

# socket类型

## TCP和UDP的区别[Socket]

```
AF_UNIX[本机通信)
AF_INET[TCP/IP - IPv4]
AF_INET6[TCP/IP - IPv6]
其中 "type" 参数指的是套接字类型，常用的类型有：
SOCK_STREAM[TCP流)
SOCK_DGRAM[UDP数据报)
SOCK_RAW[原始套接字)
```

<https://www.cnblogs.com/sanjianghuiiu/p/4653906.html>

流套接字[SOCK\_STREAM] 流套接字用于提供面向连接、可靠的数据传输服务。该服务将保证数据能够实现无差错、无重复发送，并按顺序接收。流套接字之所以能够实现可靠的数据服务，原因在于其使用了传输控制协议，即TCP[The Transmission Control Protocol]协议。

数据报套接字[SOCK\_DGRAM] 数据报套接字提供了一种无连接的服务。该服务并不能保证数据传输的可靠性，数据有可能在传输过程中丢失或出现数据重复，且无法保证顺序地接收到数据。数据报套接字使用UDP[User Datagram Protocol]协议进行数据的传输。由于数据包套接字不能保证数据传输的可靠性，对于有可能出现的数据丢失情况，需要在程序中做相应的处理。

原始套接字(SOCK\_RAW) 原始套接字与标准套接字（标准套接字指的是前面介绍的流套接字和数据报套接字）的区别在于：原始套接字可以读写内核没有处理的IP数据包，而流套接字只能读取TCP协议的数据，数据报套接字只能读取UDP协议的数据。因此，如果要访问其他协议发送数据必须使用原始套接字。

## Socket类型

### 名字/常量 描述

- SOCK\_STREAM 这个协议是按照顺序的、可靠的、数据完整的基于字节流的连接。这是一个使用最多的socket类型，这个socket是使用TCP来进行传输。
- SOCK\_DGRAM 这个协议是无连接的、固定长度的传输调用。该协议是不可靠的，使用UDP来进行它的连接。
- SOCK\_SEQPACKET 这个协议是双线路的、可靠的连接，发送固定长度的数据包进行传输。必须把这个包完整的接受才能进行读取。
- SOCK\_RAW 这个socket类型提供单一的网络访问，这个socket类型使用ICMP公共协议[ping]traceroute使用该协议)
- SOCK\_RDM这个类型是很少使用的，在大部分的操作系统上没有实现，它是提供给数据链路层使用，不保证数据包的顺序

## 1. python socket

```
socket.socket(family=AF_INET, type=SOCK_STREAM, proto=0, fileno=None)
```

Create a new socket using the given address family, socket type and protocol number. The address family should be AF\_INET (the default), AF\_INET6, AF\_UNIX, AF\_CAN, AF\_PACKET, or AF\_RDS. The socket type should be SOCK\_STREAM (the default), SOCK\_DGRAM, SOCK\_RAW or perhaps one of the

other SOCK\_ constants. The protocol number is usually zero and may be omitted or in the case where the address family is AF\_CAN the protocol should be one of CAN\_RAW, CAN\_BCM or CAN\_ISOTP.