网络编程的三个要素：

Ip、端口、协议



**Tcp/ip 是以tcp/ip为基础的不同层次上多个协议的集合**

**Tcp/ip又叫做Tcp/ip协议家族、Tcp/ip协议栈**

**Tcp:传输控制协议**

**Ip: 网络协议**

**通常我们再使用过程中会将网络进行分层：**

**常见的会将网路分为五层：物理层、数据链路层、网路层、传输层（tcp/ip协议所在）、应用层（http超文本传输协议、ftp文件传输协议、telnet远程登陆服务，也就是面向应用的协议） 也叫tcp/ip的五层模型**

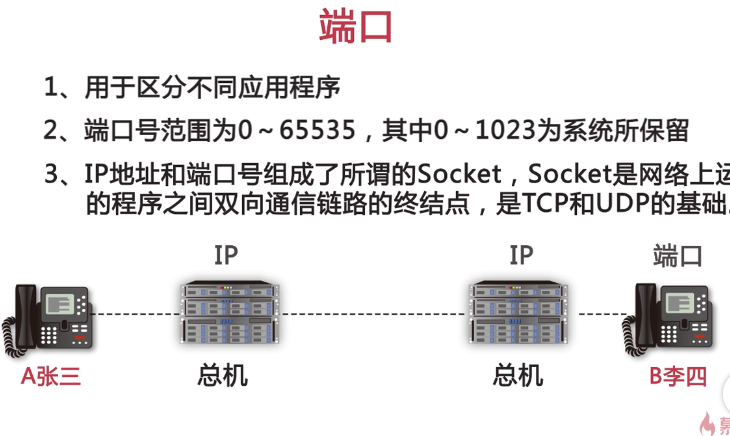


**Ip地址**相当于是每一台电脑的标识，相当于每个人的手机号

目前使用较多的是**ipV4** 也就是ip的第四个版本，在这个版本中定义**ip地址**的长度为32位的二进制组成，也就是类似于192.168.0.1

**端口：**用于区分一台电脑上不同的应用程序，端口号范围为：0~65536，其中0~1023为系统所保留；

Ip和端口号就组成了所谓的Socket,Socket是网络运行的程序之间的双向通信链路的终结点（也叫终端），是整个网络通信的基础，也是TCP和UDP的基础

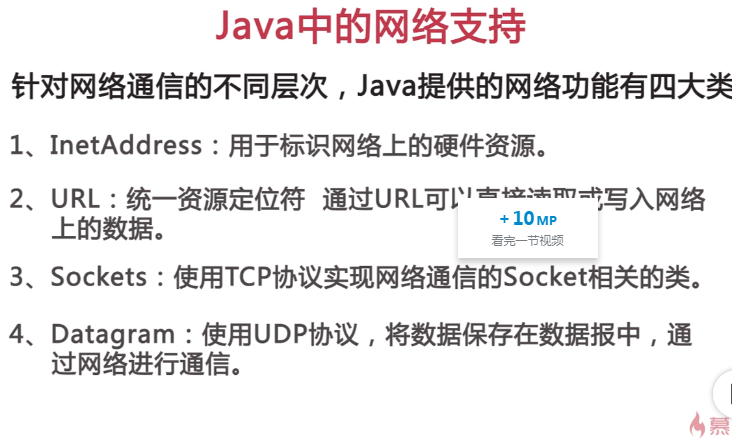


Java对网络通讯提供了良好的支持，针对网络的不同层次提供了相应的API

也就是java在网络通信中支持的四大类：

第一个：其实就是标识ip地址

最后一个：数据报：是基于UDP协议实现的网络通信，发送请求时通过将数据报保存在数据报中，通过网络进行通信



Inetaddress类的API（关于ip地址的）

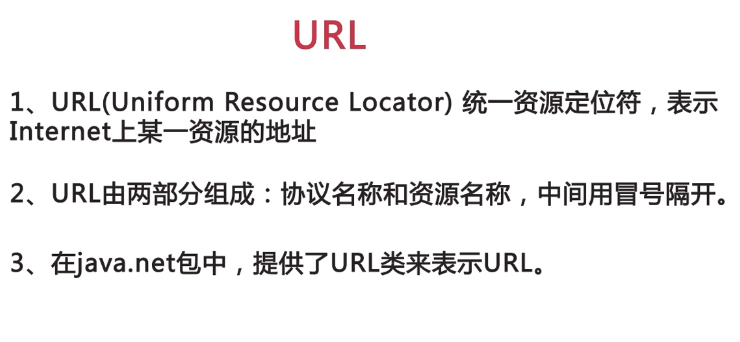
补充一点：Inetaddress他的类API里没有提供构造方法，但是提供了静态的方法用来返回Inetaddress 实例



URL：表示网络中某一资源的地址

他有两部分组成：协议名称+资源名称，中间用：隔开

在我看来就是浏览器中的地址，协议+域名





URL API文档：

构造方法：

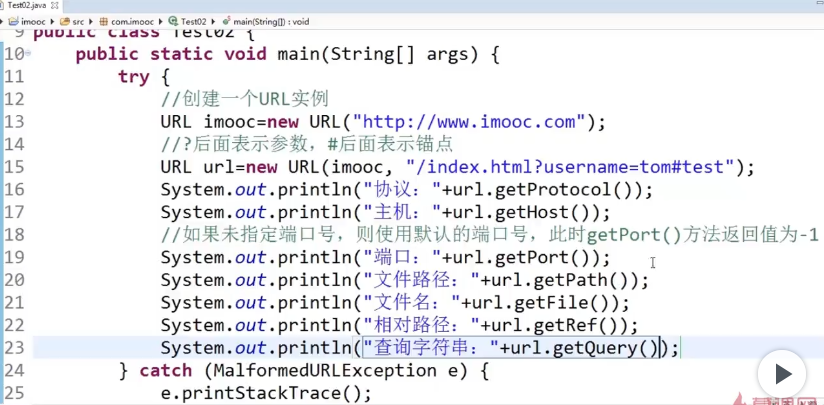


方法：



**#号后可以看成是页面位置标识符**

如果未指定端口，那么会使用协议**默认的端口号**，但是调用URL的**getport 方法**获取端口时**会返回-1**



结果：



使用URL的openstream读取页面上的数据



下面重点来了，基于TCP协议的Socket 网络通信

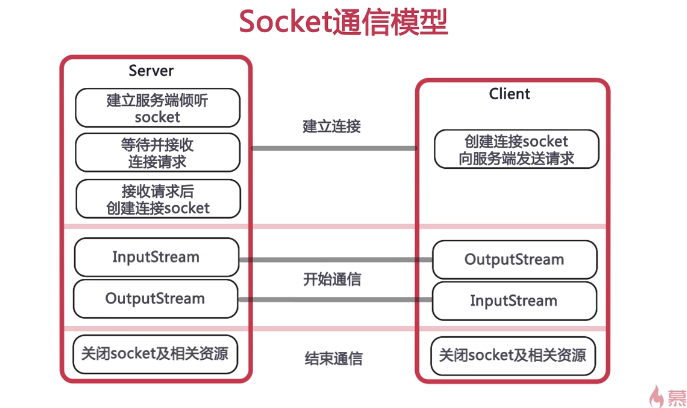
**TCP优点：**

**是面向连接的、可靠的、发送的数据是有顺序的、并且以字节流的方式发送数据**

发送数据需要建立连接，保证了数据可靠性

实现通信的有两个类：**服务器端（ServerSocket类）和客户端（Socket类）**

首先Socket的通信模型



Socket的通信步骤：

1

2

3

4

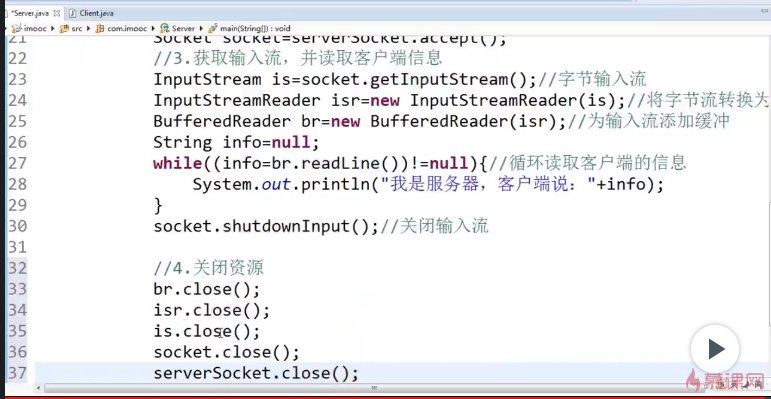


记着最后的关闭资源：：：：：

**都是先关闭Socket的输入输出流**

服务端 serverSocket





客户端： Socket



**多线程Socket连接通信**

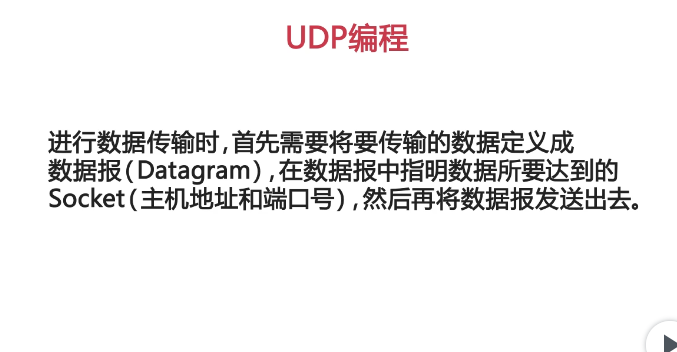
**通过将每个客户端生成的Socket单独用一个线程来进行通信，并且serverSocket进行循环监听客户端连接**

**UDP（datagram数据报传输）**

**缺点：无连接、不可靠、无顺序（不固定）**

**优点：传输的速度快**

**将数据报作为数据的载体进行数据的传输，速度快（datagram）**



**Datagram有关的两个类**

**DatagramPacket （**数据报报也叫数据报类，用来表示UDP通信中的数据单元**）**

以下为构造方法接收和发送数据



**DatagramSocket （表示用来发送和接收数据报包的Socket）**

他的构造器可以绑定到指定ip、端口也可以不指定，会匹配本地到所有可用的端口

