



웹 브라우저 주소창에 **www.naver.com**을 입력하고 엔터를 누르면 어떤 일이 일어날까?



목차

1. 웹 브라우저 주소창에 **www.naver.com**을 입력한 뒤에 엔터를 누른다
2. 웹 브라우저가 입력된 URL(**www.naver.com**)이 문법에 맞는지 해석한다
3. URL(**www.naver.com**)이 웹 브라우저에 캐싱되어 DNS 기록으로 존재하는지 확인한다
4. 만약 입력한 도메인이 캐싱되지 않은 상태라면 ISP의 DNS 서버가 해당 도메인의 IP 주소를 찾기 위해 DNS Query(DNS 질의)를 보낸다
5. 라우터를 통해 서버의 게이트웨이까지 이동하고, ARP를 통해 IP 주소를 MAC 주소로 변환한다
6. 브라우저가 해당 도메인의 IP에서 해당 포트 번호를 가져와 TCP 소켓을 열고 3-way handshake로 연결을 설정한다
7. 웹 브라우저는 웹 서버에게 IP 주소에 해당하는 페이지의 데이터를 달라고 HTTP GET 요청을 보낸다
8. 서버는 request를 처리하고 response를 생성한다
9. 서버가 HTTP response를 보낸다
10. 웹 브라우저는 서버로부터 응답받은 HTML, CSS, JS, 이미지 등 자원을 통하여 웹을 파싱 및 렌더링하고 사용자는 입력했던 **www.naver.com**을 볼 수 있다

1. 웹 브라우저 주소창에 **www.naver.com**을 입력한 뒤에 엔터를 누른다

- 사용자가 웹 브라우저 주소창에 키 하나를 누른 순간 자동완성 기능이 활성화
- 브라우저는 사용자의 검색 및 방문 기록이나 즐겨찾기에 기반하여 매 키를 누를 때마다 매번 선별된 제안 제공
- URL을 모두 다 입력하기도 전에 **www.naver.com**을 먼저 제안 가능

2. 웹 브라우저가 입력된 URL(**www.naver.com**)이 문법에 맞는지 해석한다

- URL의 문법 및 예시 :
https://developer.mozilla.org/ko/docs/Learn/Common_questions/What_is_a_URL
- URL이 문법에 맞다면 URL의 호스트 부분을 인코딩
 - **www.naver.com**만 입력한 경우 HTTP 프로토콜을 사용하여 접속 시도
 - 만약 해당 사이트가 HTTPS 프로토콜이 적용된 경우라면 301, 302 상태 코드를 통해 리다이렉트하여 다시 HTTPS 프로토콜로 접속하게 된다
 - 한번이라도 HTTPS 프로토콜로 접속한 적이 있다면 HSTS 목록에 저장된다
 - 이때 미리 불러들인 HSTS (HTTP Strict Transport Security) 목록을 확인
 - 목록에 존재한다면 HTTPS로, 없다면 HTTP로 요청
 - HSTS : 요청한 웹 서버가 HTTPS로 설계되어 있을 때 클라이언트 웹 브라우저에게 HTTPS로 요청하라고 알려주는 보안 기능

HTTP와 HTTPS의 차이는? <https://github.com/ssafy-tech-concert/ssafy-tech->

| concert/blob/master/web/HTTP_HTTPS.md#1

- **퓨니코드(Punycode)** 인코딩 : 유니코드 문자열을 호스트 이름에서 허용된 문자만으로 인코딩하는 방법
- URL이 문법에 맞지 않다면 브라우저의 기본 검색 엔진(google, bing 등)으로 검색

3. URL(www.naver.com)이 웹 브라우저에 캐싱되어 DNS 기록으로 존재하는지 확인한다

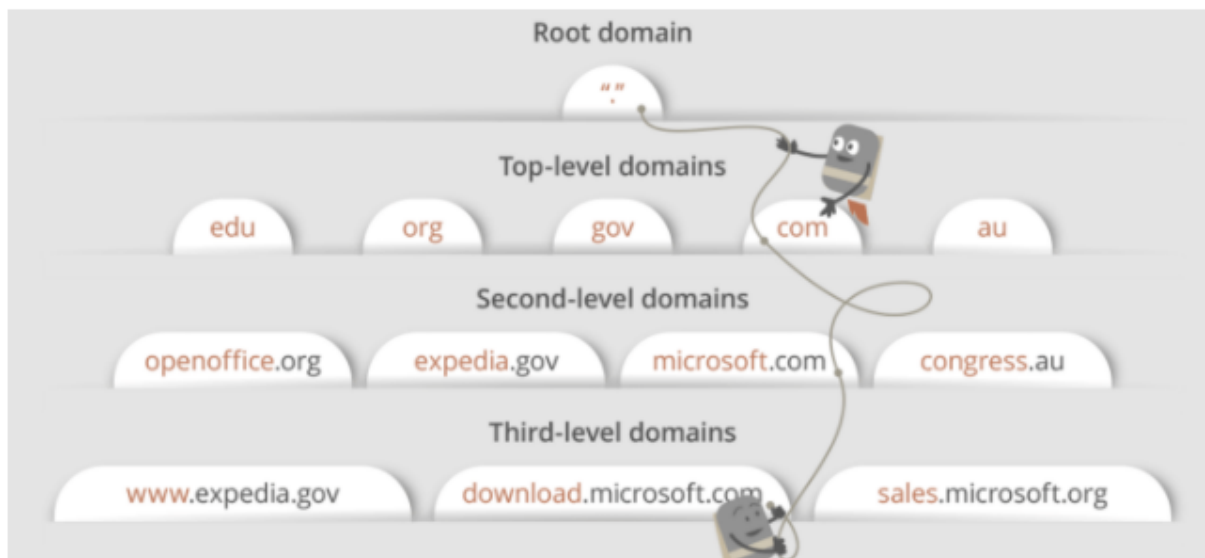
| 캐싱이란? <https://github.com/ssafy-tech-concert/ssafy-tech-concert/blob/master/web/Cache.md#5>

- **DNS (Domain Name System)** : 인터넷상의 도메인 이름을 실제로 접속 가능한 주소로 식별해서 찾을 수 있도록 하는 서비스
 - **Domain Name** ↔ **IP Address**
 - Domain Name → IP Address : 컴퓨터의 주소를 찾기 위함
 - IP Address → Domain Name : 사람이 사용하기 쉽게 하기 위함
 - **IP Address** 확인하는 방법 : nslookup www.naver.com
- 우선 웹 브라우저는 일정 기간 동안 DNS 기록을 저장하고 있기 때문에 브라우저 캐시에서 웹 사이트 이름의 IP 주소를 확인
- 다음으로 OS 캐시 확인. syscall을 통해 OS가 저장하고 있는 DNS 기록들의 캐시에 접근하여 확인
 - syscall : 운영체제의 수준에서 커널 영역을 사용자 모드의 프로그램이 사용할 수 있게 해주는 기능
- 그 다음으로 DNS 기록을 캐싱하고 있으며 OSI 7계층 중 데이터 통신 계층에 있는 router와 통신하여 캐시 확인

데이터 통신 계층이란? [https://github.com/ssafy-tech-concert/ssafy-tech-concert/blob/master/Computer-Science/OSI 7 layer.md#2](https://github.com/ssafy-tech-concert/ssafy-tech-concert/blob/master/Computer-Science/OSI%207%20layer.md#2)

- 마지막으로 DNS 서버를 구축하고 있는 ISP 캐시 확인
 - ISP (Internet Service Provider) (인터넷 서비스 제공자) : SKT, KT, LG 유플러스
 - 와이파이가 연결된 순간 ISP로부터 사용자 컴퓨터로 도메인 네임 서버의 IP를 세팅
- 캐싱되어 있는 도메인이었다면 캐싱되어 있는 데이터를 참조

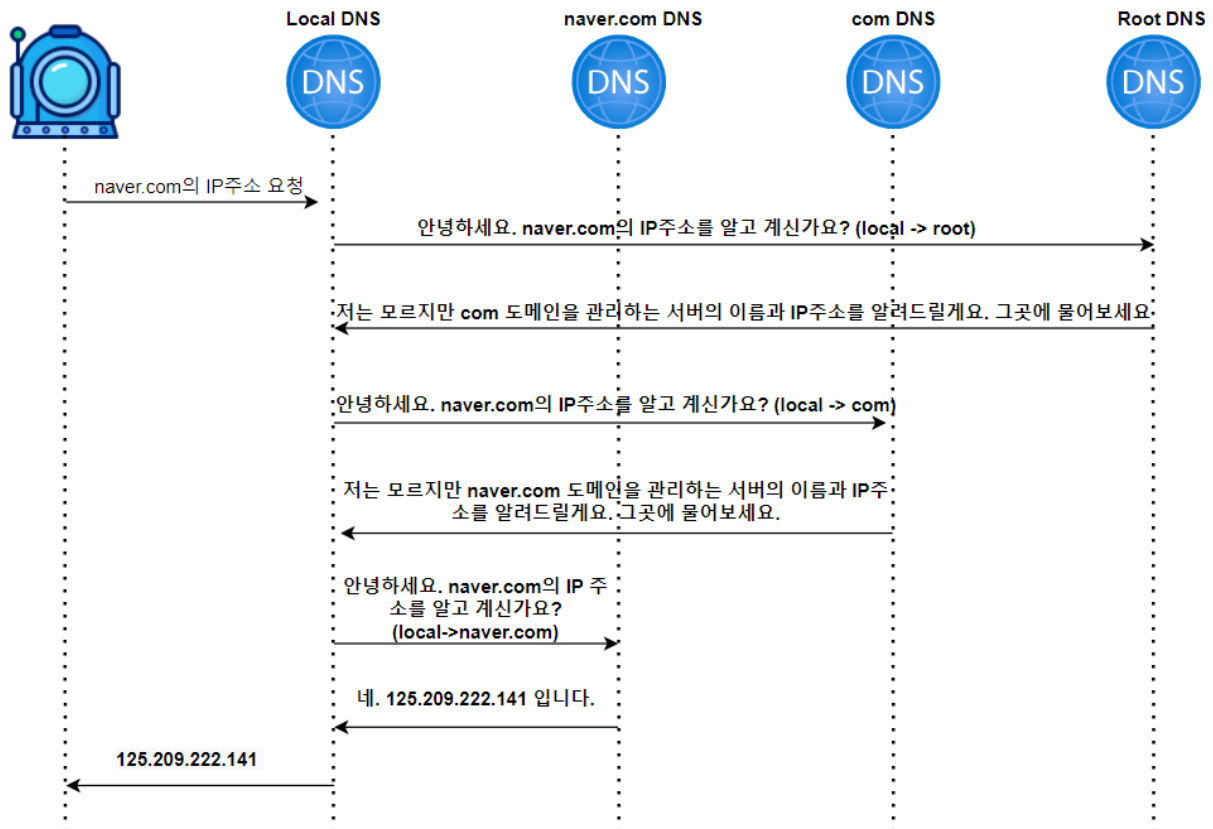
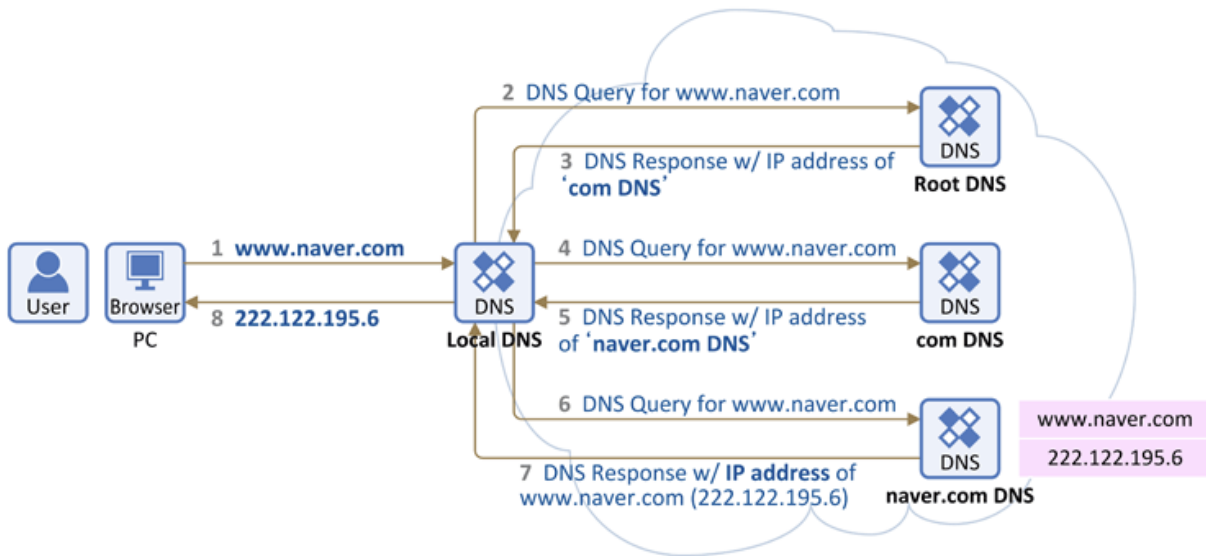
4. 만약 입력한 도메인이 캐싱되지 않은 상태라면 ISP의 DNS 서버가 해당 도메인의 IP 주소를 찾기 위해 DNS Query(DNS 질의)를 보낸다





- **Root Domain**
 - 모든 도메인의 뿌리
 - 사람이 읽을 수 있는 호스트 이름을 IP 주소로 변환(확인)하는 첫 번째 단계
 - 사이트 주소 뒤에 생략된 **.**
 - blog.example.com.으로 주소창에 입력해도 blog.example.com와 동일하게 작동
 - Root 도메인을 다루는 DNS Server는 Top-level Domain을 담당하는 DNS Server를 알고 있다
- **Top-level Domain (TLD) (최고레벨 도메인, 최상위 도메인)**
 - 가장 일반적인 정보 제공
 - 인터넷에서 도메인 네임의 가장 마지막 부분
 - 도메인을 구분하는 가장 큰 카테고리이며 도메인을 성격에 따라서 분류
 - 일반 최상위 도메인: 국가별로 고유하지 않은 도메인 (.com, .org, .net, .edu, .gov)
 - 국가 코드 최상위 도메인: 여기에는 국가 또는 주와 관련된 모든 도메인 (.uk, .us, .ru, .jp)
- **Second-level Domain (SLD)**
- **subdomain**
- 각각의 도메인은 DNS Server를 각각 가지고 있고 각각 전담하는 파트는 다르다

- 상위 DNS Server는 직접 연결된 하위 DNS Server를 알고 있다. 연결되지 않은 하위 DNS Server는 모른다.



- Recursive Search

- ISP가 제공하는 DNS 서버인 **DNS recursor**에서 DNS 정보를 찾으려 다른 DNS 서버들로 DNS query를 보낸다
- DNS 서버에서 다른 DNS 서버로 반복적으로 이동하면서 IP 주소를 찾을 때까지 또는 못 찾아서 에러가 발생할 때까지 검색 진행
- DNS query와 DNS recursor의 IP 주소가 포함된 패킷을 통해 전달 → 패킷이 유실되면 request fail error 발생
- IP 주소를 찾아내면 해당 IP 주소로 HTTP 요청을 보낸다

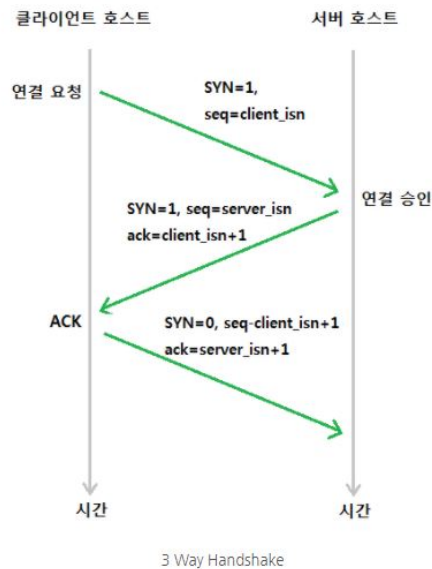
5. 라우터를 통해 서버의 게이트웨이까지 이동하고, ARP를 통해 IP 주소를 MAC 주소로 변환한다

- 실질적인 통신을 위해 논리 주소인 IP 주소를 물리 주소인 MAC 주소로 변환해야 한다
 - 접속하려는 서버의 네트워크를 찾기 위해 IP주소를 사용
 - 그 네트워크 내부에 있는 컴퓨터와 통신하기 위해 MAC 주소가 필요
- router 내에 존재할 경우 routing table을 추적하여 MAC 주소를 찾고, Local Network가 아니라면 게이트웨이를 타고 밖으로 나가 MAC 주소를 찾는다
- **ARP (Address Resolution Protocol)** : IP 주소 → MAC 주소 변환하는 프로토콜
 - 송신측은 MAC 주소가 필요하므로 ARP 요청 패킷을 브로드캐스트 방식으로 전달
 - 최종 목적지의 물리 주소를 모르기 때문에 모두에게 요청
 - 모든 Host와 Router는 송신자가 보낸 ARP 요청 수락
 - 해당되는 수신자만 자신의 IP주소와 MAC 주소를 넣어 응답
- RARP (Reverse Address Resolution Protocol) : MAC 주소 → IP 주소

6. 브라우저가 해당 도메인의 IP에서 해당 포트 번호를 가져와 TCP 소켓을 열고 3-way handshake로 연결을 설정한다

TCP란? https://github.com/ssafy-tech-concert/ssafy-tech-concert/blob/master/Computer-Science/TCP_UDP.md#1

3-Way Handshake : 클라이언트와 서버가 SYN과 ACK메세지들을 가지고 3번의 프로세스를 거친 후 연결



- 클라이언트 머신이 SYN 패킷을 서버에 보내고 connection을 열어달라고 물어본다
 - 서버가 새로운 connection을 시작할 수 있는 포트가 있다면 SYN/ACK 패킷으로 대답을 한다
 - 클라이언트는 SYN/ACK 패킷을 서버로부터 받으면 서버에게 ACK 패킷을 보낸다
-
- 이 과정이 끝나면 TCP connection 완성
 - HTTPS 요청이라면 추가적인 TLS(Transport Layer Security) handshake를 통하여 세션 키를 생성
 - 세션이 유지되는 동안 요청과 응답을 반복하는데 응답을 캐싱할 필요가 있다면 캐싱

7. 웹 브라우저는 웹 서버에게 IP 주소에 해당하는 페이지의 데이터를 달라고 GET 요청을 보낸다

웹 서버란? [https://github.com/ssafy-tech-concert/ssafy-tech-concert/blob/master/Back-end/Servlet %26 JSP.md#2](https://github.com/ssafy-tech-concert/ssafy-tech-concert/blob/master/Back-end/Servlet%20JSP.md#2)

- TCP로 연결이 되었다면, HTTP Request가 TCP Socket을 통해 전송된다

```
GET http://facebook.com/ HTTP/1.1
Accept: application/x-ms-application, image/jpeg, application/xaml+xml, [...]
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; [...])
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
Host: facebook.com
Cookie: datr=1265876274-[...]; locale=en_US; lsd=WW[...]; c_user=2101[...]
```

- 전송되는 HTTP Headers
 - Accept 헤더 : 받아들일 요청의 종류
 - User-Agent 헤더 : browser identification
 - connection 헤더 : 추가적인 요청을 위해 TCP connection 유지를 요청
 - 브라우저에서 얻은 쿠키 정보, Accept-Encoding, Host 등

8. 서버는 request를 처리하고 response를 생성한다

- 서버는 브라우저로부터 요청을 받고 request handler로 요청을 전달
- request handler는 요청과 요청의 헤더, 쿠키를 읽어서 요청이 무엇인지 파악하고 필요하다면 서버에 정보를 업데이트
- response는 특정한 포맷(JSON, XML, HTML)으로 작성

쿠키란? [https://github.com/ssafy-tech-concert/ssafy-tech-concert/blob/master/Computer-Science/Session %26 JWT.md#2](https://github.com/ssafy-tech-concert/ssafy-tech-concert/blob/master/Computer-Science/Session%26JWT.md#2)

9. 서버가 HTTP response를 보낸다

```
HTTP/1.1 200 OK
Cache-Control: private, no-store, no-cache, must-revalidate, post-check=0,
    pre-check=0
Expires: Sat, 01 Jan 2000 00:00:00 GMT
P3P: CP="DSP LAW"
Pragma: no-cache
Content-Encoding: gzip
Content-Type: text/html; charset=utf-8
X-Cnection: close
Transfer-Encoding: chunked
Date: Fri, 12 Feb 2010 09:05:55 GMT
```

- 서버의 response
 - 요청한 웹페이지
 - Status Code(응답 종류)
 - 첫번째 줄

Status Code(응답 종류)는 무엇이 있을까?

<https://github.com/ssafy-tech-concert/ssafy-tech-concert/blob/master/web/REST.md#2>

- compression type(Content-Encoding) : 어떻게 인코딩 되어 있는지 정보
- Cache-Control : 어떻게 페이지를 캐싱할지 정보
- 설정할 쿠키가 있다면 쿠키, 개인정보 등

10. 웹 브라우저는 서버로부터 응답받은 HTML, CSS, JS, 이미지 등 자원을 통하여 웹을 파싱 및 렌더링하고 사용자는 입력했던 www.naver.com을 볼 수 있다

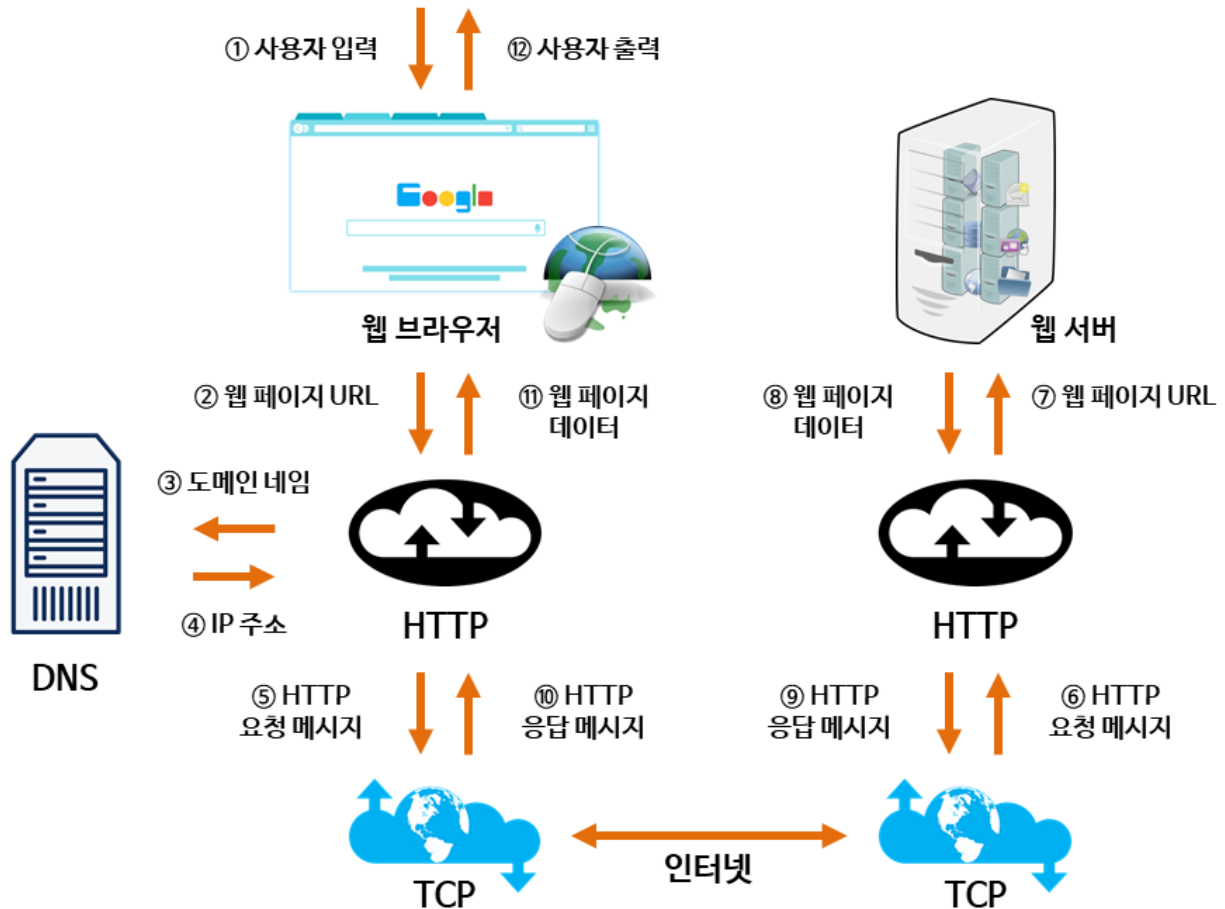
렌더링이란? <https://github.com/ssafy-tech-concert/ssafy-tech-concert/blob/master/web/SSRCSR.md#1>

- 파싱 : HTML, CSS, JS
- 렌더링 : DOM 트리 생성 → 트리 렌더링 → 렌더링 된 트리 배치 → 렌더링 된 트리 색칠

추천 자료

웹의 동작 원리(<http://tcpschool.com/webbasic/works>)

How the DNS works(<https://www.youtube.com/watch?v=2ZUxoi7YNgs>)



①② 사용자가 웹 브라우저를 통해 찾고 싶은 웹 페이지의 URL 주소를 입력함.

③ 사용자가 입력한 URL 주소 중에서 도메인 네임(domain name) 부분을 DNS 서버에서 검색함.

④ DNS 서버에서 해당 도메인 네임에 해당하는 IP 주소를 찾아 사용자가 입력한 URL 정보와 함께 전달함.

⑤⑥ 웹 페이지 URL 정보와 전달받은 IP 주소는 HTTP 프로토콜을 사용하여 HTTP 요청 메시지를 생성함.

이렇게 생성된 HTTP 요청 메시지는 TCP 프로토콜을 사용하여 인터넷을 거쳐 해당 IP 주소의 컴퓨터로 전송됨.

⑦ 이렇게 도착한 HTTP 요청 메시지는 HTTP 프로토콜을 사용하여 웹 페이지 URL 정보로 변환됨.

⑧ 웹 서버는 도착한 웹 페이지 URL 정보에 해당하는 데이터를 검색함.

⑨⑩ 검색된 웹 페이지 데이터는 또 다시 HTTP 프로토콜을 사용하여 HTTP 응답 메시지를 생성함.

이렇게 생성된 HTTP 응답 메시지는 TCP 프로토콜을 사용하여 인터넷을 거쳐 원래 컴퓨터로 전송됨.

⑪ 도착한 HTTP 응답 메시지는 HTTP 프로토콜을 사용하여 웹 페이지 데이터로 변환됨.

⑫ 변환된 웹 페이지 데이터는 웹 브라우저에 의해 출력되어 사용자가 볼 수 있게 됨.

1. 사용자가 브라우저에 **도메인 네임을 입력**한다. 'www.naver.com'
2. 사용자가 입력한 URL 주소 중 **도메인 네임 부분을 DNS 서버에 검색**하고, DNS서버에서 해당 도메인 네임에 해당하는 **IP주소를 찾아 사용자가 입력한 URL 정보와 함께 전달**한다.
3. 페이지 URL정보와 전달받은 IP주소는 HTTP 프로토콜을 사용하여 **HTTP 요청 메시지를 생성**하고, 이렇게 생성된 HTTP 요청 메시지는 **TCP 프로토콜을 사용하여 인터넷을 거쳐 해당 IP주소의 컴퓨터로 전송**된다.
4. 이렇게 도착한 HTTP 요청 메시지는 HTTP 프로토콜을 사용하여 **웹 페이지 URL 정보로 변환되어 웹 페이지 URL 정보에 해당하는 데이터를 검색**한다.
5. 검색된 웹 페이지 데이터는 또 다시 HTTP 프로토콜을 사용하여 **HTTP 응답 메시지를 생성**하고 TCP 프로토콜을 사용하여 인터넷을 거쳐 **원래 컴퓨터로 전송**된다.
6. 도착한 HTTP 응답 메시지는 HTTP 프로토콜을 사용하여 **웹 페이지 데이터로 변환**되어 웹 브라우저에 의해 출력되어 **사용자가 볼 수 있게 된다**.

출처

<https://1day1commit.tistory.com/149>

<https://oizys.tistory.com/36>

<https://1-7171771.tistory.com/134>

<https://github.com/SantonyChoi/what-happens-when-KR>
<https://www.cloudflare.com/ko-kr/learning/dns/dns-server-types/>
<https://kkoon9.tistory.com/69?category=931254>
https://velog.io/@jaeyunn_15/Network-www.naver.com을-치면-일어나는-일
<https://devjin-blog.com/what-happen-browser-search/>
<https://sorjfrh5078.tistory.com/65>
<https://sorjfrh5078.tistory.com/62>
<https://deveric.tistory.com/97>
https://github.com/vi2920va/FE_interview/blob/main/Network/google.com을 치면 일어나는 일.md
<https://blog.yevgnenll.me/posts/what-happen-browser-after-url-enter>
<https://owlgwang.tistory.com/1>
<https://rsec.kr/?p=315>