1.ip解析 2发起tcp的3次握手 3 建立连接之后发送http请求 4 服务端响应http请求 5.浏览器解析html代码 并请求html代码的资源 6 浏览器对页面进行渲染呈现给用户 <br><br>1xx: 信息性状态码<br><br>    100, 101<br><br>2xx: 成功状态码<br><br>    200：OK<br><br>3xx: 重定向状态码<br><br>    301: 永久重定向, Location响应首部的值仍为当前URL，因此为隐藏重定向;<br><br>    302: 临时重定向，显式重定向, Location响应首部的值为新的URL<br><br>    304：Not Modified  未修改，比如本地缓存的资源文件和服务器上比较时，发现并没有修改，服务器返回一个304状态码，<br><br>                        告诉浏览器，你不用请求该资源，直接使用本地的资源即可。<br><br>4xx: 客户端错误状态码<br><br>    404: Not Found  请求的URL资源并不存在<br><br>5xx: 服务器端错误状态码<br><br>    500: Internal Server Error  服务器内部错误<br><br>    502: Bad Gateway  前面代理服务器联系不到后端的服务器时出现<br><br>    504：Gateway Timeout  这个是代理能联系到后端的服务器，但是后端的服务器在规定的时间内没有给代理服务器响应

1.nginx+php

1)nginx读取配置文件，nginx读取http请求的头部信息，根据host来匹配自己所有的虚拟主机的配置文件的server\_name,看有没有匹配的，有就匹配该虚拟机

2）把PHP的文件交给fastcgi进程去处理

于是nginx把/index.php这个url交给了后端的fastcgi进程去处理，等待fastcgi处理完毕后，返回给nginx一个html文档。nginx把这个文档返回给浏览器。就是浏览器就拿到了html文件，同时写一条访问日志到日志文件里去。

注1.nginx是怎么找index.php文件的？

当nginx发现需要/web/echo/index.php问件时，就会向内核发起IO系统调用，告诉内核，我需要这个文件，内核从/开始找到web目录，在从web目录找到echo目录，再从echo目录找到index.php文件，就是就把index.php从硬盘读取到内核自身的内存空间，然后再把这个文件复制到nginx进程所在的内存空间。于是nginx就得到了自己想要的文件。