

HTTP 기반 기술

장동호

목차

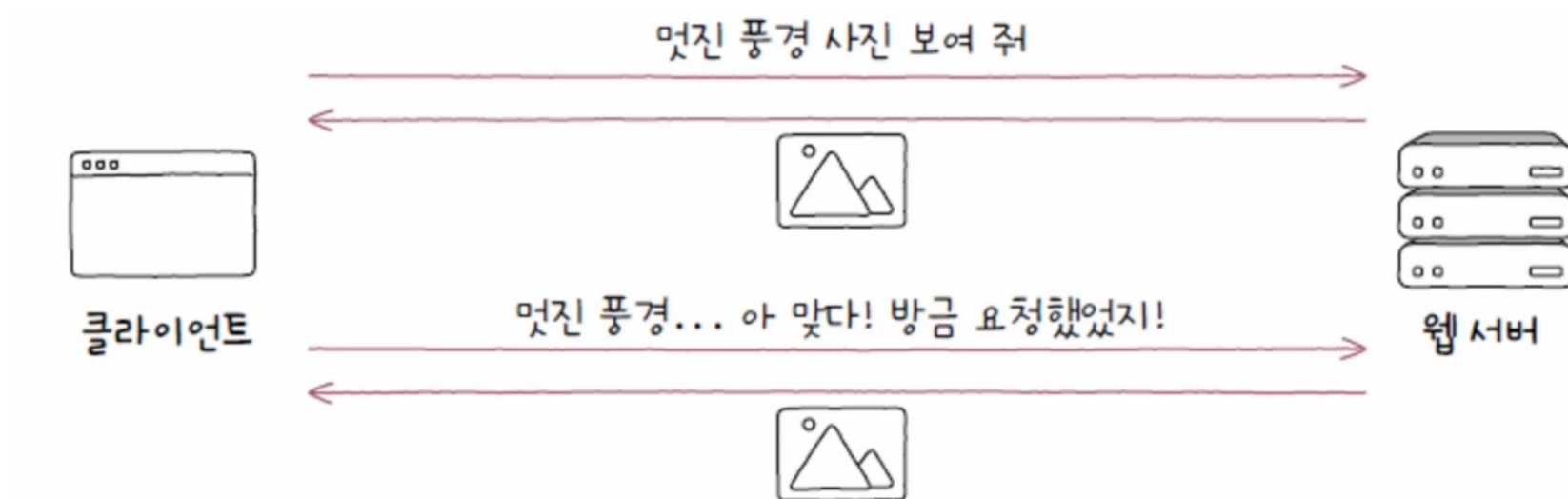
1. 캐시
2. 쿠키
3. 콘텐츠 협상

지난 시간 복습

- HTTP는 상태를 유지하지 않는 _____ 프로토콜이다.

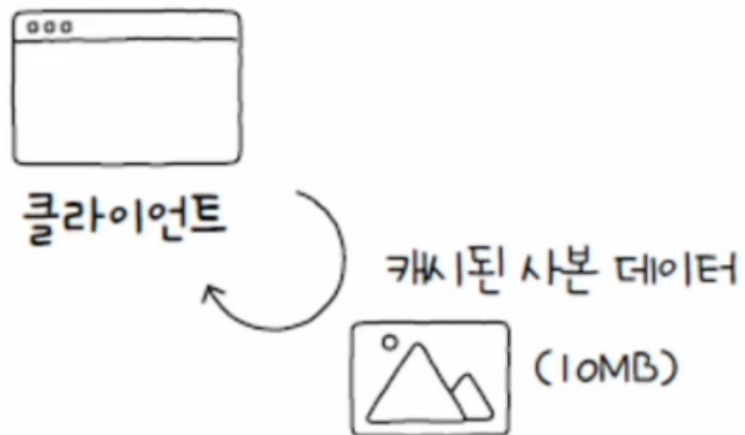
지난 시간 복습

- HTTP는 상태를 유지하지 않는 stateless 프로토콜이다.
 - 서버는 HTTP 요청을 보낸 클라이언트 관련 상태를 기억하지 않는다.



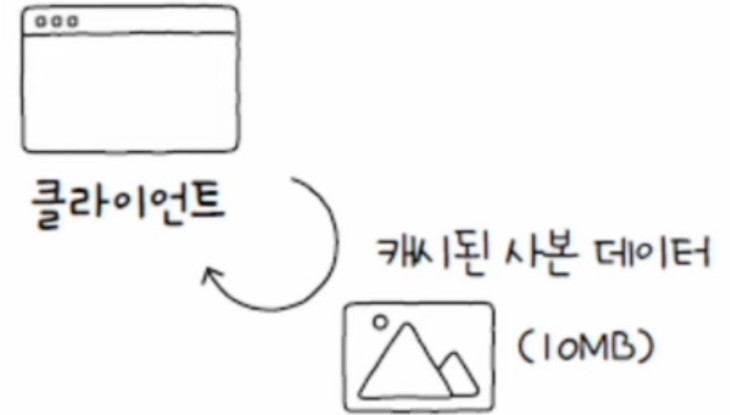
캐시

- 응답 받은 자원의 사본을 임시 저장하는 기술
 - 추후 동일한 요청에 대해 캐시된 데이터를 활용
 - 불필요한 대역폭 낭비 방지, 응답 지연 방지



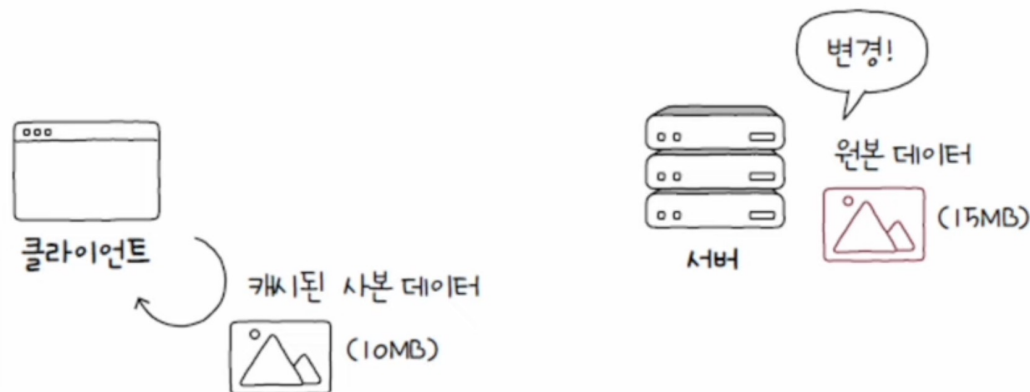
캐시

- 개인 전용 캐시(private cache)
 - 웹 브라우저에 저장
- 공용 캐시(public cache)
 - 클라이언트와 서버 사이에 위치한 중간 서버에 저장



캐시

- 캐시 신선도와 검사와 유지
 - 캐시된 데이터는 언제든지 원본 데이터와 달라질 수 있다.
 - 캐시 신선도(cache freshness)
 - 캐시된 사본 데이터가 얼마나 최신 원본 데이터와 유사한지를 표현



캐시

- 캐시 신선도 검사: 유효 기간 설정
 - 기간이 만료되었다면 원본 데이터를 다시 요청
 - 유효 기간 설정 방법: Expires 헤더(날짜) 또는 Cache-Control 헤더의 Max-Age 값(초)을 사용

```
HTTP/1.1 200 OK
Date: Mon, 05 Feb 2024 12:00:00 GMT
Content-type: text/plain
Content-length: 100
Expires: Tue, 06 Feb 2024 12:00:00 GMT
...본문 생략...
```

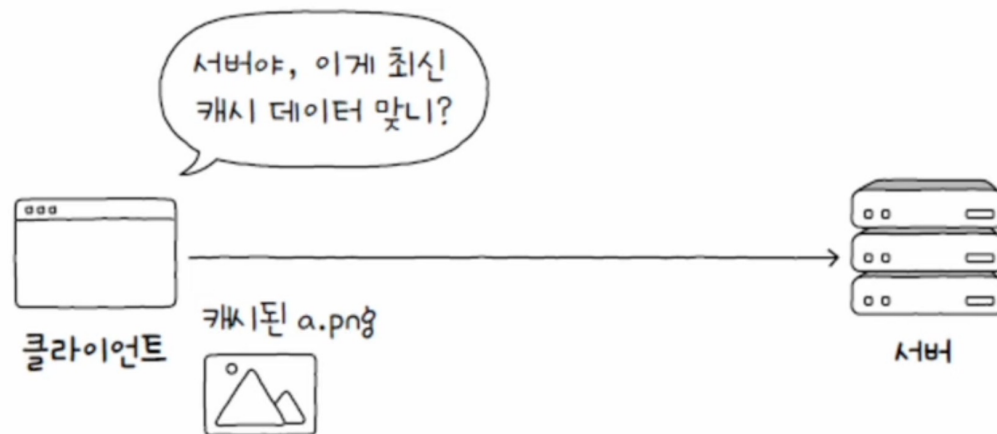
```
HTTP/1.1 200 OK
Date: Mon, 05 Feb 2024 12:00:00 GMT
Content-type: text/plain
Content-length: 100
Cache-Control: max-age=1200
...본문 생략...
```


캐시

- 캐시 신선도 재검사

1. If-Modified-Since 헤더: 날짜를 기반으로 서버에게 물어보는 방법

- 헤더에 명시된 시점 이후로 원본에 변경이 있었다면 그때만 새 자원으로 응답하도록 서버에게 요청하는 헤더



캐시

- 캐시 신선도 재검사

1. If-Modified-Since 헤더: 날짜를 기반으로 서버에게 물어보는 방법

- 헤더에 명시된 시점 이후로 원본에 변경이 있었다면 그때만 새 자원으로 응답하도록 서버에게 요청하는 헤더

요청 메시지

```
GET /index.html HTTP/1.1
Host: www.example.com
If-Modified-Since: Fri, 23 Aug 2024 09:00:00 GMT
```

캐시

1. 서버가 요청받은 자원이 변경된 경우
2. 서버가 요청받은 자원이 변경되지 않은 경우
3. 서버가 요청받은 자원이 삭제된 경우

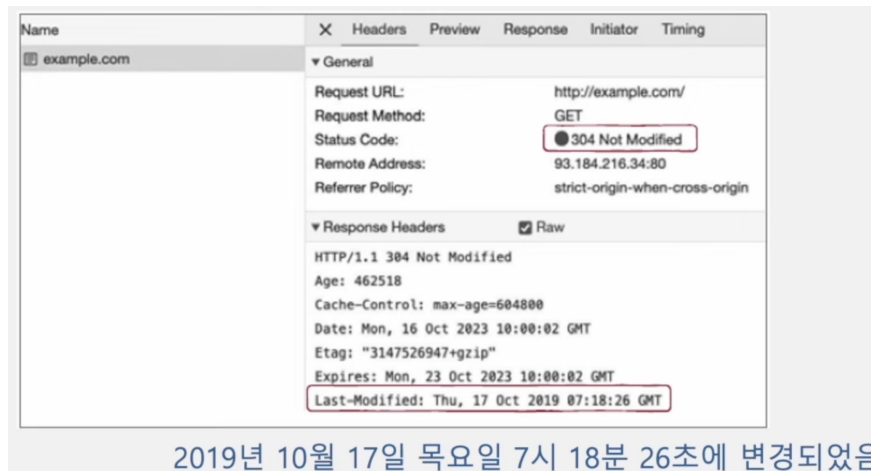
캐시

1. 서버가 요청받은 자원이 변경된 경우
 - 상태 코드 200(OK)과 함께 새로운 자원을 반환

캐시

2. 서버가 요청받은 자원이 변경되지 않은 경우

- 상태 코드 304(Not Modified) 반환
 - = 캐시된 자원을 참조하라
- Last-Modified 헤더로 자원의 '마지막 변경 시점'이 같이 알려줌.



캐시

3. 서버가 요청받은 자원이 삭제된 경우

- 상태 코드 404(Not Found) 반환

캐시

- 캐시 신선도 재검사

2. If-None-Match 헤더: 엔티티 태그(Etag)를 기반으로 서버에게 물어보는 방법

- Etag: '자원의 버전'을 식별하기 위한 정보



캐시

- 캐시 신선도 재검사


2. If-None-Match 헤더: 엔티티 태그(Etag)를 기반으로 서버에게 물어보는 방법

- 이 Etag 값과 일치하는 자원이 있는가?


```
GET /index.html HTTP/1.1  
Host: www.example.com  
If-None-Match: "abc"
```


쿠키

- HTTP는 상태를 유지하지 않는 스테이트리스한 프로토콜
 - 브라우저에서는 어떻게 로그인 상태를 유지하는걸까?

 ID/전화번호

[1] 일회용 번호

 QR코드

아이디 또는 전화번호

비밀번호

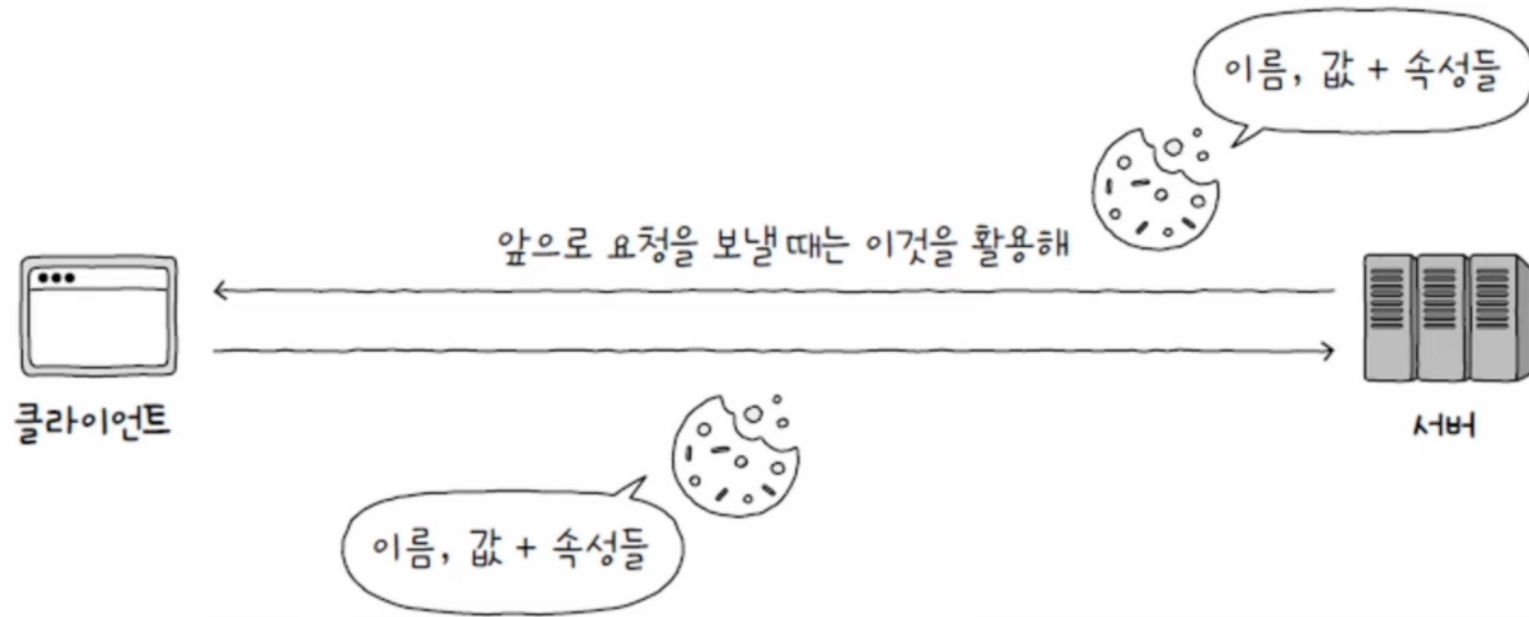
☒ 로그인 상태 유지

IP보안 ☐ OFF

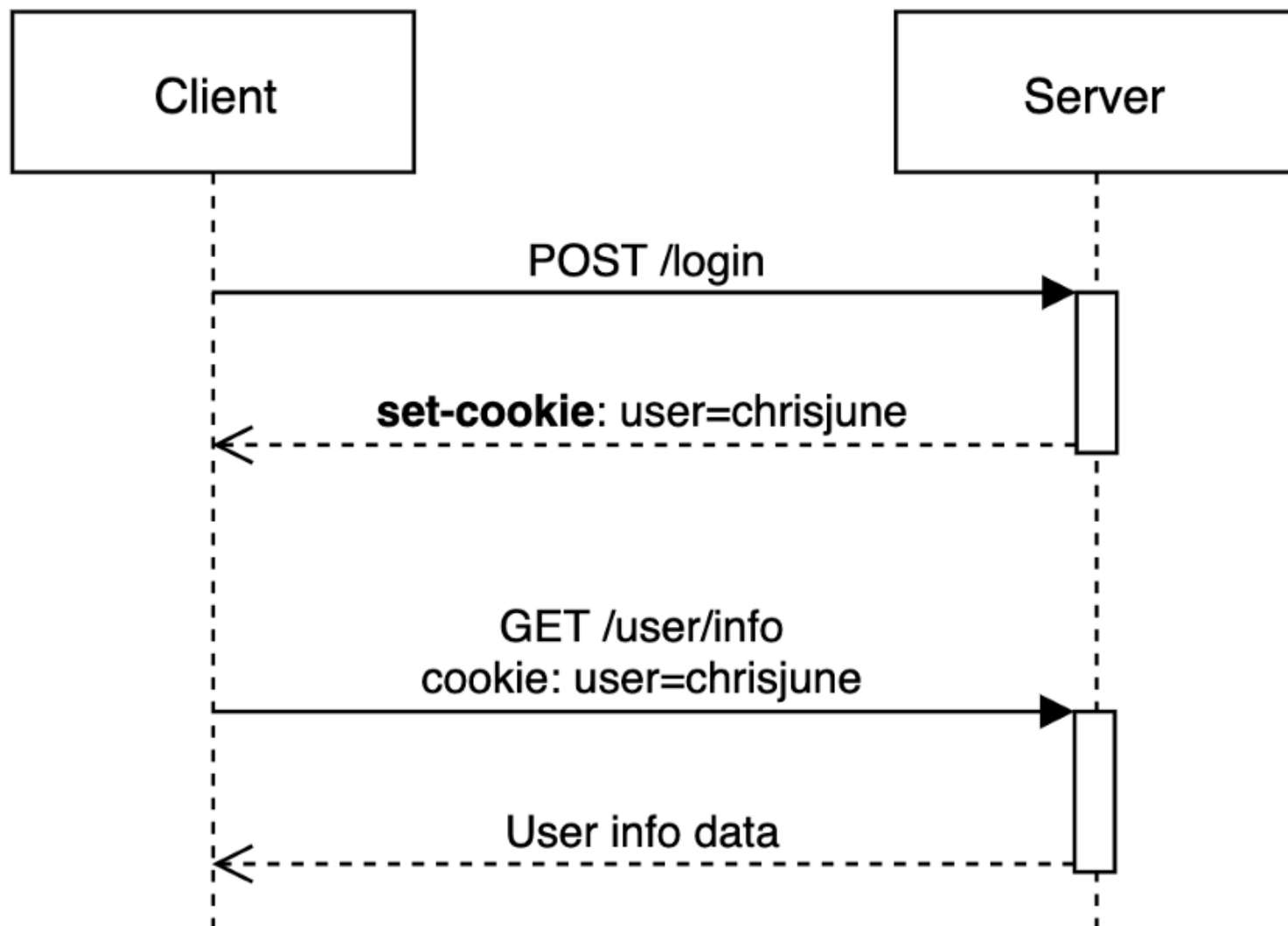


쿠키

- 서버에서 생성되어 클라이언트 측에 저장되는 데이터
 - 상태를 유지하지 않는 HTTP의 특성을 보완하기 위한 수단
 - 기본적으로 <이름, 값> 쌍의 형태 + @



쿠키



쿠키

- 여러 쿠키를 전달할 때는 세미콜론(;)으로 구분

요청 메시지

GET /next_page HTTP/1.1

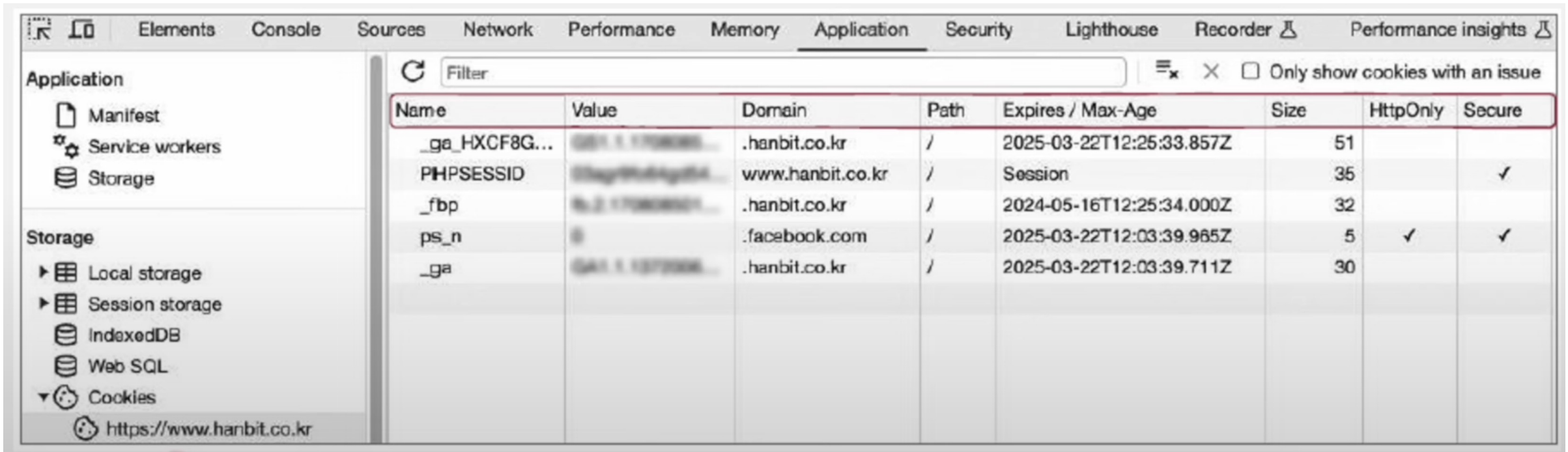
Host: example.com

Cookie: name=minchul; phone=100-100; message=Hello

...헤더 후략...

쿠키

- 브라우저에서 쿠키를 직접 확인 가능
 - 쿠키는 브라우저에서 저장되고 관리



The screenshot shows the 'Application' tab in a web browser's developer tools. The left sidebar is expanded to 'Storage' > 'Cookies'. The main pane displays a table of cookies for the current page (https://www.hanbit.co.kr). The table has columns for Name, Value, Domain, Path, Expires / Max-Age, Size, HttpOnly, and Secure. Five cookies are listed: _ga_HXCF8G..., PHPSESSID, _fbp, ps_n, and _ga. The _ga_HXCF8G... and _ga cookies are from .hanbit.co.kr, while _fbp is from .hanbit.co.kr and ps_n is from .facebook.com. The PHPSESSID cookie is a session cookie. The _ga cookie expires on 2025-03-22T12:03:39.711Z. The _fbp cookie expires on 2024-05-16T12:25:34.000Z. The ps_n cookie expires on 2025-03-22T12:03:39.965Z. The _ga_HXCF8G... cookie expires on 2025-03-22T12:25:33.857Z. The PHPSESSID cookie is a session cookie. The _ga cookie is marked as HttpOnly and Secure. The _fbp cookie is marked as HttpOnly and Secure. The ps_n cookie is marked as HttpOnly and Secure. The _ga_HXCF8G... cookie is marked as HttpOnly and Secure.

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure
_ga_HXCF8G...	GA1.1.17088...	.hanbit.co.kr	/	2025-03-22T12:25:33.857Z	51		
PHPSESSID	8b9d9d9d9d9d...	www.hanbit.co.kr	/	Session	35		✓
_fbp	fb.1.17088...	.hanbit.co.kr	/	2024-05-16T12:25:34.000Z	32		
ps_n	0	.facebook.com	/	2025-03-22T12:03:39.965Z	5	✓	✓
_ga	GA1.1.17088...	.hanbit.co.kr	/	2025-03-22T12:03:39.711Z	30		

쿠키

- 쿠키의 여러 속성
 - Domain: 사용 가능한 도메인
 - Path: 같은 도메인이더라도 경로별로 쿠키를 구분하여 사용할 때 사용
 - Expires/Max-age: 쿠키 유효 기간

쿠키

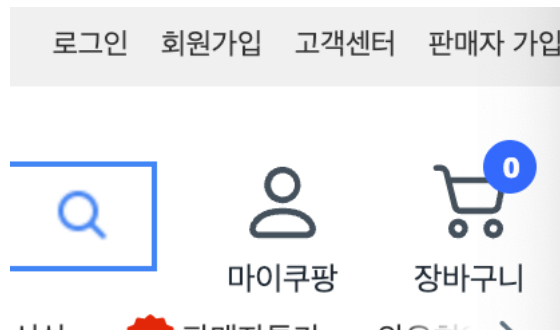
- 쿠키의 한계
 - 쿠키의 대표적인 한계는 보안
 - 쿠키 정보는 쉽게 노출되거나 조작될 수 있음
- Secure와 HttpOnly
 - Secure: HTTPS 프로토콜이 사용되는 경우에만 쿠키 전송
 - HttpOnly: HTTP 송수신을 통해서만 쿠키를 이용하도록 제한
 - 쿠키 관련 데이터는 자바스크립트 언어를 통해서도 접근 가능
 - HttpOnly는 자바스크립트에서 쿠키에 접근하지 못하도록 하는 속성

웹 스토리지

- 쿠키와 유사한 웹 스토리지(web storage)
 - 클라이언트가 저장하고 클라이언트의 상태를 추측할 수 있는 <키-값> 쌍 형태의 정보
 - 웹 스토리지는 웹 브라우저 내에 저장
 - 일반적으로 쿠키보다 더 큰 데이터 저장 가능
 - 쿠키는 서버로 자동 전송되지만, 웹 스토리지의 정보는 서버로 자동 전송X

웹 스토리지

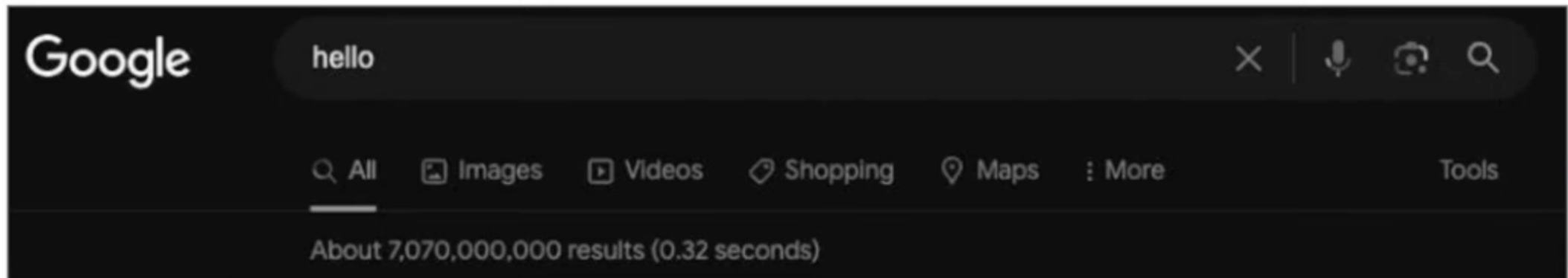
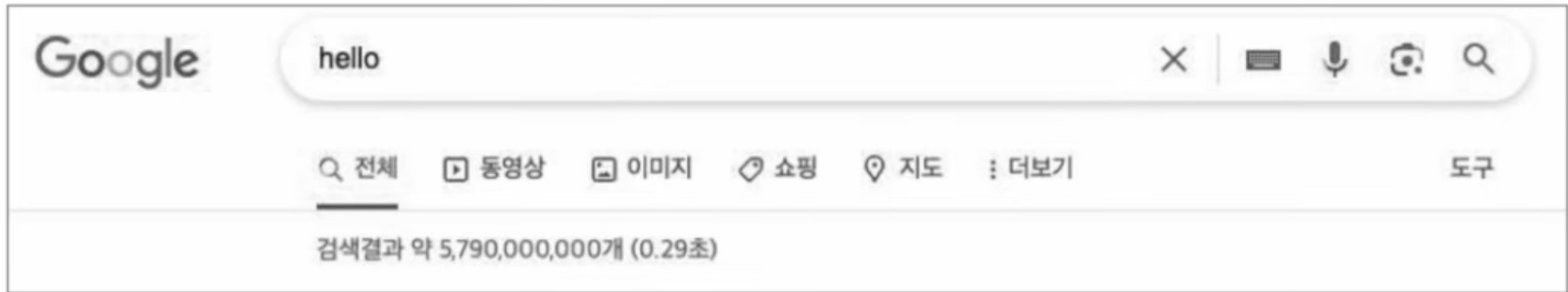
- 웹 스토리지 종류
 - 로컬 스토리지(local storage)
 - 별도로 삭제하지 않는 한 영구적으로 저장이 가능한 정보
 - 세션 스토리지(session storage)
 - 세션이 유지되는 동안(브라우저가 열려 있는 동안) 유지되는 정보



콘텐츠 협상과 표현

- 클라이언트는 서버에 자원을 요청하고, 서버는 요청받은 자원을 응답한다
- 엄밀히 따지면 서로 주고받는 것은 '자원의 표현'
 - 표현(representation): 송수신 가능한 자원의 형태
- 같은 자원에 대해서도 여러 가지 표현이 있을 수 있다.

콘텐츠 협상과 표현



콘텐츠 협상과 표현

- 콘텐츠 협상

- 같은 자원에 대해 할 수 있는 여러 표현 중 클라이언트가 가장 적합한 자원의 표현을 제공하는 기술
 - 영어로 요청하면 영어로 된 형태 제공
 - 한국어로 요청하면 한국어로 된 형태 제공

콘텐츠 협상과 표현

- 대표적인 콘텐츠 협상 헤더
 - Accept: 선호하는 미디어 타입을 나타내는 헤더
 - text/html, application/xml;q=0.9, text/plain;q=0.6, */*;q=0.5
 - Accept-Language: 선호하는 언어를 나타내는 헤더
 - ko-KR, ko;q=0.9, en-US;q=0.8, en;q=0.7
 - Accept-Encoding: 선호하는 인코딩 방식을 나타내는 헤더
 - br;q=1.0, gzip;q=0.8, */*;q=0.1