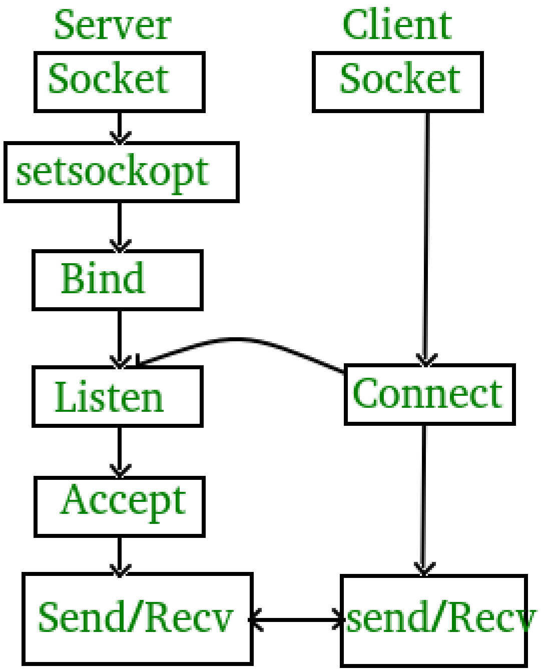
SOCKET PROGRAMMING:

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while the other socket reaches out to the other to form a connection. Server forms the listener socket while the client reaches out to the server.



There is a full-fledged approach to connect two or more sockets(nodes) on a network for communication. When it comes to programming, it is called Socket Programming. These sockets or nodes form a client-server relationship between them. The listener socket becomes the server while the client requests the server or the listening socket.

Socket() Module

To create a socket, we must use socket.socket() function available in the Python socket module, which has the general syntax as follows:

```
S = socket.socket(socket_family, socket_type, protocol=0)
```

Client Socket Methods:

connect():
To connect to a remote socket at an address. An address format(host, port) pair is used for AF_INET address family

Server Socket Methods

Following are some server socket methods:

bind( ): This method binds the socket to an address. The format of address depends on the socket family mentioned above(AF_INET).

listen(backlog): This method listens for the connection made to the socket. The backlog is the maximum number of queued connections that must be listened to before rejecting the connection.

accept( ): This method is used to accept a connection. The socket must be bound to an address and listen for connections. The return value is a pair(conn, address) where conn is a new socket object which can be used to send and receive data on that connection, and address is the address bound to the socket on the other end of the connection.

Few General Socket Methods:

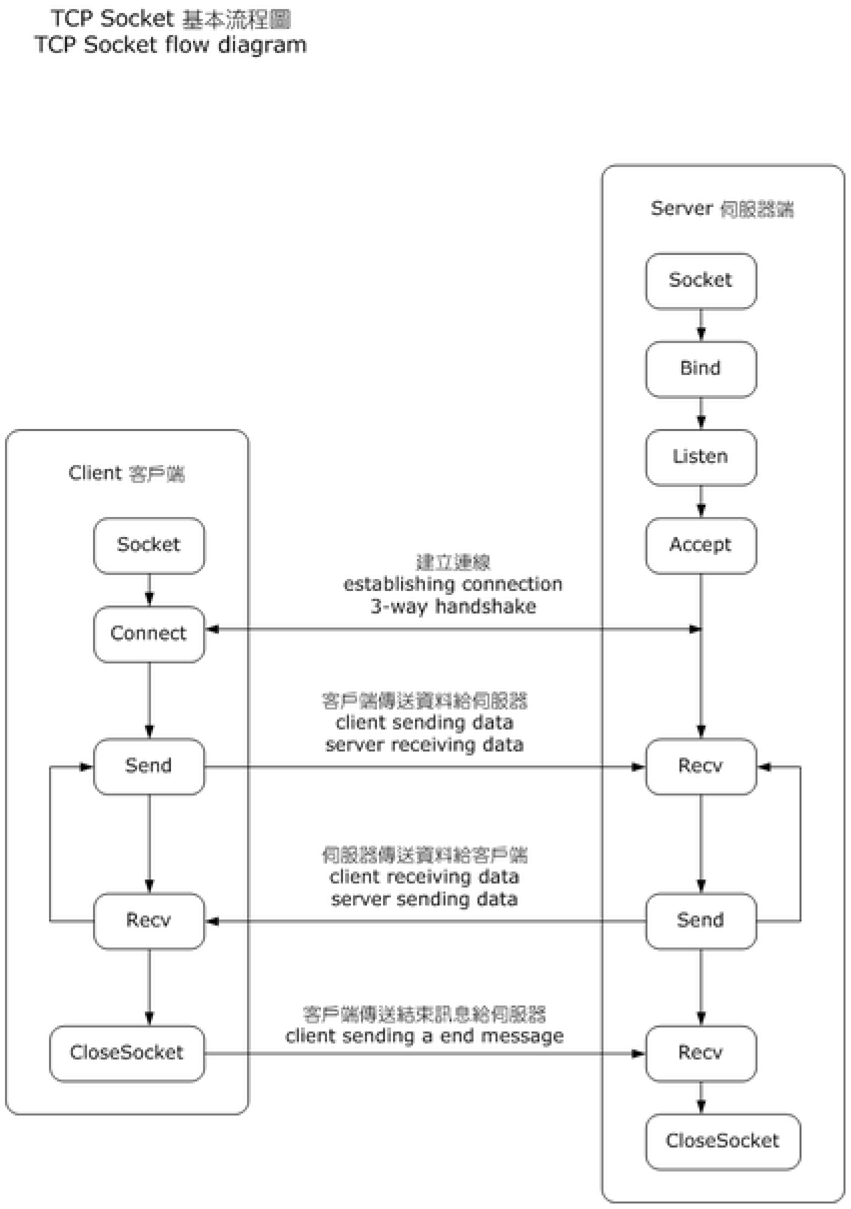
For the below defined socket object,

```
s = socket.socket(socket_family, socket_type, protocol=0)
```

TCP Socket Methods	UDP Socket Methods
s.recv()→ Receives TCP messages	s.recvfrom()→ Receives UDP messages
s.send()→ Transmits TCP messages	s.sendto()→ Transmits UDP messages

Some Basic Socket Methods:

- close() This method is used to close the socket connection.
- gethostname() This method returns a string containing the hostname of the machine where the python interpreter is currently executing. For example: localhost.
- gethostbyname() If you want to know the current machine's IP address, you may use gethostbyname(gethostname()).



Flow of socket API calls

TKinter Module GUI

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications.

Tkinter is the Python port for Tcl-Tk GUI toolkit developed by Fredrik Lundh.

There are two main methods used which the user needs to remember while creating the Python application with GUI.

1. **Tk(screenName=None, baseName=None, className='Tk', useTk=1):** To create a main window, tkinter offers a method 'Tk(screenName=None, baseName=None, className='Tk', useTk=1)'. To change the name of the window, you can change the className to the desired one. The basic code used to create the main window of the application is:
   m=tkinter.Tk() //where m is the name of the main window object
2. **mainloop():** There is a method known by the name mainloop() is used when your application is ready to run. mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event as long as the window is not closed.
   m.mainloop()

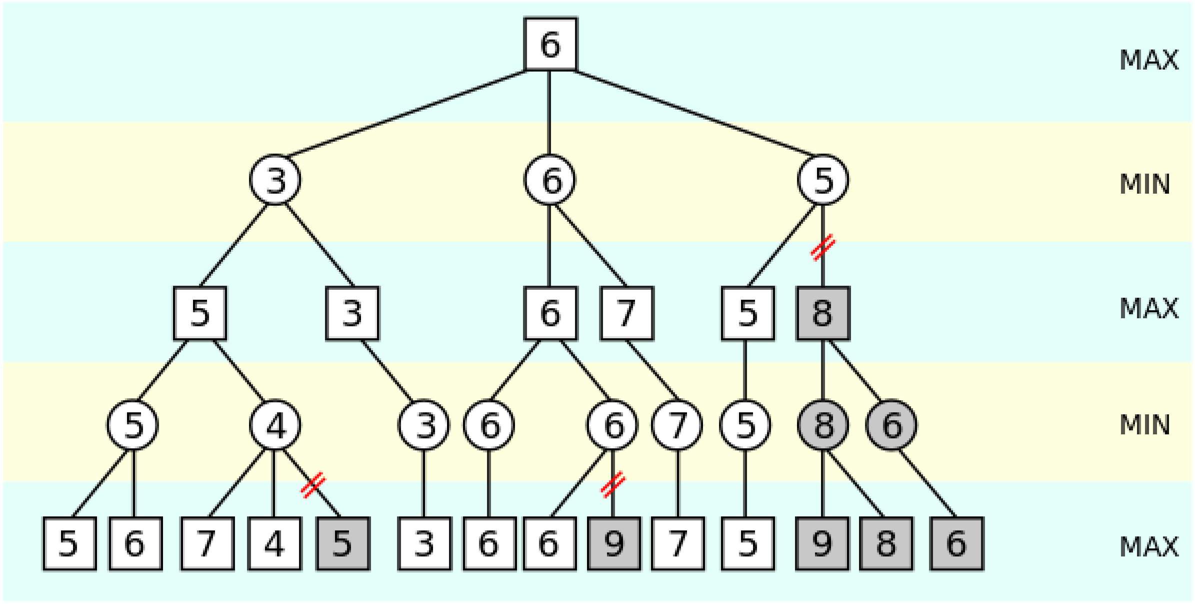
Tkinter also offers access to the geometric configuration of the widgets which can organize the widgets in the parent windows. There are mainly three geometry manager classes.

1. **pack() method:**It organizes the widgets in blocks before placing in the parent widget.
2. **grid() method:**It organizes the widgets in grid (table-like structure) before placing in the parent widget.
3. **place() method:**It organizes the widgets by placing them on specific positions directed by the programmer.

Data Flair
Python
GUI Programming with Tkinter
01
Options for Python GUI Programming
02
Tkinter
03
Widgets

Alpha-Beta Pruning:

- Alpha-beta pruning is a modified version of the minimax algorithm. It is an optimization technique for the minimax algorithm.
- As we have seen in the minimax search algorithm that the number of game states it has to examine are exponential in depth of the tree. Since we cannot eliminate the exponent, we can cut it to half. Hence there is a technique by which without checking each node of the game tree we can compute the correct minimax decision, and this technique is called **pruning**. This involves two threshold parameters Alpha and beta for future expansion, so it is called **alpha-beta pruning**. It is also called the Alpha-Beta **Algorithm**.
- Alpha-beta pruning can be applied at any depth of a tree, and sometimes it not only prunes the tree leaves but also the entire sub-tree.
- The two-parameter can be defined as:
    - **Alpha:** The best (highest-value) choice we have found so far at any point along the path of Maximizer. The initial value of alpha is **-∞**.
    - **Beta:** The best (lowest-value) choice we have found so far at any point along the path of Minimizer. The initial value of beta is **+∞**.



It is an adversarial search algorithm used commonly for machine playing of two-player games (Tic-tac-toe, Chess, Go, etc.). It stops evaluating a move when at least one possibility has been found that proves the move to be worse than a previously examined move. Such moves need not be evaluated further. When applied to a standard minimax tree, it returns the same move as minimax would, but prunes away branches that cannot possibly influence the final decision.

The algorithm maintains two values, alpha and beta, which respectively represent the minimum score that the maximizing player is assured of and the maximum score that the minimizing player is assured of. Initially, alpha is negative infinity and beta is positive infinity, i.e. both players start with their worst possible score. Whenever the maximum score that the minimizing player (i.e. the "beta" player) is assured of becomes less than the minimum score that the maximizing player (i.e., the "alpha" player) is assured of (i.e. beta < alpha), the maximizing player need not consider further descendants of this node, as they will never be reached in the actual play.

SQLite DataBase

SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. SQLite is the most used database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day.

SQLite is a relational database management system (RDBMS) contained in a C library. In contrast to many other database management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program.

SQLite is ACID-compliant and implements most of the SQL standard, generally following PostgreSQL syntax. However, SQLite uses a dynamically and weakly typed SQL syntax that does not guarantee the domain integrity.

SQLite is a popular choice as embedded database software for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems

Suggested Uses For SQLite:

- **Database For The Internet Of Things.** SQLite is popular choice for the database engine in cellphones, PDAs, MP3 players, set-top boxes, and other electronic gadgets. SQLite has a small code footprint, makes efficient use of memory, disk space, and disk bandwidth, is highly reliable, and requires no maintenance from a Database Administrator.

- **Application File Format.** Rather than using fopen() to write XML, JSON, CSV, or some proprietary format into disk files used by your application, use an SQLite database. You'll avoid having to write and troubleshoot a parser, your data will be more easily accessible and cross-platform, and your updates will be transactional. (more...)
- **Website Database.** Because it requires no configuration and stores information in ordinary disk files, SQLite is a popular choice as the database to back small to medium-sized websites.
- **Stand-in For An Enterprise RDBMS.** SQLite is often used as a surrogate for an enterprise RDBMS for demonstration purposes or for testing. SQLite is fast and requires no setup, which takes a lot of the hassle out of testing and which makes demos perky and easy to launch.