# 📶 Skill 2 Networks

هذا هو رابط الفيديو المرتبط بهذه التمارين:
https://youtube.com/playlist?list=PL5dZpxpUkHPPGwJS_xFznbdLtK0fD5WQF&si=lOibnJfUPuTA7AZ9

تنبيه مهم: إذا قمت بنسخ السؤال ولصقه في ChatGPT مباشرة للحصول على الحل، فرجاءً قم بإلغاء اشتراكك والانسحاب مبكرًا من الرحلة. صدقني، أنا أساعدك على توفير وقتك ومجهودك غير المرئي""

---

## 🔥 Day 1: IP Addresses & Subnetting

### Topics: IP Basics, IPv4, IPv6, Subnetting, CIDR Notation

🔹 **Exercises:**

1. Write a Python script that validates whether a given IP address is IPv4 or IPv6.
2. Convert an IPv4 address from dotted-decimal to binary format.
3. Extract all IP addresses from a text file.
4. Write a function that checks if an IP belongs to a private network (192.168.x.x, 10.x.x.x, etc.).
5. Given an IP and subnet mask, calculate the network address and broadcast address.
6. Generate a list of all possible IPs in a given subnet (192.168.1.0/24).
7. Create a script that checks if an IP is reachable by pinging it.
8. Convert an IPv6 address to its compressed and expanded forms.
9. Write a function that determines the class of an IPv4 address (A, B, C, D, E).
10. Given two IPs, determine if they belong to the same subnet.

---

## 🔥 Day 2: Ports & Protocols

### Topics: Port Numbers, Well-Known Ports, Protocols (TCP/UDP)

🔹 **Exercises:**

1. Print a list of common ports and their corresponding services (e.g., 80 → HTTP).

2. Write a function that checks if a given port number is valid (0-65535).
3. Write a script that scans the first 1000 ports of a given IP to see if they are open.
4. Detects whether a given port uses TCP or UDP.
5. Randomly generate 5 open ports between 1024-65535.
6. Create a function that tells whether a given port is in the privileged range (0-1023).
7. Write a script that finds all listening ports on your local machine (127.0.0.1).
8. Implement a basic TCP port scanner using Python's socket module.
9. Implement a basic UDP port scanner.
10. List all reserved ports (1-1023) that are commonly used in hacking.

---

## 🔥 Day 3: TCP & UDP

### Topics: Connection-oriented vs. Connectionless, Sockets, Packet Handling

🔹 Exercises:

1. Write a simple TCP client-server program that sends a message from client to server.
2. Modify the above program to work with UDP instead of TCP.
3. Create a TCP server that listens on port 9999 and responds with "Hello, Client!".
4. Write a function that measures the round-trip time (RTT) of a TCP connection.
5. Simulate a basic SYN flood attack (for controlled environments).
6. Write a script that monitors and logs all incoming TCP connections.
7. Implement a TCP handshake simulation in Python.
8. Create a script that tests whether a remote server allows telnet access (port 23).
9. Write a UDP client that sends a DNS request manually to 8.8.8.8.
10. Implement a simple packet sniffer that captures TCP and UDP packets.

---