

同濟大學

TONGJI UNIVERSITY

《WEB 技术》

实验报告（大作业）

实验名称

万维网运行原理分析

小组成员

学院（系）

电子与信息工程学院

专 业

计算机科学与技术专业

任课教师

郭玉臣

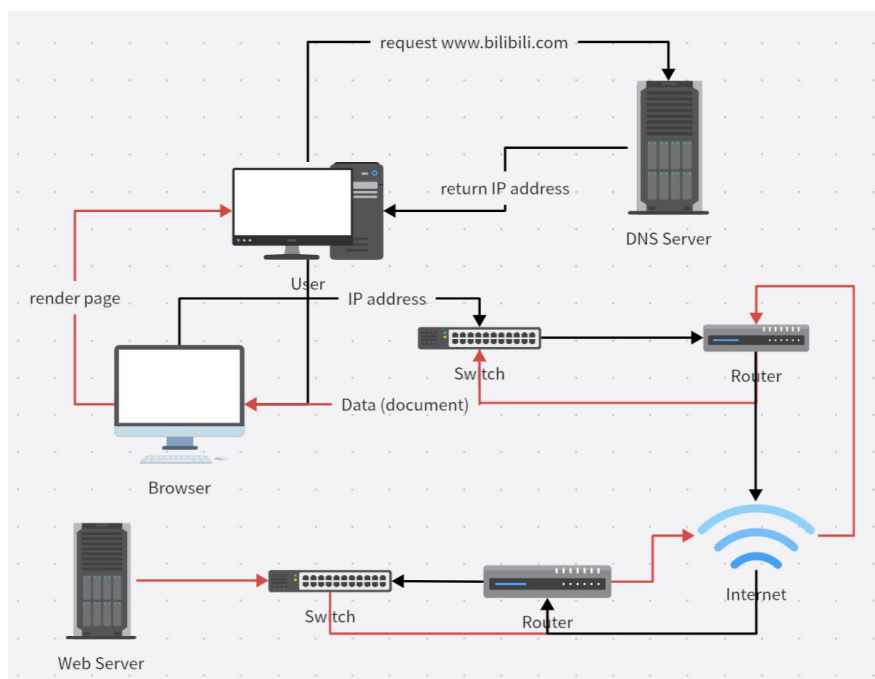
日 期

2023 年 3 月 14 日

Content

- 1.网络拓扑与数据流向
- 2.具体流程
- 3.HTTP 协议
- 4.HTML 网页结构
- 5.心得体会

1. 网络拓扑与数据流向



输入一个网址进行访问，这中间其实是客户端浏览器与服务器端的通信过程,具体如下：

1. 浏览器与网络上的域名对应的 Web 服务器建立 TCP 连接浏览器发出要求访问某个页面的 HTTP 请求
2. Web 服务器在接收到 HTTP 请求后，解析 HTTP 请求，然后发回包含目标页面的文件数据的 HTTP 响应
3. 浏览器接收到 HTTP 响应后，解析 HTTP 响应，并在其窗口中展示网页文件内容
4. 浏览器与 Web 服务器之间的 TCP 连接关闭。

2. 具体流程

此处使用 Wireshark 和 Fiddler 作为抓包工具进行试验，观察 HTTP 报文、TCP/IP 协议工作过程等

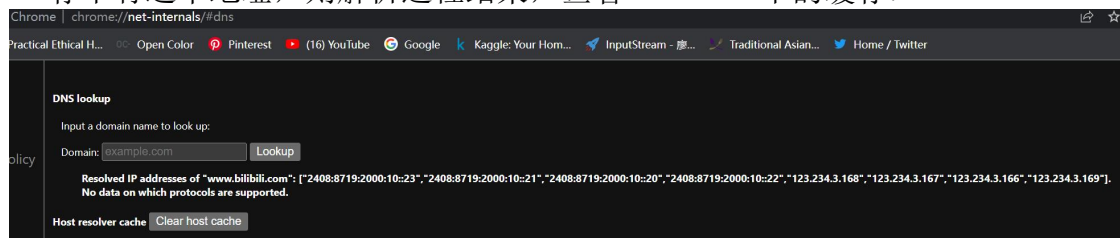


此处以请求访问 bilibili 网站为例，用户首先在浏览器输入网址 (www.bilibili.com)，使用 Chrome 浏览器进行试验

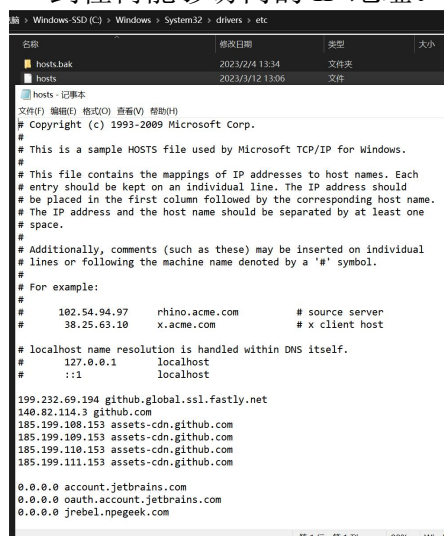


回车键后进行 DNS 域名解析，

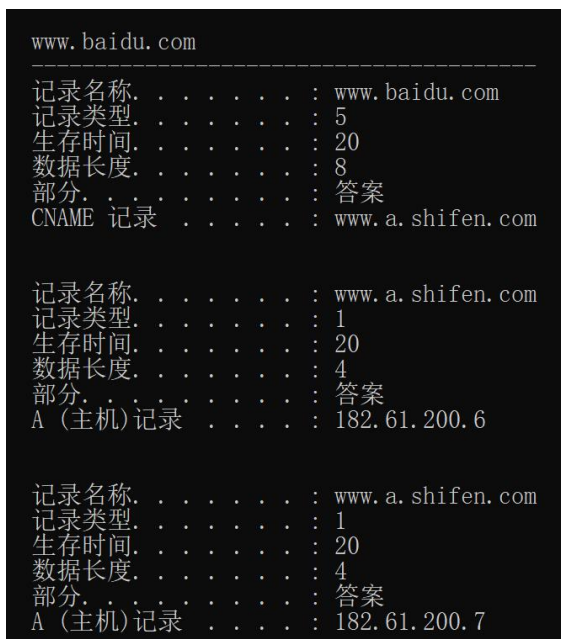
1. 本地电脑会检查浏览器缓存中有没有这个域名对应的解析过的 IP 地址，如果缓存中有这个地址，则解析过程结束，查看 Chrome 中的缓存：



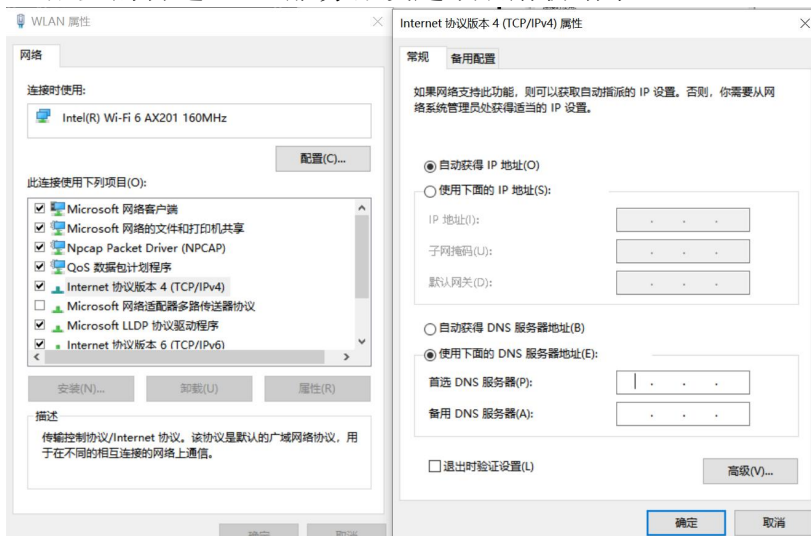
2. 浏览器没有缓存数据时，浏览器会查找操作系统缓存中是否有这个域名对应的 DNS 解析结果，如果有则解析成功。在 windows 操作系统中可以配置 C:\Windows\System32\drivers\etc\hosts 来进行设置，用户可以将任何域名解析到任何能够访问的 IP 地址。



查看操作系统自身的 DNS 缓存,终端输入: ipconfig /displaydns
此处展示 www.baidu.com 的操作系统 DNS 缓存



- 3.前两个过程无法解析时，操作系统会把此域名发送给本地 DNS 服务器，向本地配置的首选 DNS 服务器发起域名解析请求



在此处设置更改 DNS 服务器地址

运营商的 DNS 服务器首先查找自身的缓存，找到对应的条目，且没有过期，则解析成功。

- 如果本地 DNS 服务器仍然没有命中，就直接到根 DNS 服务器请求解析。根 DNS 服务器若发现所请求的域名为顶级域如 com，cn 域的域名，则返回给本地 DNS 服务器顶级域的 IP 地址，则本地又向此 IP 地址发出请求，
- 接受请求的顶级 DNS 服务器查找并返回此域名对应的 Name Server 域名服务器的地址，查询存储的域名和 IP 的映射关系表，再把查询出来的域名和 IP 地址等等信息，连同一个 TTL 值返回给本地 DNS 服务器。

TTL: time to live, ip 协议的值，它告诉网络，数据包在网络中的时间是否太长而应

被丢弃

- 本地 DNS 服务器会缓存这个域名和 IP 的对应关系，缓存时间由 TTL 值控制。
- 把解析的结果返回给本地电脑，本地电脑根据 TTL 值缓存在本地系统缓存中，域名解析过程结束

• 用 nslookup 查询 DNS 解析过程：nslookup -debug www.bilibili.com

```
C:\Users\ats14\nslookup -debug www.bilibili.com

Got answer:
HEADER:
  opcode = QUERY, id = 1, rcode = NOERROR
  header flags: response, want recursion, recursion avail.
  questions = 1, answers = 1, authority records = 0, additional = 0

QUESTIONS:
  208.190.120.202.in-addr.arpa, type = PTR, class = IN
ANSWERS:
  -> 208.190.120.202.in-addr.arpa
      name = dnscache1.tongji.edu.cn
      ttl = 14334 (3 hours 58 mins 54 secs)

服务器: dnscache1.tongji.edu.cn
Address: 202.120.190.208

Got answer:
HEADER:
  opcode = QUERY, id = 2, rcode = NXDOMAIN
  header flags: response, want recursion, recursion avail.
  questions = 1, answers = 0, authority records = 1, additional = 0

QUESTIONS:
  www.bilibili.com.tongji.edu.cn, type = A, class = IN
AUTHORITY RECORDS:
  -> tongji.edu.cn
      ttl = 50 (50 secs)
      primary name server = dns.tongji.edu.cn
      responsible mail addr = root.tongji.edu.cn
      serial = 2019093090
      refresh = 7200 (2 hours)
      retry = 600 (10 mins)
      expire = 1209600 (14 days)
      default TTL = 300 (5 mins)

Got answer:
HEADER:
  opcode = QUERY, id = 3, rcode = NXDOMAIN
  header flags: response, want recursion, recursion avail.
  questions = 1, answers = 0, authority records = 1, additional = 0

QUESTIONS:
  www.bilibili.com.tongji.edu.cn, type = AAAA, class = IN
AUTHORITY RECORDS:
  -> tongji.edu.cn
      ttl = 300 (5 mins)
      primary name server = dns.tongji.edu.cn
      responsible mail addr = root.tongji.edu.cn
      serial = 2019093090
      refresh = 7200 (2 hours)
      retry = 600 (10 mins)
      expire = 1209600 (14 days)
      default TTL = 300 (5 mins)

Got answer:
HEADER:
  opcode = QUERY, id = 4, rcode = NXDOMAIN
  header flags: response, want recursion, recursion avail.
  questions = 1, answers = 0, authority records = 1, additional = 0

QUESTIONS:
  www.bilibili.com.edu.cn, type = A, class = IN
AUTHORITY RECORDS:
  -> edu.cn
      ttl = 10291 (2 hours 51 mins 31 secs)
      primary name server = dns.edu.cn
      responsible mail addr = hostmaster.net.edu.cn
      serial = 2023031319
      refresh = 7200 (2 hours)
      retry = 1800 (30 mins)
      expire = 604800 (7 days)
      default TTL = 21600 (6 hours)

Got answer:
HEADER:
  opcode = QUERY, id = 5, rcode = NXDOMAIN
  header flags: response, want recursion, recursion avail.
  questions = 1, answers = 0, authority records = 1, additional = 0

QUESTIONS:
  www.bilibili.com.edu.cn, type = AAAA, class = IN
AUTHORITY RECORDS:
  -> edu.cn
      ttl = 10291 (2 hours 51 mins 31 secs)
      primary name server = dns.edu.cn
      responsible mail addr = hostmaster.net.edu.cn
      serial = 2023031319
      refresh = 7200 (2 hours)
      retry = 1800 (30 mins)

Got answer:
HEADER:
  opcode = QUERY, id = 5, rcode = NXDOMAIN
  header flags: response, want recursion, recursion avail.
  questions = 1, answers = 0, authority records = 1, additional = 0

QUESTIONS:
  www.bilibili.com.edu.cn, type = AAAA, class = IN
AUTHORITY RECORDS:
  -> edu.cn
      ttl = 10291 (2 hours 51 mins 31 secs)
      primary name server = dns.edu.cn
      responsible mail addr = hostmaster.net.edu.cn
      serial = 2023031319
      refresh = 7200 (2 hours)
      retry = 1800 (30 mins)

Got answer:
HEADER:
  opcode = QUERY, id = 6, rcode = NOERROR
  header flags: response, want recursion, recursion avail.
  questions = 1, answers = 5, authority records = 0, additional = 0

QUESTIONS:
  www.bilibili.com, type = A, class = IN
ANSWERS:
  -> www.bilibili.com
      canonical name = a.w.bilicdn1.com
      ttl = 50 (50 secs)
  -> a.w.bilicdn1.com
      internet address = 123.234.3.168
      ttl = 50 (50 secs)
  -> a.w.bilicdn1.com
      internet address = 123.234.3.166
      ttl = 50 (50 secs)
  -> a.w.bilicdn1.com
      internet address = 123.234.3.167
      ttl = 50 (50 secs)
  -> a.w.bilicdn1.com
      internet address = 123.234.3.169
      ttl = 50 (50 secs)

非权威应答:

Got answer:
HEADER:
  opcode = QUERY, id = 7, rcode = NOERROR
  header flags: response, want recursion, recursion avail.
  questions = 1, answers = 5, authority records = 0, additional = 0

QUESTIONS:
  www.bilibili.com, type = AAAA, class = IN
ANSWERS:
  -> www.bilibili.com
      canonical name = a.w.bilicdn1.com
      ttl = 33 (33 secs)
  -> a.w.bilicdn1.com
      AAAA IPv6 address = 2408:8719:2000:10::23
      ttl = 47 (47 secs)
  -> a.w.bilicdn1.com
      AAAA IPv6 address = 2408:8719:2000:10::20
      ttl = 47 (47 secs)
  -> a.w.bilicdn1.com
      AAAA IPv6 address = 2408:8719:2000:10::22
      ttl = 47 (47 secs)
  -> a.w.bilicdn1.com
      AAAA IPv6 address = 2408:8719:2000:10::21
      ttl = 47 (47 secs)

名称: a.w.bilicdn1.com
Addresses: 2408:8719:2000:10::23
           2408:8719:2000:10::20
           2408:8719:2000:10::22
           2408:8719:2000:10::21
           123.234.3.168
           123.234.3.166
           123.234.3.167
           123.234.3.169
Aliases: www.bilibili.com
```

可以看到逐级向上寻找，向上寻找 authority records 最终得到 5 个 answers 并返回给本地，缓存记录后产生 non-authority answer

找到对应的 IP 地址后开始建立连接，由 TCP/IP 协议进行三次握手进行连接

1	0.000000	100.80.168.174	121.194.11.73	TCP	66	64455 → 443 [SYN] Seq=0 Win=64240 Len=...
2	0.027597	121.194.11.73	100.80.168.174	TCP	62	443 → 64455 [SYN, ACK] Seq=0 Ack=1 Win=...
3	0.027701	100.80.168.174	121.194.11.73	TCP	54	64455 → 443 [ACK] Seq=1 Ack=1 Win=1313...

- 第一次：客户端给服务器发送一个 SYN 段(在 TCP 标头中 SYN 位字段为 1 的 TCP/IP 数据包)，该段中也包含客户端的初始序列号(Sequence number = J)。
- 第二次：服务器返回客户端 SYN +ACK 段(在 TCP 标头中 SYN 和 ACK 位字段都为 1 的 TCP/IP 数据包)，该段中包含服务器的初始序列号(Sequence number = K)；同时使 Acknowledgment number = J + 1 来表示确认已收到客户端的 SYN 段(Sequence number = J)。
- 第三次：客户端给服务器响应一个 ACK 段(在 TCP 标头中 ACK 位字段为 1 的 TCP/IP 数据包)，该段中使 Acknowledgment number = K + 1 来表示确认已收到服务器的 SYN 段(Sequence number = K)。

3. HTTP 协议

打开开发者工具，查看浏览器端发送 http 请求报文

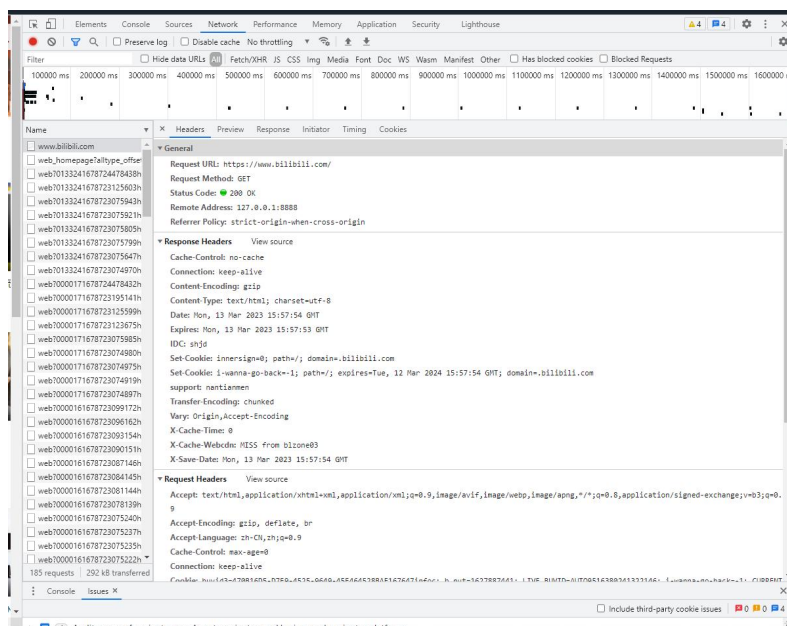
- 协议内容

HTTP 协议 (Hyper Text Transfer Protocol, 超文本传输协议), 是用于从万维网 (WWW:WorldWide Web) 服务器传输超文本到本地浏览器的传送协议。HTTP 基于 TCP/IP 通信协议来传递数据。HTTP 基于客户端/服务端 (C/S) 架构模型, 通过一个可靠的链接来交换信息, 是一个无状态的请求/响应协议

- 特点

1. HTTP 是无连接: 无连接的含义是限制每次连接只处理一个请求。服务器处理完客户的 2 请求, 并收到客户的应答后, 即断开连接。采用这种方式可以节省传输时间。
2. HTTP 是媒体独立的: 只要客户端和服务端知道如何处理的数据内容, 任何类型的数据都可以通过 HTTP 发送。客户端以及服务器指定使用适合的 MIME - type 内容类型。
3. HTTP 是无状态: 无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息, 则它必须重传, 这样可能导致每次连接传送的数据量增大。另一方面, 在服务器不需要先前信息时它的应答就较快。

3.1 请求报文



- 报文结构（以 GET 其他网页为例）

GET http://jsuacm.cn/ HTTP/1.1

Host: jsuacm.cn

Connection: keep-alive

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.25 Safari/537.36 Core/1.70.3877.400 QQBrowser/10.8.4506.400

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8

Accept-Encoding: gzip, deflate

Accept-Language: zh-CN,zh;q=0.9

请求方法	空格	URL	空格	协议版本	回车符	换行符	请求行
头部字段名	:	值		回车符	换行符		
...							
头部字段名	:	值		回车符	换行符		
回车符	换行符						
							请求数据

- 请求行

- 请求行由请求方法字段、URL 字段和 HTTP 协议版本字段 3 个字段组成，它们用空格分隔

GET http://jsuacm.cn/template/jsu/css/modifypage-1.css HTTP/1.1

- 请求方法

- HTTP 协议的请求方法有 GET、POST、HEAD、PUT、DELETE、OPTIONS、TRACE、CONNECT。GET 和 POST 是最常用的请求方法

1. **GET**: 最常见的一种请求方式，当客户端要从服务器中读取文档时，当点击网页上的链接或者通过在浏览器的地址栏输入网址来浏览网页的，使用的都是 GET 方式。GET 方法要求服务器将 URL 定位的资源放在响应报文的数据部分，回送给客户端。使用 GET 方法时，请求参数和对应的值附加在 URL 后面，利用一个问号（“?”）代表 URL 的结尾与请求参数的开始，传递参数长度受限制
2. **POST**: 和 GET 相似功能，使用 POST 方法可以允许客户端给服务器提供信息较多。
3. **HEAD**: 本质和 get 一样，只不过服务端接受到 HEAD 请求后只返回响应头，而不会发送响应内容。当我们只需要查看某个页面的状态的时候，使用 HEAD 是非常高效的，因为在传输的过程中省去了页面内容。
4. **DELETE**: 请求服务器删除某资源。和 put 都具有破坏性，可能被防火墙拦截。
5. **CONNECT**: HTTP/1.1 协议中预留给能够将连接改为管道方式的代理服务器。就是把服务器作为跳板，去访问其他网页然后把数据返回。

- URL

- 统一资源定位符，是一种资源位置的抽象唯一识别方法
- 组成：
`<协议>://<主机>:<端口>/<路径>`

- HTTP 协议版本

- 协议版本的格式为：HTTP/主版本号.次版本号，常用的有 HTTP/1.0 和 HTTP/1.1，现在最常用的是 1.1

- 请求头部

- 请求头部为请求报文添加了一些附加信息，由“名/值”对组成，每行一对，名和值之间使用冒号分隔。请求头部通知服务器有关于客户端请求的信息

- 请求头

1. **Host** : 主机 ip 地址或域名
2. **User-Agent** : 客户端相关信息，如操作系统、浏览器等信息
3. **Accept** : 指定客户端接收信息类型
4. **Accept-Charset** : 客户端接受的字符集
5. **Accept-Encoding** : 可接受的内容编码
6. **Accept-Language** : 接受的语言
7. **Authorization** : 客户端提供给服务端，进行权限认证的信息

- 9. **Cookie** : 携带的 cookie 信息
- 10. **Referer** : 当前文档的 URL

• 空行

- 请求头部的最后会有一个空行，表示请求头部结束，接下来为请求数据。

• 请求数据

- 请求数据不在 GET 方法中使用，而在 POST 方法中使用。POST 方法适用于需要客户填写表单的场合。与请求数据相关的最长使用的请求头部是 Content-Type 和 Content-Length

3.2 响应报文

```
HTTP/1.1 200 OK
Date: Mon, 13 Mar 2023 15:57:54 GMT
Content-Type: text/html; charset=utf-8
Connection: keep-alive
support: nantianmen
Set-Cookie: innersign=0; path=/; domain=.bilibili.com
Set-Cookie: i-wanna-go-back=-1; path=/; expires=Tue, 12 Mar 2024 15:57:54 GMT; domain=.bilibili.com
Vary: Origin,Accept-Encoding
IDC: shjd
Expires: Mon, 13 Mar 2023 15:57:53 GMT
Cache-Control: no-cache
X-Cache-Webcdn: MISS from blzone03
X-Cache-Time: 0
X-Save-Date: Mon, 13 Mar 2023 15:57:54 GMT
Content-Length: 93284
```

```
<!DOCTYPE html>
<html lang="zh-CN" class="gray">
  <head>
    <meta charset="UTF-8" />
    <title>哔哩哔哩 (゜-゜)つロ 干杯~-bilibili</title>
```

...

协议版本	空格	状态码	空格	协议版本	回车符	换行符	状态行
头部字段名	:	值	回车符	换行符			
							...
头部字段名	:	值	回车符	换行符			响应头部
回车符	换行符						
							响应正文

HTTP 响应报文由状态行（status line）、响应头部（headers）、空行（blank line）和响应数据（也叫响应体或响应正文）（response body）4 个部分组成

- 状态行
- 状态行由 3 部分组成，分别为：协议版本、状态码、状态码描述。其中协议版本与请求报文一致，状态码描述是对状态码的简单描述
- 常见状态码

状态码	英文名称	中文描述
100	Continue	继续。客户端应继续其请求
101	Switching Protocols	切换协议。服务器根据客户端的请求切换协议。只能切换到更高级的协议，例如，切换到 HTTP 的新版本协议
200	OK	请求成功。一般用于 GET 与 POST 请求
204	No Content	无内容。服务器成功处理，但未返回内容。在未更新网页的情况下，可确保浏览器继续显示当前文档
301	Moved Permanently	永久移动。请求的资源已被永久的移动到新 URI，返回信息会包括新的 URI，浏览器会自动定向到新 URI。今后任何新的请求都应使用新的 URI 代替
302	Found	临时移动。与 301 类似。但资源只是临时被移动。客户端应继续使用原有 URI
400	Bad Request	客户端请求的语法错误，服务器无法理解
404	Not Found	服务器无法根据客户端的请求找到资源（网页）。通过此代码，网站设计人员可设置“您所请求的资源无法找到”的个性页面
408	Request Time-out	服务器等待客户端发送的请求时间过长，超时
500	Internal Server Error	服务器内部错误，无法完成请求
502	Bad Gateway	作为网关或者代理工作的服务器尝试执行请求时，从远程服务器接收到了一个无效的响应
504	Gateway Time-out	充当网关或代理的服务器，未及时从远端服务器获取请求

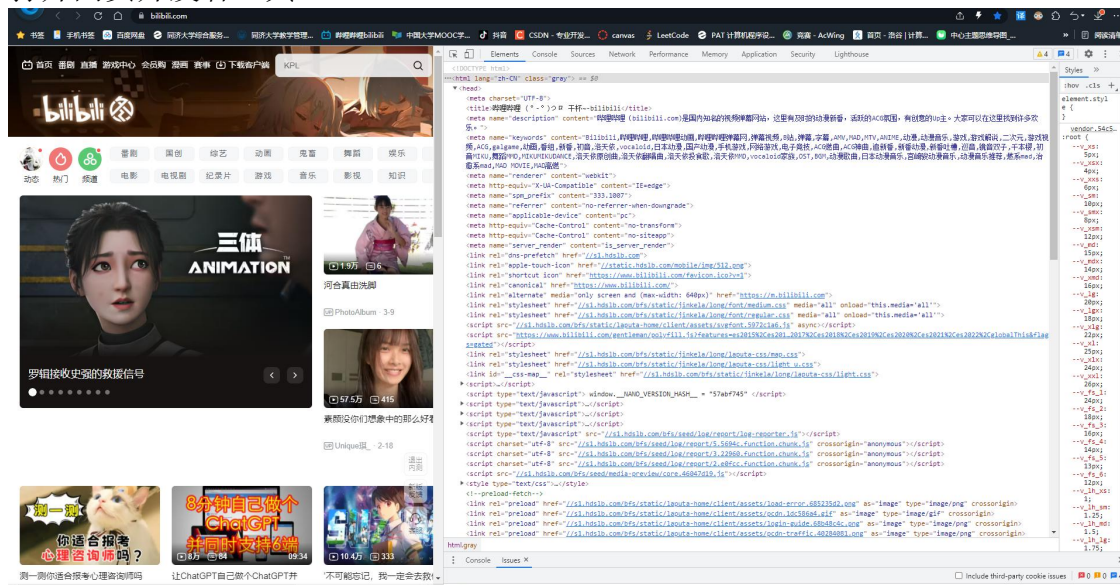
- 响应头部
- 响应头
 1. **Server**: HTTP 服务器的软件信息
 2. **Date**: 响应报文的时间
 3. **Expires**: 指定缓存过期时间
 4. **Set-Cookie**: 设置 Cookie
 5. **Last-Modified**: 资源最后修改时间
 6. **Content-Type**: 响应的类型和字符集，如: Content-Type: text/html; charset=utf-8
 7. **Content-Length**: 内容长度
 8. **Connection**: 如 Keep-Alive, 表示保持 tcp 连接不关闭，不会永久保持连接，服务器可设置
 9. **Location**: 指明重定向的位置，新的 URL 地址，如 302 的情况
- 空行

响应头部的最后会有一个空行，表示响应头部结束，接下来为响应数据。与请求报文一致。
- 响应体
 - 用于存放需要返回给客户端的数据信息。此处数据为 b 站主页的 HTML 数据。

- TCP/IP 协议四次挥手断开连接，将响应报文沿着原路径传递给浏览器，浏览器接受 HTML 数据并进行解析渲染

4. HTML 网页结构

打开网页开发者工具



由抓包工具 fiddler 抓取 html 文本

1. 文档标签

```
<html lang="zh-CN" class="gray">
</html>
```

文档标签表明此处是一个 html 文档，其中可以用 lang 标记网页默认语言，是 HTMLW 文档中最外层的元素，是所有其他 HTML 元素（除了 <!DOCTYPE> 标签）的容器。

2. 头部 head

```
<head>
```

...

```
</head>
```

用于初始化网页必要资源以及一些不是网页文本内的网页元素，如网页的标题等。head 标签内含有 meta 元标签、title 网页标题标签，link 链接标签、script 脚本标签等，网页必要成分的加载



3. meta 标签

```
<meta name="renderer" content="webkit" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="spm_prefix" content="333.1007" />
  <meta name="referrer" content="no-referrer-when-downgrade" />
  <meta name="applicable-device" content="pc">
  <meta http-equiv="Cache-Control" content="no-transform" />
  <meta http-equiv="Cache-Control" content="no-siteapp" />
  <meta name="server_render" content="is_server_render" />
```

meta 标签用于提供 HTML 文档的元数据。元数据不会显示在客户端，但是会被浏览器解析。其通常用于指定网页的描述，关键词，文件的最后修改时间，作者及其他元数据。元数据可以被使用浏览器（如何显示内容或重新加载页面），搜索引擎（关键词），或其他 Web 服务调用

4. link 标签

```
<link rel="dns-prefetch" href="//sl.hdslb.com" />
<link rel="apple-touch-icon" href="//static.hdslb.com/mobile/img/512.png" />
<link rel="shortcut icon" href="https://www.bilibili.com/favicon.ico?v=1" />
<link rel="canonical" href="https://www.bilibili.com/" />
<link rel="alternate" media="only screen and (max-width: 640px)" href="https://m.bilibili.com" />
<link
  rel="stylesheet"
  href="//sl.hdslb.com/bfs/static/jinkela/long/font/medium.css"
  media="print"
  onload="this.media='all'"
/>
<link
  rel="stylesheet"
  href="//sl.hdslb.com/bfs/static/jinkela/long/font/regular.css"
  media="print"
  onload="this.media='all'"
/>
<link rel="stylesheet" href="//sl.hdslb.com/bfs/static/jinkela/long/laputa-css/map.css"/>
<link rel="stylesheet" href="//sl.hdslb.com/bfs/static/jinkela/long/laputa-css/light_u.css"/>
<link id="__css-map__" rel="stylesheet" href="//sl.hdslb.com/bfs/static/jinkela/long/laputa-css/light.css"/>
```

定义了文档与外部资源的关系，可以用于链接样式表（CSS），可以同时链接多个外部文件，比如链接多个样式表，从而使得 CSS 的设计可复用性更高，同时 link 元素可以链接配置文件等，在响应式设计时可以解决安卓用户的图片加载问题

5. script 标签

```
<script type="text/javascript">
window.__NANO_VERSION_HASH__ = "57abf745"
</script>
<script type="text/javascript">
    ;(function () {
        if (document.querySelector('meta[name=server_render]')) {
            return
        }
        var ua = window.navigator.userAgent,
            agents = ['Android', 'iPhone', 'SymbianOS', 'Windows Phone', 'iPod'],
            isPC = true
        if (/sVR\s/g.test(ua)) return
        for (var i = 0, len = agents.length; i < len; i++) {
            if (ua.indexOf(agents[i]) > 0) {
                isPC = false
                break
            }
        }
        if (!isPC) {
            window.location.href = window.location.href.replace('www', 'm')
        }
    })()
</script>
<script type="text/javascript">
    window.spmReportData = {}
    window.reportConfig = {
        sample: 1,
        msgObjects: 'spmReportData',
        errorTracker: true
    }
    function getCookie(name) {
        var reg = new RegExp('(\\| )' + name + '=(\\[\\;\\]*)\\(\\;$)')
        var r = document.cookie.match(reg)
        return r ? unescape(r[2]) : null
    }
    function fsrCb() {
        if (window.performance && window.performance.timing) {
            window.performance.timing.firstscreenfinish = new Date().getTime()
        }
    }
    function lqipCb(img) {
        var lqip =
            img && img.parentNode && img.parentNode.querySelector('.lqip')
        if (lqip) {
            lqip.classList.add('is-active')
        }
    }
}
```

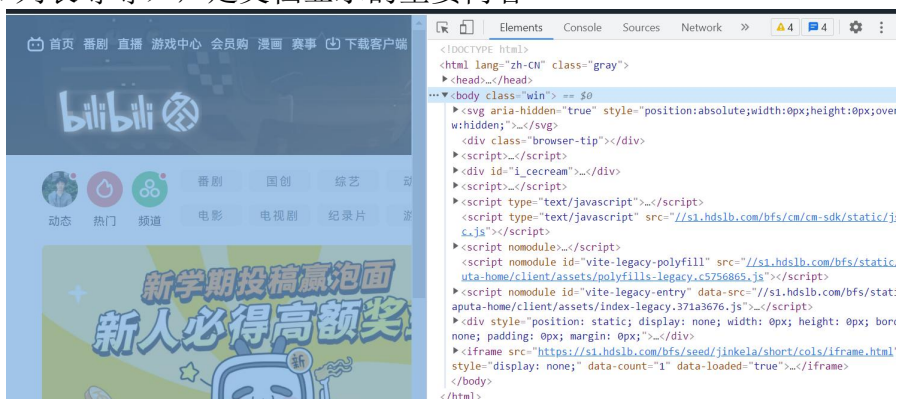
```
if (history.scrollRestoration) {
  history.scrollRestoration = 'manual'
}
...
</script>
```

用于定义客户端脚本,比如 JavaScript, 其既可包含脚本语句,也可以通过 "src" 属性指向外部脚本文件。JavaScript 通常用于图像操作、表单验证以及动态内容更改。其中在 body 标签内同样会有 script 标签,协助进行页面内容的操作

6. body 标签

```
<body class="win">
  <svg aria-hidden="true" style="position:absolute;width:0px;height:0px;overflow:hidden;"></svg>
  <div class="browser-tip"></div>
  <script></script>
  <div id="i_ccream"></div>
  <script></script>
  <script type="text/javascript" src="//s1.hdslb.com/bfs/cm/cm-sdk/static/js/pc.js"></script>
  <script nomodule></script>
  <script nomodule id="vite-legacy-polyfill" src="//s1.hdslb.com/bfs/static/laputa-home/client/assets/polyfills-legacy.c5756865.js"></script>
  <script nomodule id="vite-legacy-entry" data-src="//s1.hdslb.com/bfs/static/laputa-home/client/assets/index-legacy.371a3676.js"></script>
  <div style="position: static; display: none; width: 0px; height: 0px; border: none; padding: 0px; margin: 0px;"></div>
  <iframe src="https://s1.hdslb.com/bfs/seed/jinkela/short/cols/iframe.html" style="display: none;" data-count="1" data-loaded="true"></iframe>
</body>
```

body 标签定义文档的主体, 包含文档的所有内容(比如文本、超链接、图像、表格和列表等等), 是文档显示的主要内容



7. svg 标签

```
<body class="win">
  <svg aria-hidden="false" style="position:absolute;width:0px;height:0px;">
  </svg>
```

矢量图形式图片(此处并不是图片显示作用)在实际开发过程中, 矢量图形式图片似乎有些繁琐, 在某些 icon 处可以用一些前端 UI 库进行 js 注入, 从而使 html 文档更加语义化且更加整洁

8. div 标签



常用的分区标签，可以对网页区域继续划分，注意 div 是块级元素，在排版时辉进行自动换行，在 css 中可以重写 display 进行调整。另一种可以在水平进行排版的标签还有 span 等，主要在某些水平分区，也可以用 div 进行替代

9. iframe 标签

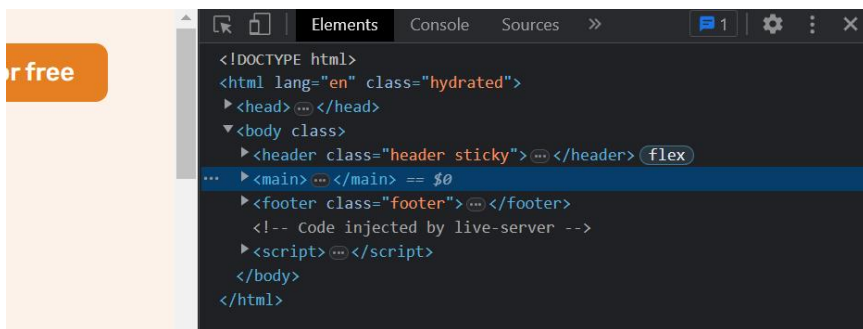


用于在网页中嵌入另一个 html 文档，也就是在网页中嵌套网页，可以看到此处 iframe 标签中含有另一个网页的 html 文档

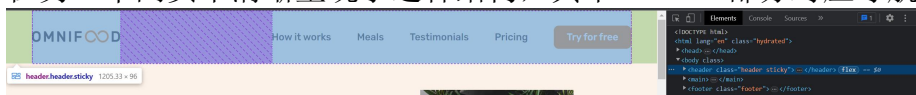
10. main 标签



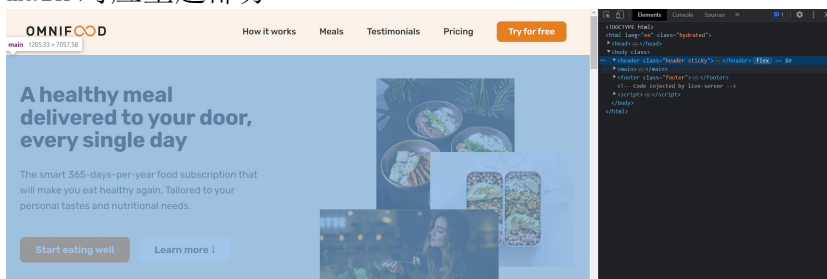
main 标签代表的是网页的主要内容，其作为语义化分块标签对于网页的划分有一定意义。对于介绍类网页，主要可以分为 header, main, footer 的几个部分



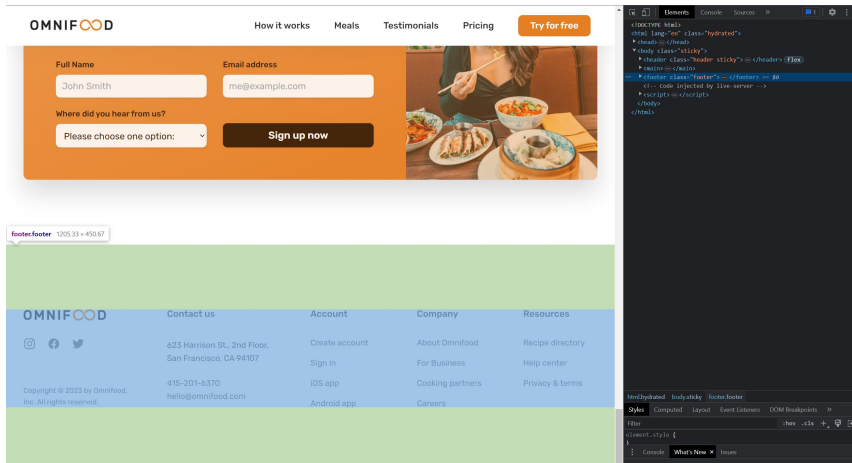
在另一个网页中清晰呈现了这种结构，其中 header 部分对应导航栏部分



main 对应主题部分



footer 对应尾部联系栏部分



11. 文本类标签

文本类标签包括 p 段落标签

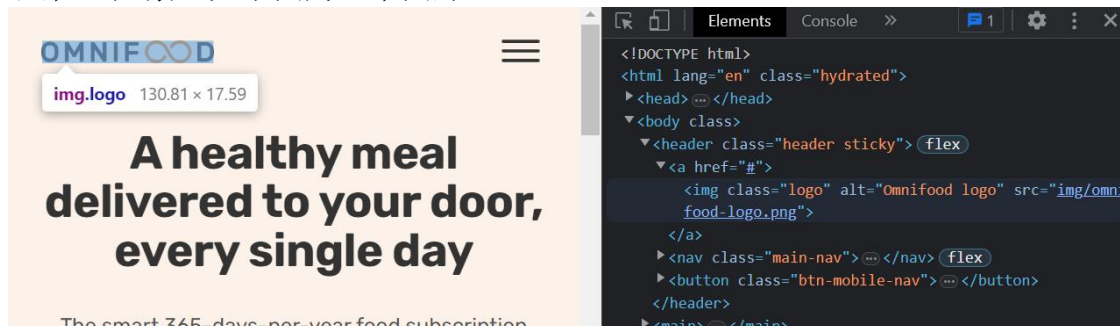
▼<p title="【不过重读】24屠皓民唐迟颀斌斌英语一VIP" class="title" data-v-1d8f90b1>

文本会处理成段落并自动换行

i em (倾斜)b strong (加粗)...等文本处理标签，可以将文本样式进行处理，但是一定要注意 html 标签的 semantic 语义化，将 html 标签尽量贴近清晰明了的含义很重要 比如使用 em，strong 等进行处理

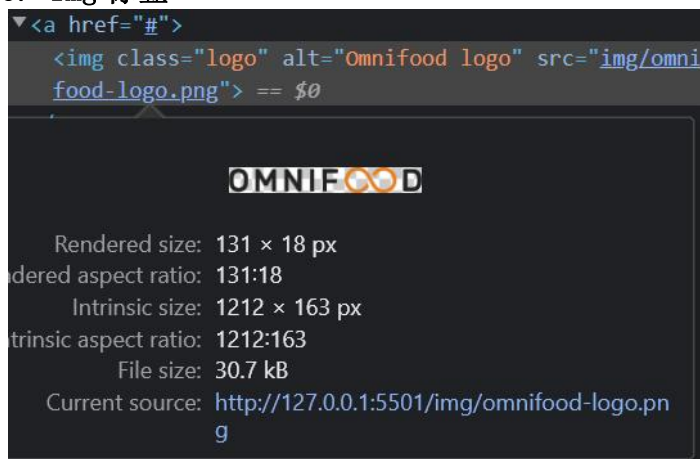
12. a 标签

a 标签用于进行 html 之间的跳转链接，也可以用作页面内的跳转链接，其内容可以是任意元素，如下图为一个图片。



注意 a 标签是块级元素，说明其在排版时会自动换行，对于想要处理水平链接情况，可以重写 css，display:inline-block 实现。作为页外跳转可以在 href 标记好 html 链接，作为页内跳转可以用 id 标记唯一标签的方式 href=#id 进行跳转（Id 作为标识不会有第二个位置）

13. img 标签



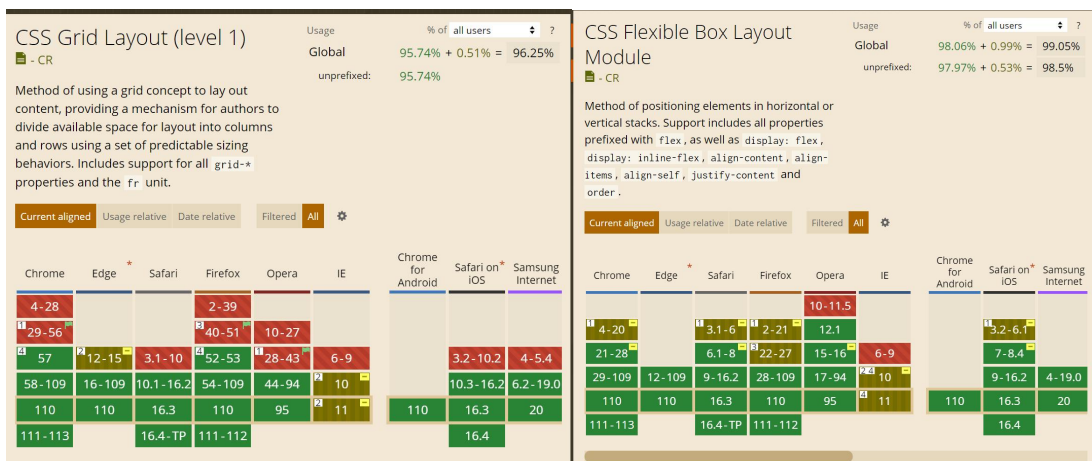
img 标签可以向 html 中嵌入图片，注意 alt 元素一定要补充，作为在图片页面没有被加载成功的时候进行的文字显示，同时可以被屏幕阅读器捕捉到，对于视力缺陷的用户更加友好

img 图片的大小应是最终在 html 文档里大小的二倍左右，过大会造成网页加载过程被拖慢，过小时会使得当前图片在较大屏幕上显示时模糊不清

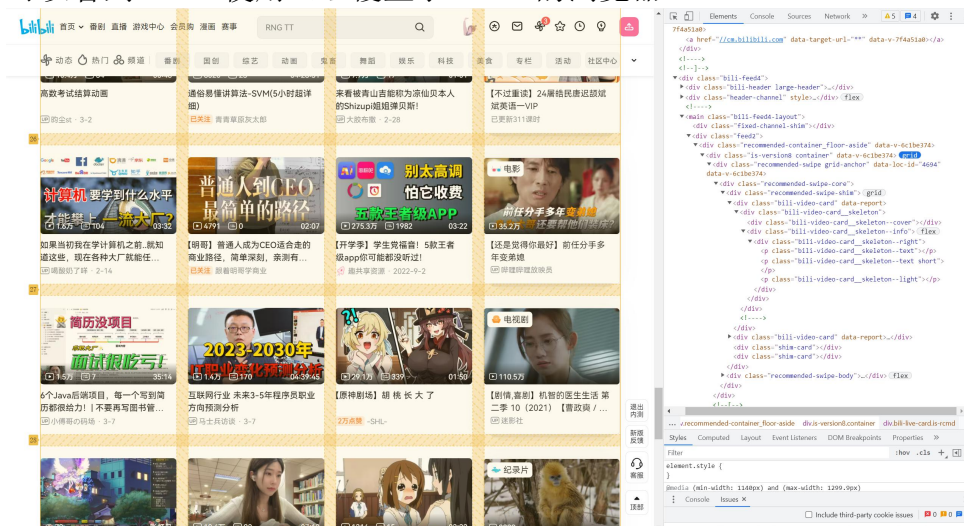
- grid 布局
- flex 布局

这两种布局方式是 CSS 的内容，但是在 html 布局方面更需要提前考虑 grid 布局和 flex 布局相对于传统的表格布局具有明显的优势，其操作更加灵活多变，产生 css 效果更加丰富

这两种布局在极少数浏览器不适用，对于主流浏览器来说正常工作



可以看到 flex 使用已经覆盖了 99.5% 的浏览器



可以看到 b 站也应用了 grid 布局, 这种布局方式让 html 在顺序书写的时候更加清晰流畅, 突出了内容与样式分离的思想

5. 心得体会

网页请求以及传输涉及到计算机网络的方方面面, 其中设计到 DNS 解析, http 协议, TCP/IP 协议, 网络传输, 文件解析, 网页渲染等。而在用户浏览器端 html, css 渲染中又涉及到结构问题、样式问题、动态响应问题等。文件在计算机网络中的传输时 Web 技术的核心基础, 而进行网页端以及前后端信息交互是 Web 的技术重点。