The Basics

Congratulations, we made it! We can finally get to writing some code.

We'll cover the following

Socket Programming

- Recall that network applications usually consist of two programs: the **server** program and the **client** program. These programs reside on two separate end systems.
- When any of these programs want to communicate with another, they write the data they want to send to their **sockets**. The underlying protocols then deliver the data to the appropriate destination.

Types of Network Applications

- **Standard**: applications that use well-known protocols based on meticulous standards laid down by standards documents.
  - Pros: any other developer can write applications that are compatible with standard ones.
  - Cons: some customizability will be compromised.
- **Proprietary**: applications that use protocols of the developer's own design. The source code for such applications is generally not disclosed.
  - Pros: Extremely customizable, which allows for optimizing the application for particular use cases.
  - Cons: It's incredibly difficult to write applications compatible with a proprietary one. Also, designing protocols can lead to security

loopholes.

> 🔎 **Did You Know?** Writing applications compatible with a proprietary one is doable but hard. A case in point is **Windows network file sharing**. Windows had a proprietary Server Message Block (SMB) protocol for file sharing. Only Windows clients could use Windows shared files, folders and printers. But then some hackers collected packet traces and reverse-engineered the protocol and came up with an open-source implementation named SAMBA, using which, Linux clients could access Windows file shares and even share files, folders, and printers with a Windows domain. A challenge with this approach is that when Windows updates their proprietary protocol, the open-source clients and servers don't talk to the new version of Windows and the hackers have to pull up their sleeves all over again.

Socket Programming in Python

Anyways, we're going to be learning the very basics of writing an application in Python. It's a great way to truly apply what you've learned.

Here's what to expect from the rest of this chapter:

1. We'll start with the very basics. You'll learn how to set up a UDP socket using the `socket` library.

2. We'll then write a complete server and a client program that runs on UDP. We'll also see them in action.

3. We'll then look at some improvements on the UDP server.

4. You'll get to do a project next! You'll be writing your very own chat app.

5. Lastly, we'll see how a few tweaks can change the program into one that runs on TCP.

Let's start with setting up a socket next!