

# Web和网络

---

刘挺立

Liu.Tingli@outlook.com

QQ: 1819181820

# 讲师介绍



## 刘挺立

---

8年软件开发和自动化测试经验

敏捷测试项目管理专家

曾长期供职于外企，担任高级软件开发、数据挖掘以及自动化测试等职位

2014年加入泽林咨询，从事测试开发研究

# 学习目标

通过本课学习，您将拥有以下收获：

1. 掌握计算机网络基础知识
2. 了解Web系统的概念与组成
3. 掌握Web系统的基础元素
4. 初步理解XAMPP网站服务器架构

# 目录

1

计算机网络基础

2

认识 Web系统

3

XAMPP 基本介绍



# 计算机网络基础

# 计算机网络基础

- 计算机网络的定义
- 网络通信协议
- 网络通信基本对象



# 计算机网络的定义

# 计算机网络的定义

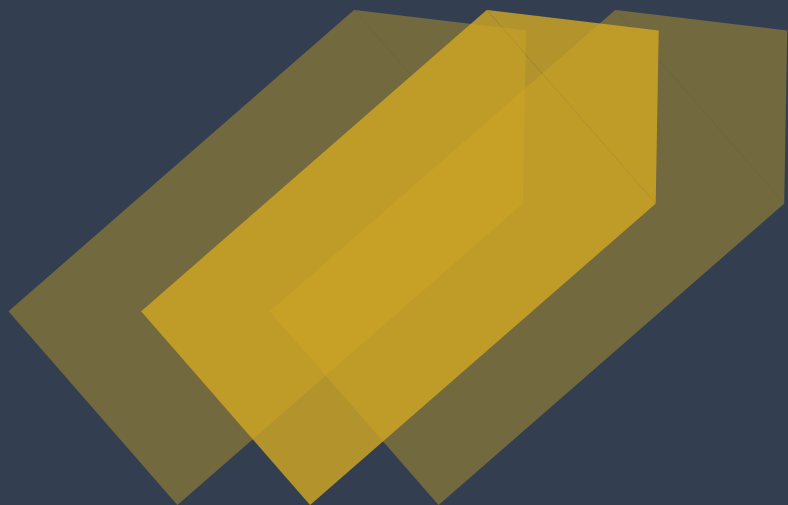
- 将地理位置不同的具有独立功能的多台计算机及其外部设备，通过通信线路连接起来，在网络操作系统，网络管理软件及网络通信协议的管理和协调下，实现资源共享和信息传递的计算机系统。
- 组成
  - 计算机
  - 网络操作系统
  - 传输介质
  - 应用软件





# 计算机网络的定义

• 右侧的哪些不是计算机网络？



- 计算机？
- 网络工作站？
- 路由器？
- 交换机？
- 通信线路？
- QQ软件？
- 网卡？

# 网络通信协议

# 网络通信协议

- 通信协议

- 双方实体完成通信所必须遵循的规则和约定
- 数据单元使用的格式
- 信息单元信息与含义
- 连接方式
- 信息发送和接收的时序

- 三要素

- 语法

- 如何讲
- 数据的格式
- 编码
- 信号等级

- 语义

- 讲什么
- 内容
- 控制信息

- 定时规则

# 网络通信协议

- 层次性
  - 复杂的问题分解，下层为上层服务
- 可靠性
- 有效性

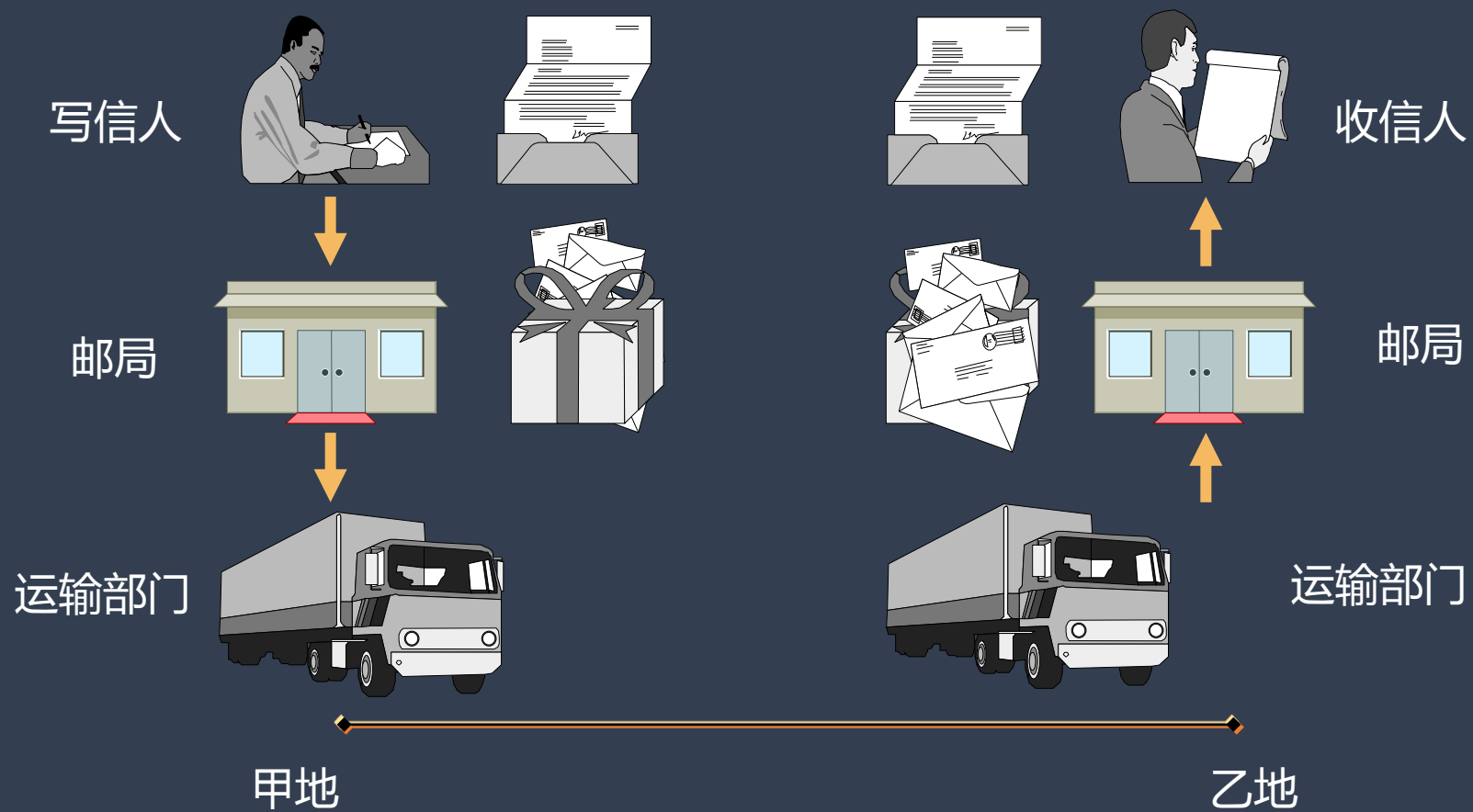


# 网络通信协议

- OSI模型
  - OSI
  - Open System Interconnection
  - 开发系统互联
  - ISO 1977年提出OSIRM (Reference Model)
- TCP/IP协议
  - TCP/IP
  - Transmission Control Protocol / Internet Protocol
  - 网络通讯协议



# 网络通信协议



# 网络通信协议

应用层

表示层

会话层

传输层

网络层

数据链路层

物理层

APDU

- 应用协议数据单元

PPDU

- 表示层协议数据单元

SPDU

- 会话层协议数据单元

TPDU

- 传输层协议数据单元

报文

- 解析帧

帧

- 一段有限的比特流

比特

- 比特流，0/1

# 网络通信协议

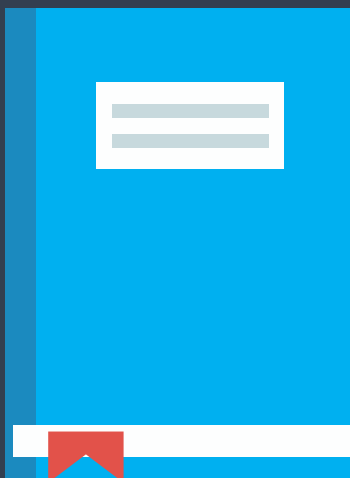
- 下层为上层服务
- 同一层使用同一协议



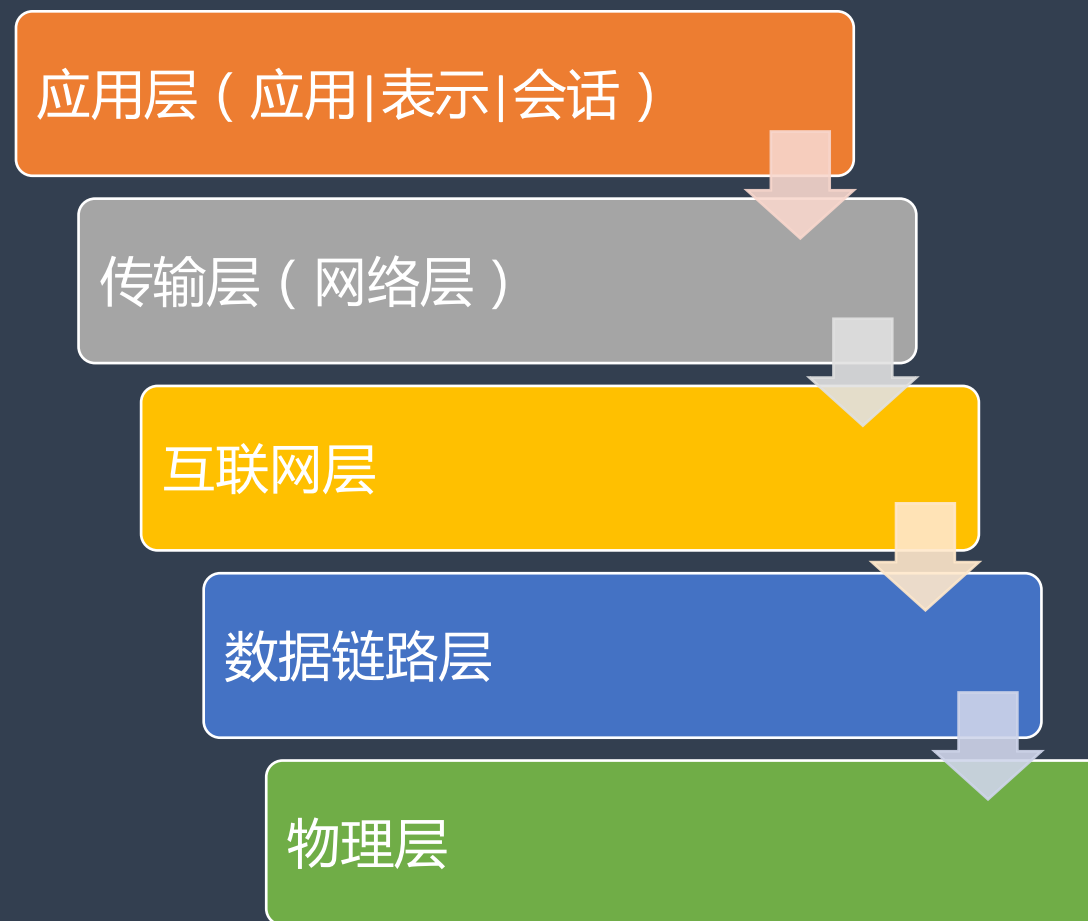
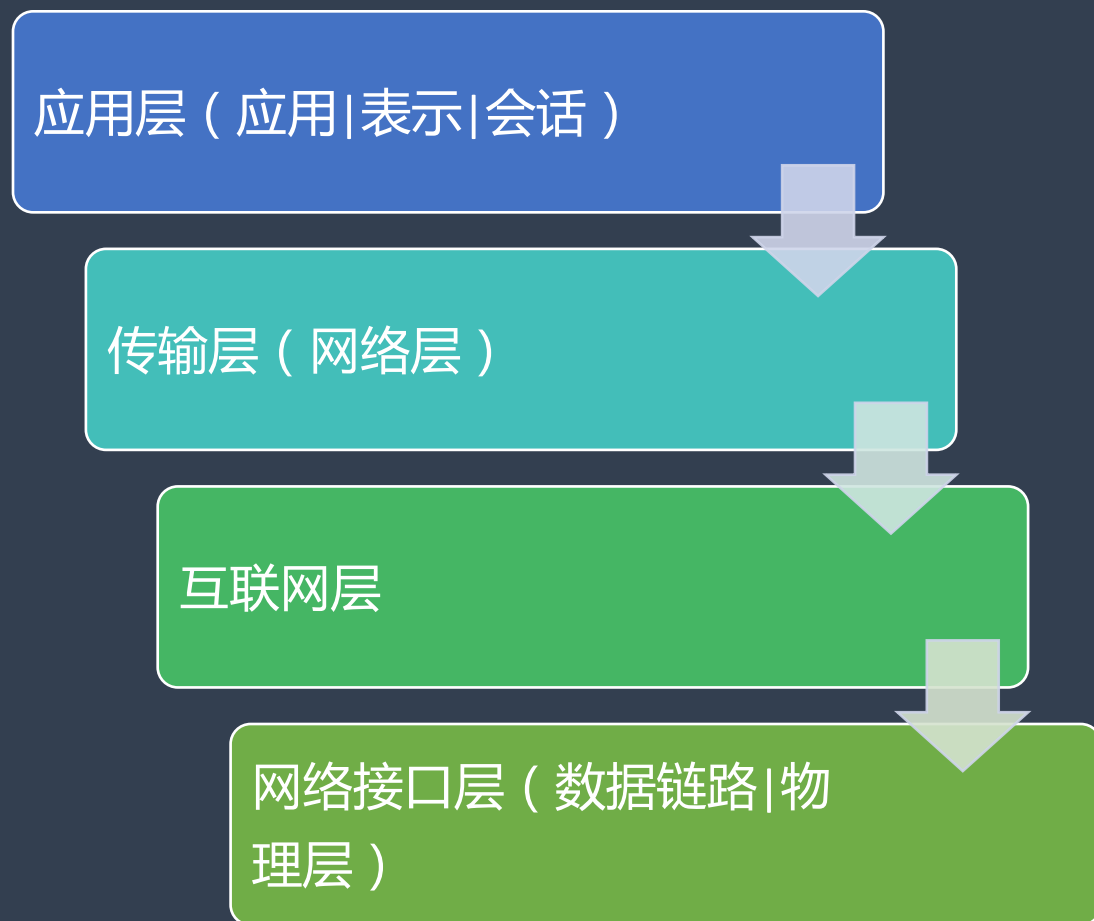
# 网络通信协议

- TCP/IP
  - TCP/IP是20世纪70年代中期美国国防部为ARPANET开发的网络体系结构

- TCP/IP 五层
- TCP/IP 四层



# 网络通信协议



# 网络通讯协议

## • TCP/IP协议历史

- 1972年，罗伯特·卡恩（ Robert E. Kahn ）被DARPA的信息技术处理办公室雇佣
- 1973年夏天，卡恩和瑟夫很快就开发出了一个基本的改进形式
- 1975年，两个网络之间的TCP/IP通信在斯坦福和伦敦大学学院（ UCL ）之间进行了测试
- 1978年到1983年间，其他一些TCP/IP原型在多个研究中心之间开发出来。
- 1984年，美国国防部将TCP/IP作为所有计算机网络的标准。

## • OSI模型历史

- 世界上第一个网络体系结构由IBM公司提出（ 74年，SNA ）
- 以后其他公司也相继提出自己的网络体系结构如：Digital公司的DNA，美国国防部的TCP/IP
- 国际标准化组织ISO于1977年成立了一个委员会，在现有网络的基础上，提出了不基于具体机型、操作系统或公司的网络体系结构，称为开放系统互联模型

# 网络通信协议

- 两者都是以协议栈的概念为基础
- 协议栈中的协议彼此相互独立
- 下层对上层提供服务
- OSI是先有模型；TCP/IP是先有协议，后有模型
- OSI适用于各种协议栈；TCP/IP只适用于TCP/IP网络
- 层次数量不同
- TCP/IP是由一些交互性的模块做成的分层次的协议，其中每个模块提供特定的功能；OSI则指定了哪个功能是属于哪一层的。

# 网络通信协议

- TCP/IP的传输层两个协议
  - UDP
    - 用户数据报协议
    - User Datagram Protocol
    - 无连接的服务
  - TCP
    - 传输控制协议
    - Transmission Control Protocol
    - 面向连接的服务



# 网络通信协议

- 面向连接的服务

- 先建立连接再传输数据，之后再断开连接
- 数据传输过程中，数据包不需要携带目的地址
- 保证数据传输的可靠性

- 无连接的服务

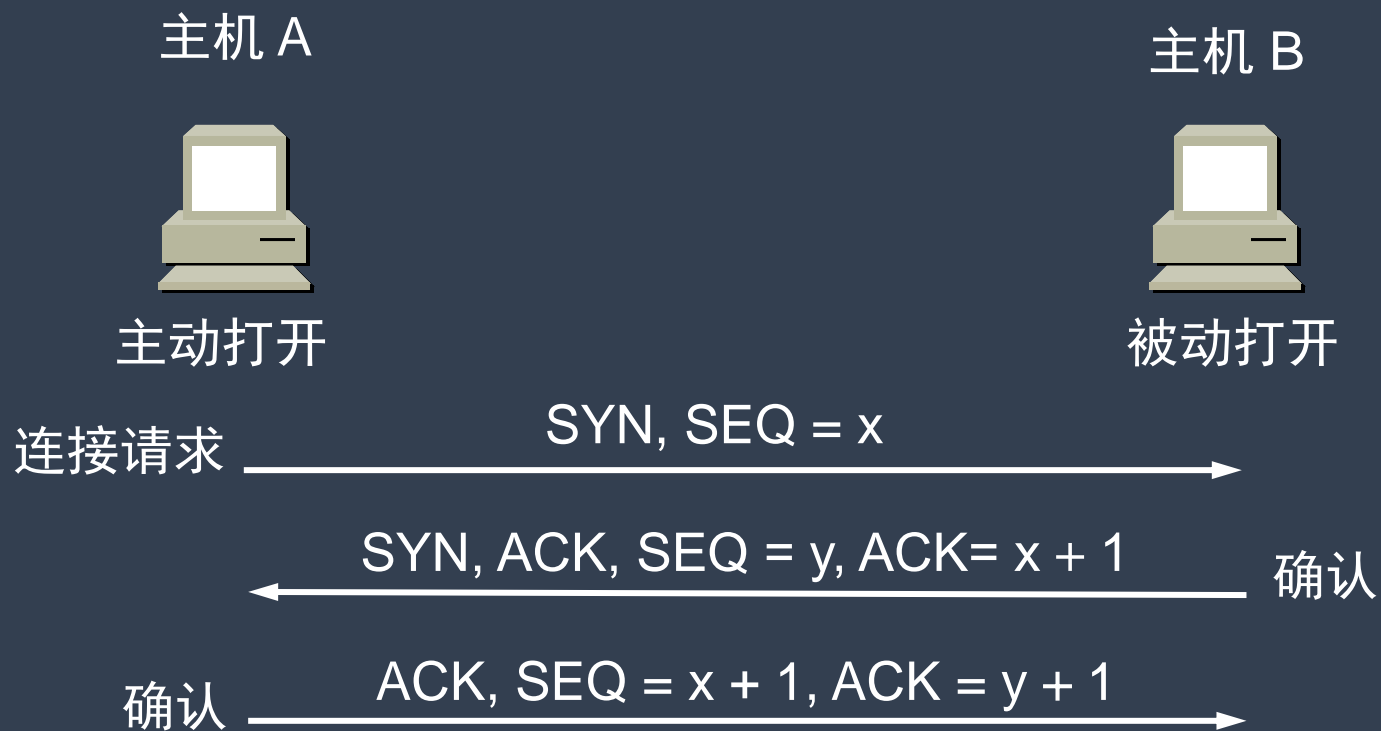
- 不需要事先建立连接，直接发送数据
- 每个报文都带有完整的目的地址
- 不保证报文传输的可靠性

# 网络通信协议

- 小结TCP与UDP的区别：
  - 1.基于连接与无连接；
  - 2.对系统资源的要求（TCP较多，UDP少）；
  - 3.UDP程序结构较简单；
  - 4.流模式与数据报模式；
  - 5.TCP保证数据正确性，UDP可能丢包，TCP保证数据顺序，UDP不保证。

# 网络通信协议

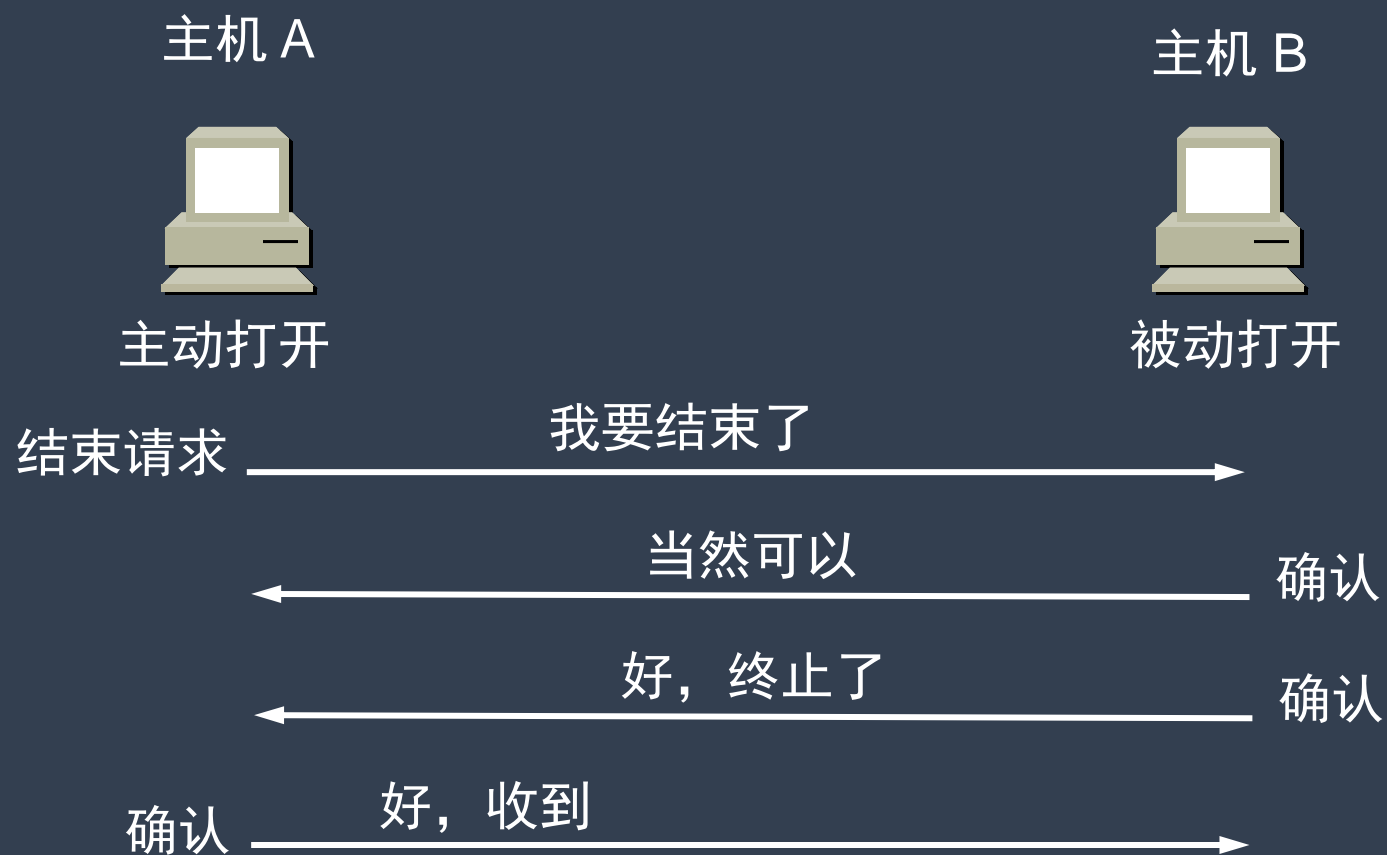
- TCP三次握手 开始





# 网络通信协议

- TCP四次握手结束



# 网络通信基本对象

# 网络通信基本对象

- 都有哪些？
- 端口
- IP地址
- DNS



# 网络通信基本对象

## 端口

- 端口的作用就是让应用层的各种应用进程都能将其数据通过端口向下交付给运输层，以及让运输层知道应当将其报文段中的数据向上通过端口交付给应用层相应的进程。

## 运输层服务访问点(TSAP)

从这个意义上讲，端口是用来标志应用层的进程

# 网络通信基本对象

- IP地址

- 网络号确定计算机所在的网络，主机号确定计算机在该网络中的所处的位置。
- 在INTERNET网中，根据TCP/IP协议规定，每个IP地址是由32bit的二进制数组成的。主要分为三类：



## IP Address Classes

- Class A: 

N	H	H	H
---	---	---	---
- Class B: 

N	N	H	H
---	---	---	---
- Class C: 

N	N	N	H
---	---	---	---
- Class D: for multicast
- Class E: for research

N = Network number assigned by NIC

H = Host number assigned by network administrator

6

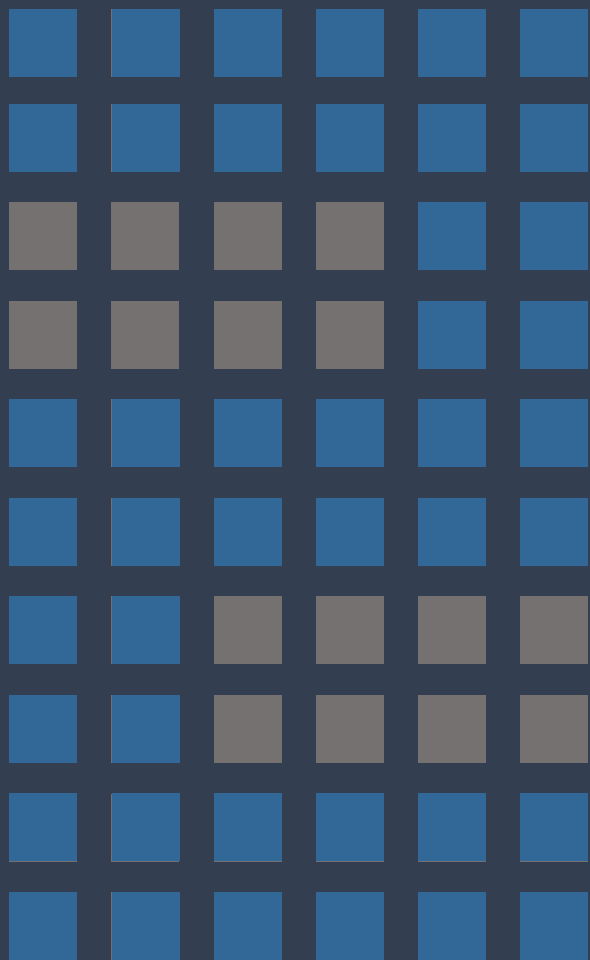
# 网络通信基本对象

- 保留地址：
  - 10.0.0.0—10.255.255.255
  - 172.16.0.0—172.31.255.255
  - 192.168.0.0—192.168.255.255

特征位	二进制表示的前八位码组范围	十进制表示的前八位码组范围	地址类别	地址举例
0	01111111	0—127	A类	98.31.57.88
10	10000000—10111111	128—191	B类	134.160.133.100
110	11000000—11011111	192—223	C类	218.30.66.101

# 网络通信基本对象

- DNS
  - Domain Name Server ( 域名服务器 ) : 域名服务器有两大类型Authoritative Name Server ( 权威域名服务器 ) 和Caching Name Server ( 存储缓冲域名服务器 )。权威域名服务器又包含了主服务器、辅服务器等子类型。
  - 存储缓冲域名服务器 : 为了节省查询时间, 提高性能, 支持递归查询的域名服务器通常会在一定的时间范围内将查询结果放入高速缓存, 使用这种方式的域名服务器称为存储缓冲域名服务器。
- The Primary Master ( 主服务器 ) / Slave Server ( 辅服务器 ) :
  - 每个区至少有一台权威域名服务器负责解析工作, 考虑到服务器和网络的不稳定性, 通常把区交由两台或多台权威域名服务器解析。其中的一台被指定为主服务器, 其它的则为辅服务器。区完整的配置信息保存在主服务器上, 辅服务器会定期将主服务器上区的资料复制到本机。



# 认识 Web系统



# 认识Web系统

- Web系统的基本介绍
- HTTP协议和状态码
- Get请求和Post请求
- Cookie和Session



# Web系统的基本介绍

# Web系统的基本介绍

- Web系统
- 万维网
  - 一种网络使用环境又是一些相关技术的总称
- 技术角度看WEB
  - 从技术的角度来看，Web技术包含网站网页的布局设计、代码编写、数据库建立、网络平台选择等相关技术
- 使用角度看WEB
  - 从网络使用环境的角度来看，Web就是我们平常使用浏览器上网时所浏览的网页

# Web系统的基本介绍

- B/S架构

- B/S(Browser/Server)指浏览器和服务器端，在客户端不用装专门的软件，只要一个浏览器即可

- C/S架构

- C/S(Client/Server)指客户机和服务器，在客户端必须装客户端软件后才能访问服务器，如QQ、飞信等

# HTTP协议和状态码

# HTTP协议和状态码

- HTTP协议
  - Hypertext Transfer Protocol
  - 超文本转移协议
- HTTP是一个应用层协议，由请求和响应构成，是一个标准的客户端服务器模型。HTTP是一个无状态的协议。
- 用于从WWW服务器传输超文本到本地浏览器的传送协议。它可使浏览器更加高效，使网络传输减少。它不仅保证计算机正确快速地传输超文本文档，还确定传输文档中的哪一部分，以及哪部分内容首先显示(如文本先于图形)等。

# HTTP协议和状态码

- 100-199 用于指定客户端应相应的某些动作。
- 200-299 用于表示请求成功。
- 300-399 用于已经移动的文件并且常被包含在定位头信息中指定新的地址信息。
- 400-499 用于指出客户端的错误。
- 500-599 用于支持服务器错误。

# HTTP协议和状态码

1**状态码	成功处理了请求的状态码
100	客户端应当继续发送请求。这个临时响应是用来通知客户端它的部分请求已经被服务器接收，且仍未被拒绝。客户端应当继续发送请求的剩余部分，或者如果请求已经完成，忽略这个响应。服务器必须在请求完成后向客户端发送一个最终响应。
101	服务器已经理解了客户端的请求，并将通过Upgrade 消息头通知客户端采用不同的协议来完成这个请求。在发送完这个响应最后的空行后，服务器将会切换到在Upgrade 消息头中定义的那些协议。只有在切换新的协议更有好处的时候才应该采取类似措施。例如，切换到新的HTTP 版本比旧版本更有优势，或者切换到一个实时且同步的协议以传送利用此类特性的资源。
102	由WebDAV（RFC 2518）扩展的状态码，代表处理将被继续执行。



# HTTP协议和状态码

2**状态码	成功处理了请求的状态码。
200	请求已成功，请求所希望的响应头或数据体将随此响应返回。
204	服务器已经理解了客户端的请求，并将通过Upgrade 消息头通知客户端采用不同的协议来完成这个请求。在发送完这个响应最后的空行后，服务器将会切换到在Upgrade 消息头中定义的那些协议。只有在切换新的协议更有好处的时候才应该采取类似措施。例如，切换到新的HTTP 版本比旧版本更有优势，或者切换到一个实时且同步的协议以传送利用此类特性的资源。
202	服务器已接受请求，但尚未处理。正如它可能被拒绝一样，最终该请求可能会也可能不会被执行。在异步操作的场合下，没有比发送这个状态码更方便的做法了。返回202状态码的响应的目的是允许服务器接受其他过程的请求（例如某个每天只执行一次的基于批处理的操作），而不必让客户端一直保持与服务器的连接直到批处理操作全部完成。在接受请求处理并返回202状态码的响应应当在返回的实体中包含一些指示处理当前状态的信息，以及指向处理状态监视器或状态预测的指针，以使用户能够估计操作是否已经完成。

# HTTP协议和状态码

3**状态码	每次请求中使用重定向
301	被请求的资源已永久移动到新位置，并且将来任何对此资源的引用都应该使用本响应返回的若干个 URI 之一。注意：对于某些使用 HTTP/1.0 协议的浏览器，当它们发送的 POST 请求得到了一个301响应的话，接下来的重定向请求将会变成 GET 方式。
302	请求的资源现在临时从不同的 URI 响应请求。
304	如果客户端发送了一个带条件的 GET 请求且该请求已被允许，而文档的内容（自上次访问以来或者根据请求的条件）并没有改变，则服务器应当返回这个状态码。304响应禁止包含消息体，因此始终以消息头后的第一个空行结尾。
303	对应当前请求的响应可以在另一个 URI 上被找到，而且客户端应当采用 GET 的方式访问那个资源。这个方法的存在主要是为了允许由脚本激活的POST请求输出重定向到一个新的资源。

# HTTP协议和状态码

4**状态码	表示请求可能出错，妨碍了服务器的处理
400	1、语义有误，当前请求无法被服务器理解。除非进行修改，否则客户端不应该重复提交这个请求。 2、请求参数有误。
403	服务器已经理解请求，但是拒绝执行它。与401响应不同的是，身份验证并不能提供任何帮助，而且这个请求也不应该被重复提交。
404	请求失败，请求所希望得到的资源未被在服务器上发现。没有信息能够告诉用户这个状况到底是暂时的还是永久的。
410	被请求的资源在服务器上已经不再可用，而且没有任何已知的转发地址。这样的状况应当被认为是永久性的。

# HTTP协议和状态码

5**状态码	表示服务器在处理请求时发生内部错误
500	服务器遇到了一个未曾预料的情况，导致了它无法完成对请求的处理。一般来说，这个问题都会在服务器的程序码出错时出现。
503	由于临时的服务器维护或者过载，服务器当前无法处理请求。这个状况是临时的，并且将在一段时间以后恢复。
504	作为网关或者代理工作的服务器尝试执行请求时，未能及时从上游服务器（URI标识出的服务器，例如HTTP、FTP、LDAP）或者辅助服务器（例如DNS）收到响应。 注意：某些代理服务器在DNS查询超时时会返回400或者500错误
505	服务器不支持，或者拒绝支持在请求中使用的 HTTP 版本。这暗示着服务器不能或不愿使用与客户端相同的版本。响应中应当包含一个描述了为何版本不被支持以及服务器支持哪些协议的实体。

# Get请求和Post请求

# Get请求和Post请求

- Http定义了与服务器交互的不同方法，最基本的方法有4种，分别是GET，POST，PUT，DELETE。
- URL全称是资源描述符，我们可以这样认为：一个URL地址，它用于描述一个网络上的资源，而HTTP中的GET，POST，PUT，DELETE就对应着对这个资源的查，改，增，删4个操作。

# Get请求和Post请求

- 两种 HTTP 请求方法：GET 和 POST
  - GET - 从指定的资源请求数据。
  - POST - 向指定的资源提交要被处理的数据
- 在客户机和服务器之间进行请求-响应时，两种最常被用到的方法是：GET 和 POST。



# Get请求和Post请求

- GET 方法
- 有关 GET 请求的其他一些注释：
  - GET 请求可被缓存
  - GET 请求保留在浏览器历史记录中
  - GET 请求可被收藏为书签
  - GET 请求不应在处理敏感数据时使用
  - GET 请求有长度限制
  - GET 请求只应当用于取回数据
- 请注意，查询字符串（名称/值对）是在 GET 请求的 URL 中发送的：
  - `/test/demo_form.asp?name1=value1&name2=value2`



# Get请求和Post请求

- POST 方法
  - 有关 POST 请求的其他一些注释：
  - POST 请求不会被缓存
  - POST 请求不会保留在浏览器历史记录中
  - POST 不能被收藏为书签
  - POST 请求对数据长度没有要求
- 请注意，查询字符串（名称/值对）是在 POST 请求的 HTTP 消息主体中发送的：
  - POST /test/demo\_form.asp  
HTTP/1.1
  - Host: w3schools.com
  - name1=value1&name2=value2

# Get请求和Post请求

	Get	Post
后退按钮/刷新	无害	数据会被重新提交（浏览器应该告知用户数据会被重新提交）
书签	可收藏为书签	不可收藏为书签
缓存	能被缓存	不能缓存
编码类型	application/x-www-form-urlencoded	application/x-www-form-urlencoded 或 multipart/form-data。为二进制数据使用多重编码。
历史	参数保留在浏览器历史中。	参数不会保存在浏览器历史中。
对数据长度的限制	是的。当发送数据时，GET 方法向 URL 添加数据；URL 的长度是受限制:204	无限制。
对数据类型的限制	只允许 ASCII 字符。	没有限制。也允许二进制数据。
安全性	与 POST 相比，GET 的安全性较差，因为所发送的数据是 URL 的一部分。	POST 比 GET 更安全，因为参数不会被保存在浏览器历史或 web 服务器日志中。
可见性	数据在 URL 中对所有人都是可见的。	

# Cookie和Session

# Cookie和Session

- 会话（Session）跟踪是Web程序中常用的技术，用来跟踪用户的整个会话。

- 常用的会话跟踪技术是Cookie与Session。Cookie通过在客户端记录信息确定用户身份，Session通过在服务器端记录信息确定用户身份。



# Cookie和Session

- Cookie
- Cookie意为“甜饼”
- W3C组织提出，最早由Netscape社区发展的一种机制。
- 由于HTTP是一种无状态的协议，服务器单从网络连接上无从知道客户身份。怎么办呢？
  - 给客户端们颁发一个通行证吧，每人一个，无论谁访问都必须携带自己通行证。这样服务器就能从通行证上确认客户身份了。这就是Cookie的工作原理。



# Cookie和Session

- HTTP协议是无状态的协议。一旦数据交换完毕，客户端与服务器的连接就会关闭，再次交换数据需要建立新的连接。这就意味着服务器无法从连接上跟踪会话。

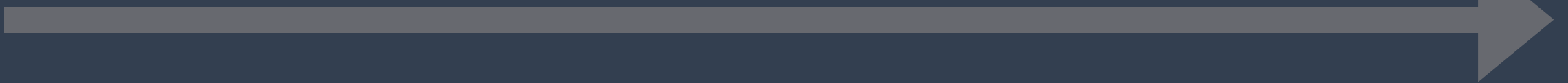


# Cookie和Session

- **Session机制**
- Session是另一种记录客户状态的机制，不同的是Cookie保存在客户端浏览器中，而Session保存在服务器上。客户端浏览器访问服务器的时候，服务器把客户端信息以某种形式记录在服务器上。

# Cookie和Session

- Cookie机制是通过检查客户身上的“通行证”来确定客户身份的话，那么Session机制就是通过检查服务器上的“客户明细表”来确认客户身份。
- Session相当于程序在服务器上建立的一份客户档案，客户来访的时候只需要查询客户档案表就可以了。





# Cookie和Session

- Session对象是在客户端第一次请求服务器的时候创建的。
- Session的使用比Cookie方便，但是过多的Session存储在服务器内存中，会对服务器造成压力。



# Cookie和Session

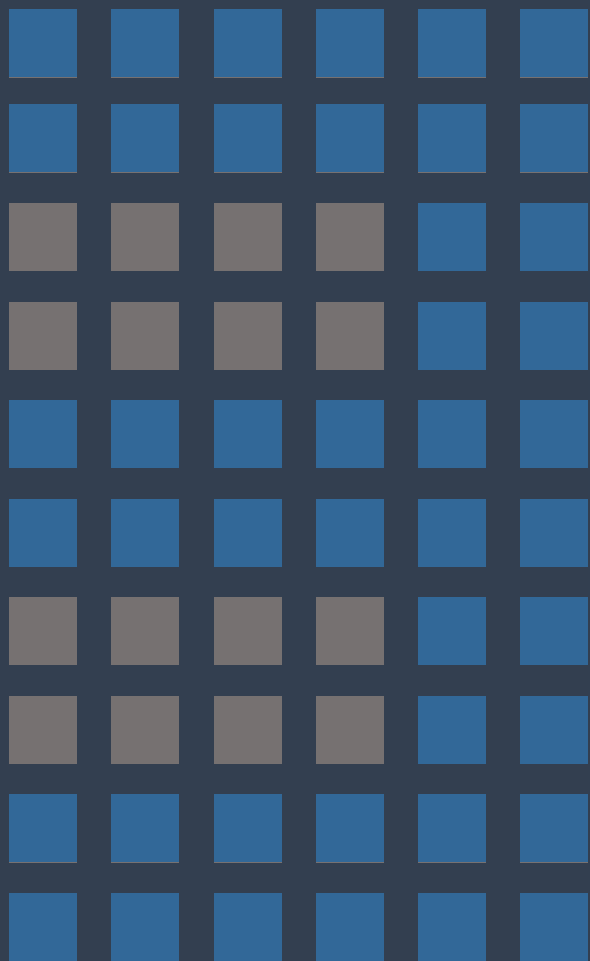
- Session保存在服务器端。
- 为了获得更高的存取速度，服务器一般把Session放在内存里。每个用户都会有一个独立的Session。如果Session内容过于复杂，当大量客户访问服务器时可能会导致内存溢出。因此，Session里的信息应该尽量精简。



# Cookie和Session

- 为防止内存溢出，服务器会把长时间内没有活跃的Session从内存删除。这个时间就是Session的超时时间。如果超过了超时时间没访问过服务器，Session就自动失效了。





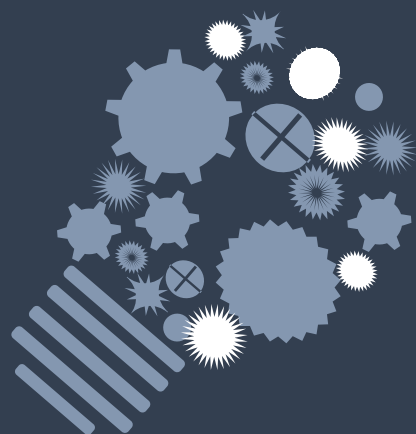
# XAMPP 基本介绍

# XAMPP的基本介绍

# XAMPP的基本介绍

- XAMPP
- Apache + MySQL + PHP + PERL

- 主流的PHP开发环境



# XAMPP的基本介绍

- 禅道是专业的项目管理软件
- 细分需求、任务、缺陷和用例
- 完整覆盖研发项目核心流程
- 软件生命周期管理

# 总结

本课学习告一段落，总结一下学习内容：

1. 计算机网络的基本概念
2. Web系统的基本认识
3. XAMPP的初步介绍



问与答

THANK YOU

Liu.Tingli@outlook.com

QQ : 1819181820

刘挺立