

端口: 按一定规律分配

知名端口: 0~1023

HTTP: 80 ; FTP: 21

动态端口: 1024~65535

查看端口: netstat -an

Isot -i [tcp/udp]: 2425

socket:

网络层可用"IP地址"唯一标识网络中的主机,  
传输层可用"协议+端口"唯一标识主机中的应用进程  
IP地址、协议、端口可标识网络的进程

创建 socket:

```
import socket
socket.socket(AddressFamily, Type)
```

AddressFamily: AF\_INET / AF\_INET6 (IPv4 / IPv6)  
AF\_UNIX (同一台机器进程)

Type: SOCK\_STREAM (流式, TCP)  
SOCK\_DGRAM (数据报, UDP)

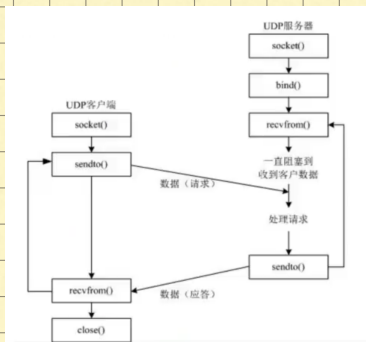
创建一个tcp socket (tcp套接字)

```
import socket
# 创建tcp的套接字
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# ...这里使用套接字的功能 (省略) ...
# 不用的时候, 关闭套接字
s.close()
```

创建一个udp socket (udp套接字)

```
import socket
# 创建udp的套接字
s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
# ...这里使用套接字的功能 (省略) ...
# 不用的时候, 关闭套接字
s.close()
```

基于udp网络程序



发数据

```
#coding=utf-8
from socket import *
# 1. 创建udp套接字
udp_socket = socket(AF_INET, SOCK_DGRAM)
# 2. 准备接收方的地址
# '192.168.1.103' 表示目的ip地址
# 8080表示目的端口
dest_addr = ('192.168.1.103', 8080) # 注意 是元组, ip是字符串, 端口是数字
# 3. 从键盘获取数据
send_data = input("请输入要发送的数据:")
# 4. 发送数据到指定的电脑上的指定程序中
udp_socket.sendto(send_data.encode('utf-8'), dest_addr)
# 5. 关闭套接字
udp_socket.close()
```

收数据

```
#coding=utf-8
from socket import *
# 1. 创建套接字
udp_socket = socket(AF_INET, SOCK_DGRAM)
# 2. 绑定本地的相关信息, 如果一个网络程序不绑定, 则系统会随机分配
local_addr = ('', 7788) # ip地址和端口号, ip一般不用写, 表示本机的任何一个ip
udp_socket.bind(local_addr) # 绑定时必须指定本机ip和port
# 3. 等待接收对方发送的数据
recv_data = udp_socket.recvfrom(1024) # 1024表示本次接收的最大字节数
# recv_data 中存储的是一个元组(接收到的数据, (发送的ip, port))
# 4. 显示接收到的数据
print(recv_data[0].decode('gbk'))
#print("%s:%s"%(str(recv_data[1]), recv_data[0].decode('gbk'))))
# 5. 关闭套接字
udp_socket.close()
```