The Python standard library module I want to introduce is Socket module. Socket module is used to construct network communication channels between servers and clients. Many network application developers may need it since it contains a lot of functions which can be used to construct specific channels according to specific situations. For example, bank application developers may need to develop their own network channel because of security concerns. Some encrypted video applications may also need authorization and authentication functions to ensure user's network security. Thus, Socket module is valuable to these developers because it can be used to construct basic and encrypted channels for data communications.

The basic idea of socket application is client-server mode. The server creates sockets and waits for client to connect and the client sends requests to server and waits for response from the server. There are two basic types of protocols that can be created by Socket module, which are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) respectively. The primary functions and methods in Socket module are socket(), bind(), listen(), accept(), connect(), send(), recv(), and close(). In the process of creating UDP, the server side firstly creates a socket and bind the socket to the IP address of the server and the port number of the service. The client side then creates a socket and connects to the IP address and the port number. If the connection is made successfully, the client side sends message to the server side and the server side receives the message from the client side, vice versa. After the message transmission has finished, both the client side and the server side close their sockets and ends the message communication. UDP has advantages of fast and easy to construct, but it also has disadvantages of unreliable and no error checking. The process of TCP is almost the same as the process of UDP. The only difference between these two protocols is that the server side of TCP needs to listen and accept the connections from the client side and the client side also needs the confirmation of channel established from the server side. This extra step makes the channel more reliable and enables the data transmitted to be received in order. However, TCP makes the application heavier, which leads to the difficulty of maintenance.

In conclusion, Socket module is valuable for network communication related application developers since it supplies many useful functions and methods for the developers to customize their own needed network communication channels. The socket() function allows developers to create a new socket object, the send() function allows the new created socket to send the message through the network, and the recv() function allows the socket to receive the message from the specified IP address. A reliable channel can be formed by using the key functions of the Socket module.