总体分为以下几个部分：

* DNS解析：将域名解析成IP地址；
* TCP连接：TCP三次握手；
* 发送HTTP请求；
* 服务器处理并返回HTTP报文；
* 浏览器解析渲染页面；
* 断开连接：TCP四次挥手；

URL：统一资源定位符，俗称网址；（协议：//主机.域名：端口号/服务器上的路径/资源名称）

* 域名解析：

IP 地址是指互联网协议地址，是 IP Address 的缩写。IP 地址是 IP 协议提供的一种统一的地址格式，它为互联网上的每一个网络和每一台主机分配一个逻辑地址，以此来屏蔽物理地址的差异。IP 地址是一个 32 位的二进制数，比如 127.0.0.1 为本机 IP；

域名就相当于 IP 地址乔装打扮的伪装者，带着一副面具。它的作用就是便于记忆和沟通的一组服务器的地址。用户通常使用主机名或域名来访问对方的计算机，而不是直接通过 IP 地址访问。因为与 IP 地址的一组纯数字相比，用字母配合数字的表示形式来指定计算机名更符合人类的记忆习惯。但计算机更擅长处理一长串数字。为了解决上述的问题，DNS 服务应运而生。

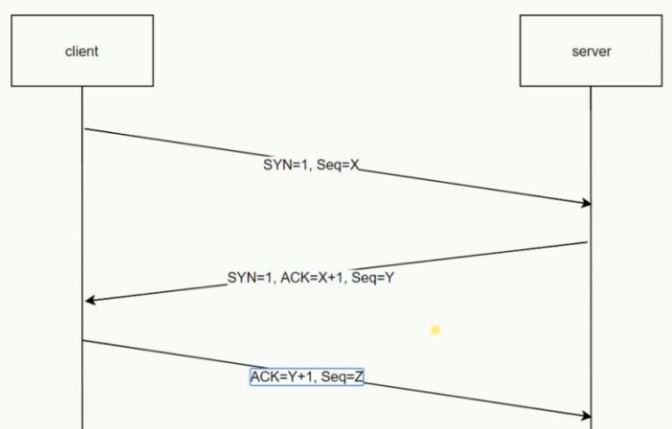
## 浏览器如何通过域名去查询 URL 对应的IP ？

* 浏览器缓存：浏览器会按照一定的频率缓存 DNS 记录。
* 操作系统缓存：如果浏览器缓存中找不到需要的 DNS 记录，那就去操作系统中找。
* 路由缓存：路由器也有 DNS 缓存。
* ISP 的 DNS 服务器：ISP 是互联网服务提供商(Internet Service Provider)的简称，ISP 有专门的 DNS 服务器应对 DNS 查询请求。
* 根服务器：ISP 的 DNS 服务器还找不到的话，它就会向根服务器发出请求，进行递归查询（DNS 服务器先问根域名服务器.com 域名服务器的 IP 地址，然后再问.baidu 域名服务器，依次类推）
* TCP三次握手：

## 客户端发送一个带有SYN = 1，Seq = X的数据包到服务器端口;（第一次握手，由浏览器发起，告诉服务器我要发送请求了）

## 服务器端回发一个带有SYN = 1，ACK = X + 1，Seq = Y的响应包以示传达确认信息；（第二次握手，由服务器发起，告诉浏览器我准备接受了，你赶紧发送吧）

## 客户端再回传一个带有ACK = Y + 1，Seq = Z的数据包，代表“握手结束”；（第三次握手，由浏览器发送，告诉服务器，我马上就发了，准备接受吧）



* 发送HTTP请求报文，请求报文包括：

## 请求行：

POST /chapter17/user.html HTTP/1.1

*请求方法 URL 协议版本*

（请求方法：GET、POST、PUT、DELETE、PATCH、HEAD、OPTIONS、TRACE）

## 请求头：包含请求的附加信息，由键值对组成

## 空行

## 请求体：承载多个请求参数的数据；

name=tom&password=1234&realName=tomson

* 服务器处理请求并返回HTTP报文：

## MVC后台处理阶段：MVC 是一个设计模式，将应用程序分成三个核心部件：

## 模型（model）：负责数据交互。

## 视图（view）：提供给用户操作界面。

## 控制器（controller）：负责根据用户从"视图层"输入的指令，选取"模型层"中的数据，然后对其进行相应的操作，产生最终结果。

* HTTP响应报文，组成部分包括：

## 响应行：包括协议版本、状态码、状态码描述；

## 响应头部：包括响应报文的附加信息，由键值对组成；

## 空行

## 响应主体：包含回车符、换行符和响应返回数据；

* 浏览器解析渲染页面，分为五个步骤：
* 根据 HTML 解析出 DOM 树：

## DOM 树解析的过程是一个深度优先遍历。即先构建当前节点的所有子节点，再构建下一个兄弟节点。

## 构建 DOM 树的过程中，若遇到 script 标签，则 DOM 树的构建会暂停，直至脚本执行完毕。

* 根据 CSS 解析生成 CSS 规则树：

## 解析 CSS 规则树时 js 执行将暂停，直至 CSS 规则树就绪。

## 浏览器在 CSS 规则树生成之前不会进行渲染。

* 结合 DOM 树和 CSS 规则树，生成渲染树；
* 根据渲染树计算每一个节点的信息（布局）
* 根据计算好的信息绘制页面
* 断开连接（TCP四次挥手）：
* 发起方向被动方发送报文，Fin、Ack、Seq，表示已经没有数据传输了。并进入 FIN\_WAIT\_1 状态。

(第一次挥手：由浏览器发起的，发送给服务器，我请求报文发送完了，你准备关闭吧)

* 被动方发送报文，Ack、Seq，表示同意关闭请求。此时主机发起方进入 FIN\_WAIT\_2 状态。

(第二次挥手：由服务器发起的，告诉浏览器，我请求报文接受完了，我准备关闭了，你也准备吧)

* 被动方向发起方发送报文段，Fin、Ack、Seq，请求关闭连接。并进入 LAST\_ACK 状态。

(第三次挥手：由服务器发起，告诉浏览器，我响应报文发送完了，你准备关闭吧)

* 发起方向被动方发送报文段，Ack、Seq。然后进入等待 TIME\_WAIT 状态。被动方收到发起方的报文段以后关闭连接。发起方等待一定时间未收到回复，则正常关闭。

(第四次挥手：由浏览器发起，告诉服务器，我响应报文接受完了，我准备关闭了，你也准备吧)

