PYTHON

Chapter: Introduction

Introduction

Low level apis provided by – sockets
High level apis through libraries
Part of standard library

Two main info

Address :

IP address + port number

In python represented as a (host, port) combination

Transport:

TCP / UDP

Transport

There are two basic types of communication

- **Streams (TCP)**: Computers establish a connection with each other and read/write data in a continuous stream of bytes---like a file. This is the most common.

- **Datagrams (UDP)**: Computers send discrete packets (or messages) to each other. Each packet contains a collection of bytes, but each packet is separate and self-contained.

Python - Sockets

Programming abstraction for network code

- Socket: A communication endpoint

- Supported by socket library module

- Allows connections to be made and data to be transmitted in either direction

Python - Sockets

Steps :

1. Two ways to use
   1. Create a socket object
   2. Use socket directly to fetch information
2. Address Families
   1. socket.AF_INET
   2. socket.AF_INET6
3. Socket type
   1. socket.SOCK_STREAM
   2. socket.SOCK_DGRA

Sockets – Getting information

1. socket.gethostname()
2. socket.gethostbyname(‘www.google.com’)
3. socket.gethostbyaddr(**'172.217.26.164'**)
4. Get service information
   1. socket.getservbyport(port, protocolname)
   2. socket.getservbyname(‘ssh’,’tcp’)

Sockets – Client program

1. Create a socket object

   socket.socket(socket.AF_INET,socket.SOCK_STREAM)

2. Connect to an address

   s.connect(('www.google.com', 80))

3. Send information

   s.send("GET /index.html HTTP/1.0\n\n")

4. Receive Information

   data = s.recv(1000)

Sockets – Server program

1. Create a socket object

   socket.socket(socket.AF_INET,socket.SOCK_STREAM)

2. Bind to a port

   s.bind(('127.0.0.1', 9999))

3. Allow for clients to listen

   s.listen(5)

4. Accept connections

   s.accept()

5. **Send and receive**

   t[0].send, t[0].receive