第3章 ソケットの生成からコネクション確立要求の待ち受け，データ受信，切断処理まで

レッスン②… TCPソケット通信の受信

国野 亘

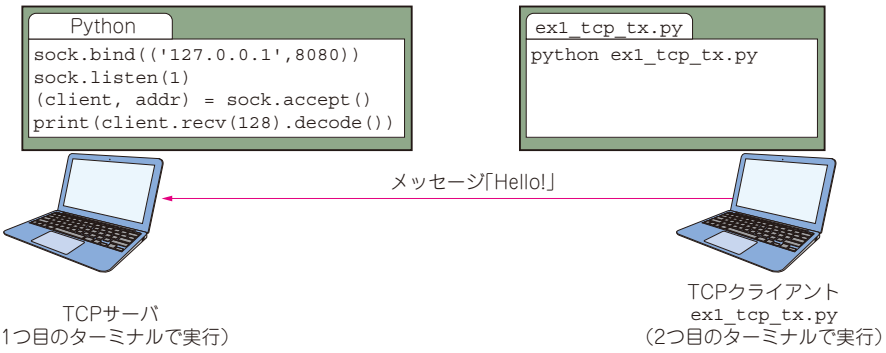

図1 TCPサーバの通信実験の構成
TCPの接続を待機し，TCPクライアントからのメッセージを受信する

レッスン②では，Pythonの対話モードを使ってTCPサーバの動作を確認します．

1台のPC上で2つのターミナルを使用し，片側はTCPクライアント，もう片側はTCPサーバとして使用します（**図1**）．レッスン①から続けて行う場合は，テスト用TCPサーバのターミナルを閉じます．

ステップ①…対話モードで通信実験

● 手順1：Pythonの対話モードを起動する

ターミナル上でPythonの対話モードを起動します．次のコマンドを入力します．

```
> python
```

● 手順2：ソケット通信用オブジェクトを生成する

ソケット通信用のライブラリ・モジュールであるsocketを組み込み，sockオブジェクトを生成します．

```
>>> import socket
>>> sock = socket.socket(socket.
      AF_INET, socket.SOCK_STREAM)
```

● 手順3：ソケットを生成する

bindメソッドを使って，ソケット通信インターフェースを生成します．引数はTCPサーバとして使用するIPアドレスとポート番号です．IPアドレスを''（シングル・クォート2つ）で省略することもできます．

次のコマンドを実行すると，ローカル・ループバック・アドレス127.0.0.1とポート番号8080に接続します．

```
>>> sock.bind(('127.0.0.1', 8080))
```

エラー「Address already in use」が表示された場合はコラムを参照してください．

次に，listenメソッドを使って同時待ち受け数を設定します．次のコマンドは同時待ち受け数1の場合の実行例です．

```
>>> sock.listen(1)
```

● 手順4：TCPコネクション確立要求を待ち受ける

ソケットに到達するTCPコネクション確立要求（以降，ソケット通信でよく使われているSYNと表記）を待ち受けるには，acceptメソッドを使います．次のコマンドを実行すると，クライアントからの接続要求を受け付けます．

```
>>> (client, addr) = sock.accept()
```

SYNを受信するまで何も表示されないので，次に進みます．

● 手順5：TCPクライアントからコネクション確立要求を行う

ここで，別のターミナルを起動して，レッスン①で