid --greet
```

Output:

```
Hello, Zahid!
```

👈**Usage**: Build custom command-line utilities, parse user input.

🏐requests in Python

- A **third-party library** (`pip install requests`).
- Makes it easy to send **HTTP requests** (GET, POST, PUT, DELETE).

Example

```python
import requests

# GET request
response = requests.get("https://api.github.com/users/octocat")
print(response.json())

# POST request
payload = {"username": "zahid", "password": "123"}
response = requests.post("https://httpbin.org/post", data=payload)
print(response.text)
```

👈**Usage**: Access web APIs, send/receive data, automate HTTP tasks.

👎With **Command-Line Utilities, cURL, argparse, and requests**, you can **interact with systems, the web, and APIs efficiently** 😢