同济大学

TONGJI UNIVERSITY

《WEB 技术》

实验报告

实验名称	实验 1 万维网运行原理分析
小组成员	郑博远（2154312）
学院（系）	电子与信息工程学院计算机科学与技术系
专　　业	计算机科学与技术
任课教师	郭玉臣
日　　期	2023 年 3 月 8 日

一、实验介绍

本实验通过对特定网站分析，了解网站运行原理和相关技术； 通过使用抓包工具或浏览器自带工具采集 HTTP 协议包并进行分析。

二、实验目标

1. 深入了解万维网结构、原理、技术；

2. 深入了解并掌握 WEB 页面组成；

3. 深入了解并掌握 HTTP 协议。

装

三、实验原理与方法

3.1 万维网（www）运行原理

订

输入一个网址进行访问，其实是客户端浏览器与服务器端的通信过程，具体如下：

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，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 响应浏览器接受到 HTTP 响应后，解析 HTTP 响应，并在其窗口中展示网页文件内容，浏览器与 Web 服务器之间的 TCP 连接关闭。

线

3.2 服务器

接受来自浏览器的 TCP 的请求接收并解析 HTTP 请求创建并发送 HTTP 响应。

常用的 Web 服务器有 IIS，Tomcat，Weblogic，jboss 等。

3.3 浏览器

请求与 Web 服务器建立 TCP 连接创建并发送 HTTP 请求接受并解析 HTTP 响应展示 html 文档。

HTTP 客户程序（浏览器）和 HTTP 服务器分别由不同的软件开发商提供,目前最流行的浏览器有 IE，Firefox，Google Chrome，Apple Safari 等。

3.4 HTTP 协议

简介：

HTTP 协议（Hyper Text Transfer Protocol，超文本传输协议），是用于从万维网（WWW:World Wide 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 是无连接的：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：只要客户端和服务器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 是无状态的：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。

HTTP 请求报文：

1) 请求行：

①是请求方法，GET 和 POST 是最常见的 HTTP 方法，除此以外还包括 DELETE、HEAD、OPTIONS、PUT、TRACE。

②为请求对应的 URL 地址，它和报文头的 Host 属性组成完整的请求 URL。

③是协议名称及版本号。

2) 请求头：

④是 HTTP 的报文头，报文头包含若干个属性，格式为“属性名:属性值”，服务端据此获取客户端的信息。与缓存相关的规则信息，均包含在 header 中。

3) 请求体：

⑤是报文体，它将一个页面表单中的组件值通过 param1=value1&param2=value2 的键值对形式编码成一个格式化串，它承载多个请求参数的数据。不但报文体可以传递请求参数，请求 URL 也可以通过类似于“/chapter15/user.html?param1=value1&param2=value2”的方式传递请求参数。

HTTP 请求报文头属性：

1) Accept

请求报文可通过一个“Accept”报文头属性告诉服务端，客户端接受什么类型的响应。Accept 属性的值可以为一个或多个 MIME 类型的值（描述消息内容类型的因特网标准， 消息能包含文本、图像、音频、视频以及其他应用程序专用的数据）。

2) Cookie

客户端的 Cookie 就是通过这个报文头属性传给服务端的。 服务端是如何知道客户端的多个请求是隶属于一个 Session？就是通过 HTTP 请求报文头的 Cookie 属性的 jsessionid 的值关联起来的。

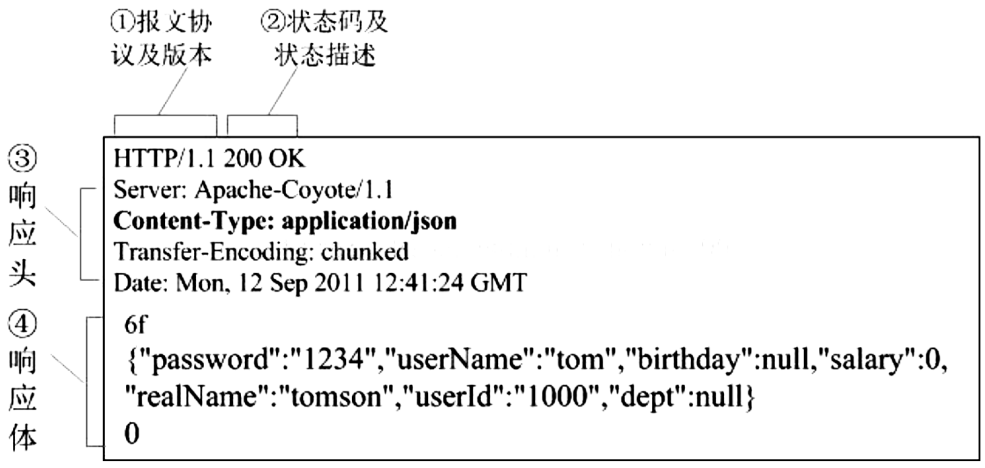
3) Referer

表示这个请求是从哪个 URL 过来的。

4) Cache-Control

对缓存进行控制，例如一个请求希望响应返回的内容在客户端要被缓存一年或不希望被缓存，就可以通过这个报文头达到目的。

HTTP 响应报文：



1) 响应行：

① 报文协议及版本；

② 状态码及状态描述；

2) 响应头：

③ 响应报文头，也是由多个属性组成；

3) 响应体：

④ 响应报文体；

4) 响应状态码：

和请求报文相比，响应报文多了一个“响应状态码”，它以“清晰明确”的语言告诉客户端本次请求的处理结果。

HTTP 的响应状态码由 5 段组成：

1xx	告诉客户端，请求已收到
2xx	处理成功
3xx	重定向，让客户端再发起一个请求
4xx	处理错误，一般是客户端异常
5xx	处理错误，一般是服务器异常

以下是几个常见的状态码：

200 OK	请求成功
303 See Other	查看其他地址
304 Not Modified	资源未修改，可读取本地缓存
403 Forbidden	服务器拒绝执行客户端请求
404 Not Found	服务器无法找到客户端的请求
500 Internal Server Error	服务器内部错误，无法完成请求
502 Bad Gateway	网关或者代理的服务器执行请求时， 从远程服务器收到无效的响应
503 Service Unavailable	由于超载或者维护，服务器无法处理请求
504 Gateway Time-out	网关或者代理服务器未能及时从远程服务器获

HTTP 响应报文头属性：

1) Cache-Control

响应输出到客户端后，服务端通过该报文头属告诉客户端如何控制响应内容的缓存。常见的取值有 private、public、no-cache、max-age，no-store，默认为 private，缓存时间为 31536000 秒（365 天）；也就是说，在 365 天内再次请求这条数据，都会直接获取缓存数据库中的数据，直接使用。

2) Etag

一个代表响应服务端资源（如页面）版本的报文头属性，如果某个服务端资源发生变化了，这个 ETag 就会相应发生变化。它是 Cache-Control 的有益补充，可以让客户端“更智能”地处理什么时候要从服务端取资源，什么时候可以直接从缓存中返回响应。

3) Location

在 JSP 中让页面 Redirect 到一个某个 A 页面中，其实是让客户端再发一个请求到 A 页面，这个需要 Redirect 到的 A 页面的 URL，其实就是通过响应报文头的 Location 属性告知客户端的。

4) Set-Cookie

服务端可以设置客户端的 Cookie，其原理就是通过这个响应报文头属性实现的：

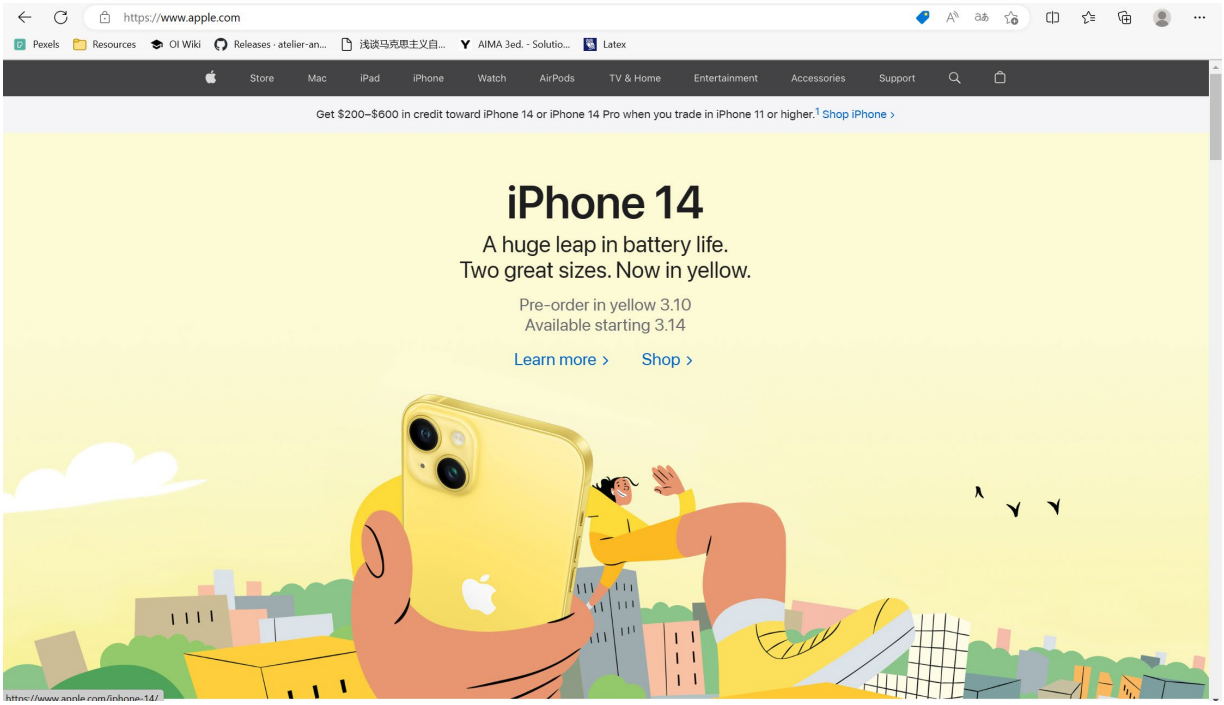
客户端请求服务器，如果服务器需要记录该用户状态，就使用 response 向客户端浏览器颁发一个 Cookie。客户端浏览器会把 Cookie 保存起来。当浏览器再请求该网站时，浏览器把请求的网址连同该 Cookie 一同提交给服务器。服务器检查该 Cookie，以此来辨认用户状态。服务器还可以根据需要修改 Cookie 的内容。

Cookie 的 maxAge 决定着 Cookie 的有效期，单位为秒（Second）。Cookie

中通过 getMaxAge()方法与 setMaxAge(int maxAge)方法来读写 maxAge 属性。如果 maxAge 属性为正数，则表示该 Cookie 会在 maxAge 秒之后自动失效。如果 maxAge 为负数，则表示该 Cookie 仅在本浏览器窗口以及本窗口打开的子窗口内有效，关闭窗口后该 Cookie 即失效。如果 maxAge 为 0，则表示删除该 Cookie。Cookie 并不提供修改、删除操作。如果要修改某个 Cookie，只需要新建一个同名的 Cookie，添加到 response 中覆盖原来的 Cookie。如果要删除某个 Cookie，只需要新建一个同名的 Cookie，并将 maxAge 设置为 0，并添加到 response 中覆盖原来的 Cookie。

四、实验步骤

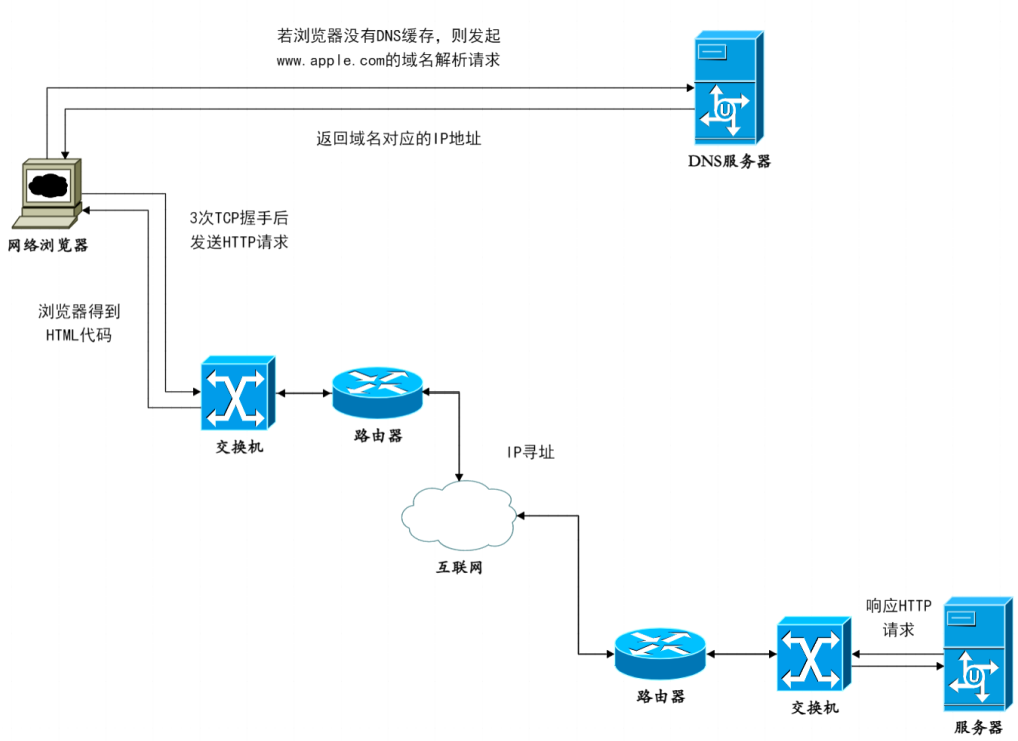
1. 选择一个网站（www.apple.com）；



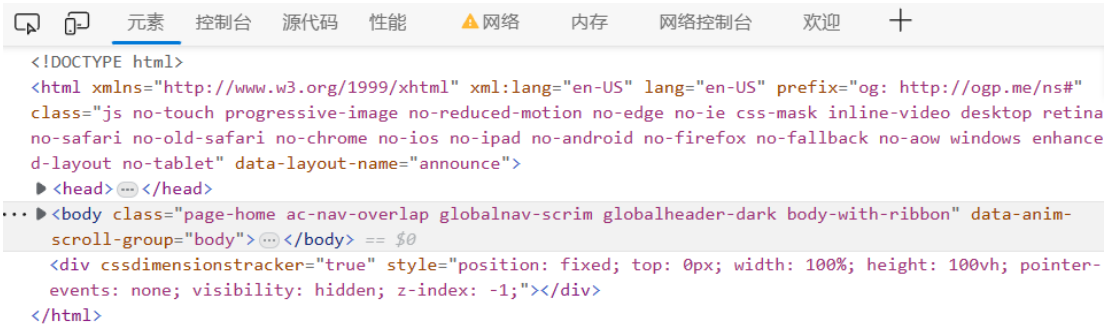
2. 登陆该网站，多次操作，请求页面，使用浏览器的开发者工具，查看 http 协议内容（具体内容见下文）。

五、实验内容

5.1 网络拓扑与数据流向图



5.2 分析网页的 HTML 组成要素



网页包含一个描述 html 文档类型的<html>标签，其包含<head>与<body>两个子标签。其中<head>包含关于文档的元信息，<body>，元素包含可见页面的所有内容。

<head>标签中包含如下子标签：

```
<link rel="alternate" href="https://www.apple.com/sa/" hreflang="en-SA">
<link rel="alternate" href="https://www.apple.com/se/" hreflang="sv-SE">
<link rel="alternate" href="https://www.apple.com/sg/" hreflang="en-SG">
<link rel="alternate" href="https://www.apple.com/si/" hreflang="en-SI">
<link rel="alternate" href="https://www.apple.com/sk/" hreflang="sk-SK">
<link rel="alternate" href="https://www.apple.com/sn/" hreflang="fr-SN">
<link rel="alternate" href="https://www.apple.com/th/" hreflang="th-TH">
<link rel="alternate" href="https://www.apple.com/tj/" hreflang="en-TJ">
<link rel="alternate" href="https://www.apple.com/tm/" hreflang="en-TM">
<link rel="alternate" href="https://www.apple.com/tn/" hreflang="fr-TN">
<link rel="alternate" href="https://www.apple.com/tr/" hreflang="tr-TR">
<link rel="alternate" href="https://www.apple.com/tw/" hreflang="zh-TW">
<link rel="alternate" href="https://www.apple.com/ua/" hreflang="uk-UA">
<link rel="alternate" href="https://www.apple.com/ug/" hreflang="en-UG">
<link rel="alternate" href="https://www.apple.com/uk/" hreflang="en-GB">
<link rel="alternate" href="https://www.apple.com/uz/" hreflang="en-UZ">
<link rel="alternate" href="https://www.apple.com/vn/" hreflang="en-VN">
<link rel="alternate" href="https://www.apple.com/za/" hreflang="en-ZA">
<link rel="alternate" href="https://www.apple.com.cn/" hreflang="zh-CN">
<meta name="viewport" content="width=device-width, initial-scale=1, viewport-fit=cover">
<link rel="stylesheet" type="text/css" href="/api-www/global-elements/global-header/v1/assets/globalheader.css">
<link rel="stylesheet" type="text/css" href="/ac/globalfooter/8/en_US/styles/ac-globalfooter.built.css">
<link rel="stylesheet" type="text/css" href="/ac/localnav/8/styles/ac-localnav.built.css">
<link rel="stylesheet" type="text/css" href="/ac/localeswitcher/4/zh_CN/styles/localeswitcher.built.css">
<title>Apple</title>
<meta property="analytics-track" content="Apple - Index/Tab">
<meta property="analytics-s-channel" content="homepage">
<meta property="analytics-s-bucket-0" content="applestoreww">
<meta property="analytics-s-bucket-1" content="applestoreww">
<meta property="analytics-s-bucket-2" content="applestoreww">
<meta name="Description" content="Discover the innovative world of Apple and shop everything iPhone, iPad, Apple Watch, M
ac, and Apple TV, plus explore accessories, entertainment, and expert device support.">
<meta property="og:title" content="Apple">
<meta property="og:description" content="Discover the innovative world of Apple and shop everything iPhone, iPad, Apple W
atch, Mac, and Apple TV, plus explore accessories, entertainment, and expert device support.">
<meta property="og:url" content="https://www.apple.com/">
<meta property="og:locale" content="en_US">
<meta property="og:image" content="https://www.apple.com/ac/structured-data/images/open_graph_logo.png?202110180743">
<meta property="og:type" content="website">
<meta property="og:site_name" content="Apple">
<link rel="stylesheet" href="/wss/fonts?families=SF+Pro,v3|SF+Pro+Icons,v3" type="text/css" media="all">
<link rel="stylesheet" href="/v/home/ay/built/styles/main.built.css" type="text/css">
<script src="/v/home/ay/built/scripts/head.built.js" type="text/javascript" charset="utf-8"></script>
</head>
```

1) <meta>标签：

<meta>元素定义文档元数据，常用来描述当前页面的特性。如“charset = “utf-8”” 表明页面使用 UTF-8 字符集编码；再如，“property=“og:xxx”” 网页遵守 Facebook 公布的开放内容协议(Open Graph Protocol)，SNS 网站能从页面上提取最有效的信息并呈现给用户。

2) <link>标签：

外部资源链接元素 (<link>) 规定了当前文档与外部资源的关系。图中<link>的 rel 属性对应的“alternate”表明这些 link 都是可供选择的，即不同语言对应的苹果官网。href 提供了各个官网的 url 链接，hreflang 表明了具体对应哪一种语言。

3) <title >标签：

<title>元素定义文档的标题，显示在浏览器的标题栏或标签页上。此处即对应显示在标签页上的字样“Apple”。

4) <script>标签：

<script> 元素用于嵌入或引用可执行脚本，通常用作嵌入或者指向 JavaScript 代码。

<body>标签中包含如下子标签：

```
<body class="page-home ac-nav-overlap globalnav-scrim globalheader-dark body-with-ribbon" data-anim-scroll-group="body"> == $0
  <h1 class="visuallyhidden">Apple</h1>
  <meta name="globalnav-store-key" content="SFX9YPYY9PPXCU9KH">
  <div id="globalheader">…</div>
  <script id="__ACGH_DATA__" type="application/json">…</script>
  <script type="text/javascript" src="/api-www/global-elements/global-header/v1/assets/globalheader.umd.js"></script>
  <script src="/metrics/ac-analytics/2.17.1/scripts/ac-analytics.js" type="text/javascript" charset="utf-8"></script>
  <main class="main" role="main">…</main>
  <footer class="js" lang="en-US" id="ac-globalfooter" data-analytics-region="global footer" role="contentinfo" aria-
  labelledby="ac-gf-label">…</footer>
  <script type="application/ld+json">…</script>
  <script type="application/ld+json">
        {
            "@context": "http://schema.org",
            "@id": "https://www.apple.com/#webpage",
            "@type": "WebPage",
            "url": "https://www.apple.com/",
            "name": "Apple"
        }
  </script>
  <script src="/v/home/ay/built/scripts/main.built.js" type="text/javascript" charset="utf-8"></script>
  <span style="visibility: hidden; position: absolute; top: 0px; bottom: 0px; z-index: -1;">…</span>
  <div id="viewport-emitter" data-viewport-emitter-dispatch data-viewport-emitter-state="{"viewport":"xlarge","orientation":"l
  andscape","retina":true}">…</div>
  <link rel="stylesheet" href="/ac/ac-films/6.8.2/styles/modal.css">
  <script src="/ac/ac-films/6.8.2/scripts/autofilms.built.js" type="text/javascript"></script>
  <script src="/metrics/data-relay/1.1.4/scripts/data-relay.js" type="text/javascript" charset="utf-8"></script>
  <script src="/metrics/data-relay/1.1.4/scripts/auto-relay.js" type="text/javascript" charset="utf-8"></script>
</body>
```

1) <meta>标签（同<head>中）

2) <script>标签（同<head>中）

3) <link>标签（同<head>中）

4) <div 标签>

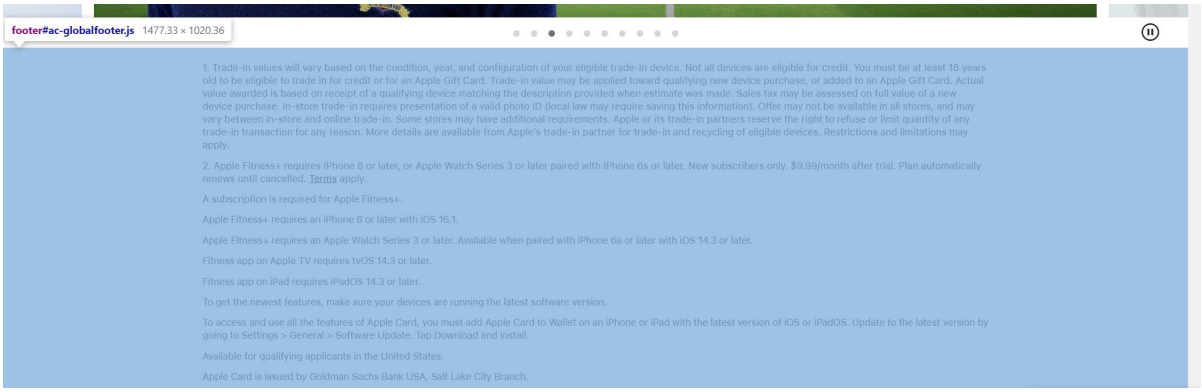
 元素即 HTML 文档分区元素是一个通用型的流内容容器。在不使用 CSS 的情况下，其对内容或布局没有任何影响。通过 标签可将页面不同的部分进行分组，如顶端的 header 选择栏就在一个 div 分组中。

5) <main>标签

<main> 元素呈现了文档的 <body> 或应用的主体部分。主体部分由与文档直接相关，或者扩展于文档的中心主题、应用的主要功能部分的内容组成。

6) <footer>标签

<footer> 元素表示最近一个章节内容或者根节点（sectioning root）元素的页脚。一个页脚通常包含该章节作者、版权数据或者与文档相关的链接等信息。



7) <h1>标签（<h1>-<h6>）

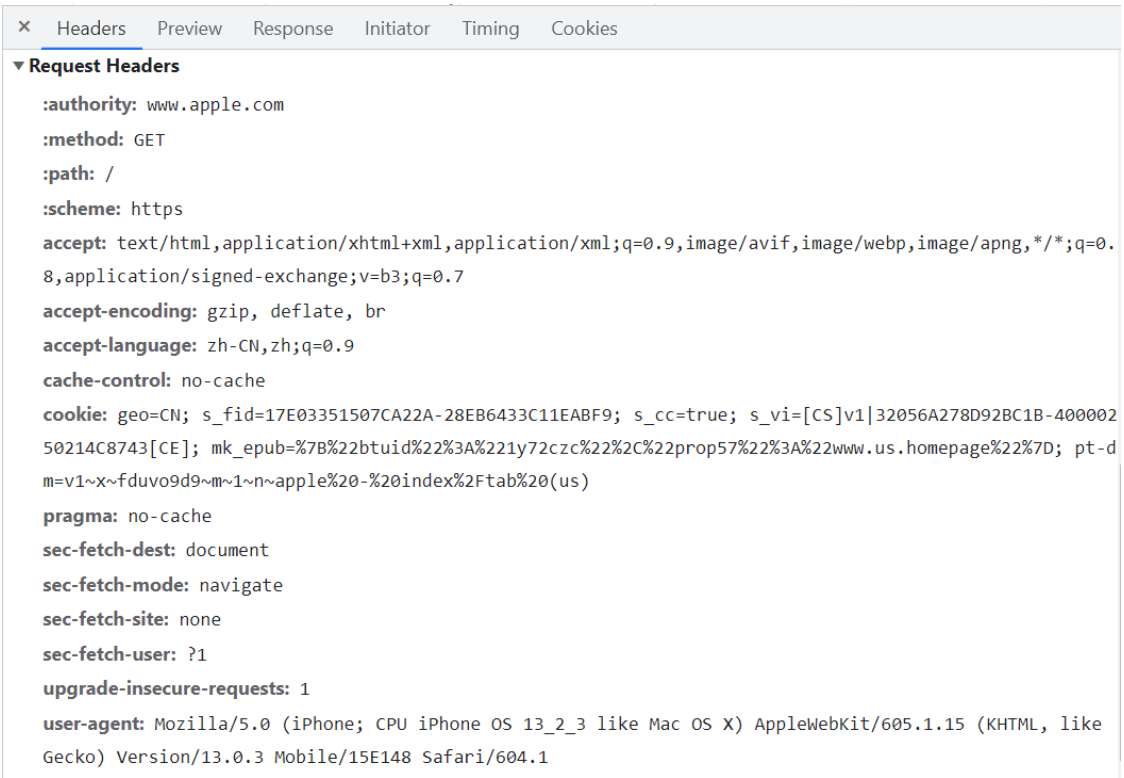
<h1>－<h6> 标题 (Heading) 元素呈现了从一级到六级由大到小的六个不同的级别的标题，其中<h1>级别最高，而 <h6>级别最低。

5.3 分析网页的 HTTP 协议

苹果官网采用了 HTTP2 协议，因此与前文所述的 HTTP1 协议有些许不同。HTTP2 协议进行了二进制分帧，而非直接传输可见的 ASCII 码。同时，头部压缩算法将实际发送的 header 采用下图静态表中的索引值而代替。如，若发送“:method = GET”，则可以用索引值 2 表示。因此，直接查看源字段信息是没有意义的，浏览器的 network 中也不提供 HTTP2 的“view source”选项。

Index	Header Name	Header Value
1	:authority	
2	:method	GET
3	:method	POST
4	:path	/
5	:path	/index.html
6	:scheme	http
7	:scheme	https
8	:status	200
9	:status	204
10	:status	206
11	:status	304
12	:status	400

5.3.1 请求报文头



:authority：表示资源所在的主机名，此处为 www.apple.com。

:method：如上文所述表示请求方法，此处为 GET。

:path：主机上资源所在的路径，此处“/”即在根目录下。

:scheme：所使用的协议名，此处为 https。

Accept ：表示浏览器可以接收的内容类型（Content-types），此处表示可以接受 text/html 等类型的内容。

Accept-Encoding：表示浏览器可以处理的编码方式，此处表示浏览器可以接收 gzip,deflate 以及 br 的编码方式。

Accept-Language：浏览器接收的语言，即表示用户所在的语言地区，例如简体中文的就是 Accept-Language: zh-CN。

Cache-Control：指示缓存系统应该怎样处理缓存，此处表示需要使用对比缓存来验证缓存数据。

Cookie：同上文介绍类似，浏览器向服务器发送请求时发送 cookie 即在此处。

User-Agent：发出请求的用户信息为 Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.0.0 Safari/537.36。

5.3.2 响应报文头

Cache-Control：指示浏览器应该怎样处理缓存，此处表明失效的时间为 34261 秒。

Content-Encoding：表示 web 服务器支持的返回内容压缩编码类型为 gzip。

Content-Length：表示响应体的长度为 34261。

Content-Type : 表示返回的内容类型为 text/html，字符集是 utf-8。

Date：原始服务器消息发出的时间为 Fri, 10 Mar 2023 07:17:17 GMT。

Expires：响应过期的日期和时间为 Fri, 10 Mar 2023 07:20:35 GMT。

六、心得体会

在这次的万维网运行原理分析实验中，我了解到了通过 HTTP 协议来发送请求报文信息、接收响应报文信息的过程，从而了解到服务器与客户端之间建立联系并发送信息的方式。同时，我也学习到了报文信息头与响应信息头的具体格式，对报文关键点进行了分析。在通过浏览器的开发者工具查看苹果公司官网时，我发现其使用的 HTTP2 协议与课上所学习的 HTTP1 协议的差异，因此扩展了解了 HTTP2 协议的报文形式。此外，我也了解了 html 网页中重要的标签及其含义。

通过本次实验，我了解到，平常看似简单的网页加载其实背后有着复杂的过程；同时，我也探究了解了服务器与客户端建立联系的过程与报文的格式。此外，在查看苹果官网的 html 源代码时，我也深刻感受到了其代码的规范性与完备性。无论是 html 的逐步完善还是 HTTP 协议不断发展，都令我对 web 技术的飞速进步感到钦佩，也希望在未来的学习过程中能够将本次学习到的知识更好地学以致用。