

Python 101

Sockets

Sockets

Los sockets son canales bidireccionales de comunicación. Estos, pueden conectar procesos, procesos en una misma computadora o diferentes tipos de dispositivos.

Los Sockets pueden implementarse sobre diferentes tipos de canales de comunicación.

Algunos: Unix, de dominio, tcp, udp, etc.

Una biblioteca de sockets provee diferentes clases para manejar los tipos más convencionales de comunicación, así como tipos genéricos para manejar el resto.

Características de socket

Dominio: Son una familia de protocolos (AF_INET, PF_INET, PF_UNIX, PF_X25)

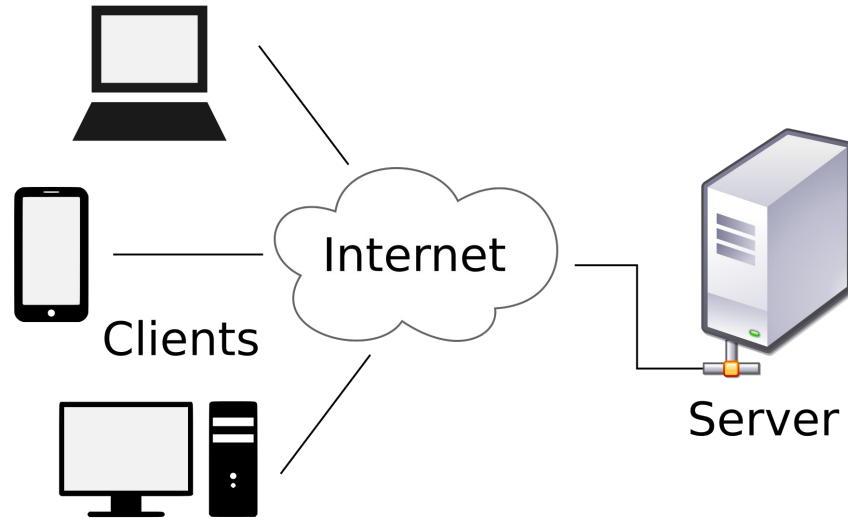
Tipo: denomina el tipo de comunicación a ser realizada, típicamente SOCK_STREAM para casos “orientados a conexión” y SOCK_DGRAM orientados a no conexión.

Protocolo: típicamente 0, sirve para identificar variantes de un protocolo/tipo

Host: identifica la interfaz de red (un string: ipv4, ipv6, un broadcast address)

Puerto: Cada servidor/servicio escucha en uno o más puertos por cliente.
(Algunos puertos bien conocidos son 53 domserver, 22 ssh, 21 ftp, 80 http, 443 https, etc.

Diagrama clásico cliente - servidor



Server - SIDE

Método	Descripción
s.bind()	Este método configura la tupla (hostname, port) al socket.
s.listen()	Configura e inicia un "listener" TCP
s.accept()	De manera pasiva acepta una conexión TCP. Es pasivo y bloqueante.

Client - SIDE

Método	Descripción
s.connect()	Activamente inicia una conexión TCP

Métodos generales

Method	Description
s.recv()	Recibe mensaje TCP
s.send()	Emite mensaje TCP
s.recvfrom()	Recibe mensaje UDP
s.sendto()	Transmite mensaje UDP
s.close()	Termina socket
socket.gethostname())	Retorna hostname

Protocol	Common function	Port No	Python module
HTTP	Web pages	80	httplib, urllib, xmlrpclib
NNTP	Usenet news	119	nntplib
FTP	File transfers	20	ftplib, urllib
SMTP	Sending email	25	smtplib
POP3	Fetching email	110	poplib
IMAP4	Fetching email	143	imaplib
Telnet	Command lines	23	telnetlib
Gopher	Document transfers	70	gopherlib, urllib