

第一部分： 基础篇

一、HTML、HTTP、web综合问题

1 前端需要注意哪些SEO

- 合理的 `title`、`description`、`keywords`：搜索对着三项的权重逐个减小，`title` 值强调重点即可，重要关键词出现不要超过2次，而且要靠前，不同页面 `title` 要有所不同；`description` 把页面内容高度概括，长度合适，不可过分堆砌关键词，不同页面 `description` 有所不同；`keywords` 列举出重要关键词即可
- 语义化的 `HTML` 代码，符合W3C规范：语义化代码让搜索引擎容易理解网页
- 重要内容 `HTML` 代码放在最前：搜索引擎抓取 `HTML` 顺序是从上到下，有的搜索引擎对抓取长度有限制，保证重要内容一定会被抓取
- 重要内容不要用 `js` 输出：爬虫不会执行js获取内容
- 少用 `iframe`：搜索引擎不会抓取 `iframe` 中的内容
- 非装饰性图片必须加 `alt`
- 提高网站速度：网站速度是搜索引擎排序的一个重要指标

2 `` 的 `title` 和 `alt` 有什么区别

- 通常当鼠标滑动到元素上的时候显示
- `alt` 是 `` 的特有属性，是图片内容的等价描述，用于图片无法加载时显示、读屏器阅读图片。可提图片高可访问性，除了纯装饰图片外都必须设置有意义的值，搜索引擎会重点分析。

3 HTTP的几种请求方法用途

- `GET` 方法
 - 发送一个请求来取得服务器上的某一资源
- `POST` 方法
 - 向 `URL` 指定的资源提交数据或附加新的数据
- `PUT` 方法

- 跟 **POST** 方法很像，也是想服务器提交数据。但是，它们之间有不同。**PUT** 指定了资源在服务器上的位置，而 **POST** 没有
- **HEAD** 方法
 - 只请求页面的首部
- **DELETE** 方法
 - 删除服务器上的某资源
- **OPTIONS** 方法
 - 它用于获取当前 **URL** 所支持的方法。如果请求成功，会有一个 **Allow** 的头包含类似“**GET,POST**”这样的信息
- **TRACE** 方法
 - **TRACE** 方法被用于激发一个远程的，应用层的请求消息回路
- **CONNECT** 方法
 - 把请求连接转换到透明的 **TCP/IP** 通道

4 从浏览器地址栏输入url到显示页面的步骤

基础版本

- 浏览器根据请求的 **URL** 交给 **DNS** 域名解析，找到真实 **IP**，向服务器发起请求；
- 服务器交给后台处理完成后返回数据，浏览器接收文件（**HTML**、**JS**、**CSS**、图象等）；
- 浏览器对加载到的资源（**HTML**、**JS**、**CSS** 等）进行语法解析，建立相应的内部数据结构（如 **HTML** 的 **DOM**）；
- 载入解析到的资源文件，渲染页面，完成。

详细版

1. 在浏览器地址栏输入URL
2. 浏览器查看缓存，如果请求资源在缓存中并且新鲜，跳转到转码步骤
 1. 如果资源未缓存，发起新请求
 2. 如果已缓存，检验是否足够新鲜，足够新鲜直接提供给客户端，否则与服务器进行验证。
3. 检验新鲜通常有两个HTTP头进行控制 **Expires** 和 **Cache-Control**：
 - HTTP1.0提供Expires，值为一个绝对时间表示缓存新鲜日期
 - HTTP1.1增加了Cache-Control: max-age=,值为以秒为单位的最大新鲜时间

3. 浏览器解析URL获取协议， 主机， 端口， path
4. 浏览器组装一个HTTP（GET）请求报文
5. 浏览器获取主机ip地址， 过程如下：
 1. 浏览器缓存
 2. 本机缓存
 3. hosts文件
 4. 路由器缓存
 5. ISP DNS缓存
 6. DNS递归查询（可能存在负载均衡导致每次IP不一样）
6. 打开一个socket与目标IP地址， 端口建立TCP链接， 三次握手如下：
 1. 客户端发送一个TCP的SYN=1， Seq=X的包到服务器端口
 2. 服务器发回SYN=1， ACK=X+1， Seq=Y的响应包
 3. 客户端发送ACK=Y+1， Seq=Z
7. TCP链接建立后发送HTTP请求
8. 服务器接受请求并解析， 将请求转发到服务程序， 如虚拟主机使用工TTP 工ost头部判断请求的服务程序
9. 服务器检查HTTP请求头是否包含缓存验证信息如果验证缓存新鲜， 返回304等对应状态码
10. 处理程序读取完整请求并准备工TTP响应， 可能需要查询数据库等操作
11. 服务器将响应报文通过TCP连接发送回浏览器
12. 浏览器接收工TTP响应， 然后根据情况选择关闭TCP连接或者保留重用， 关闭TCP连接的四次握手如下：
 1. 主动方发送Fin=1， Ack=Z， Seq= X报文
 2. 被动方发送ACK=X+1， Seq=Z报文
 3. 被动方发送Fin=1， ACK=X， Seq=Y报文
 4. 主动方发送ACK=Y， Seq=X报文
13. 浏览器检查响应状态码：是否为1XX， 3XX， 4XX， 5XX， 这些情况处理与2XX不同
14. 如果资源可缓存， 进行缓存
15. 对响应进行解码（例如gzip压缩）
16. 根据资源类型决定如何处理（假设资源为工TML文档）
17. 解析HTML文档， 构件DOM树， 下载资源， 构造CSSOM树， 执行js脚本， 这些操作没有严格的先后顺序， 以下分别解释
18. 构建DOM树：
 1. Tokenizing：根据工TML规范将字符流解析为标记
 2. Lexing：词法分析将标记转换为对象并定义属性和规则
 3. DOM construction：根据工TML标记关系将对象组成DOM树
19. 解析过程中遇到图片、样式表、js文件， 启动下载
20. 构建CSSOM树：
 1. Tokenizing：字符流转换为标记流