

# 1장, 2장

## 프로토콜 버전

### 1. HTTP 0.9 (1991년)

- GET 메서드만 지원, 상태 코드, 버전 번호, HTTP 헤더, MIME 타입 지원 안함

### 2. HTTP 1.0 (1996년)

- POST, HEAD 메서드 추가. 상태 코드, 버전 번호, HTTP 헤더, MIME 타입 지원
- If-modified-since를 이용해서 캐싱지원
- keep-alive 커넥션 지원

### 3. HTTP 1.1 (1997년)

- HTTP 설계의 구조적 결함 교정, 성능 최적화, 잘못된 기능 제거에 집중
- 복잡해진 웹 애플리케이션과 배포 지원
- 다중 요청/응답 불가능

HTML 문서 내의 여러가지 요청, CSS 파일 로드하는 Link태그, 이미지 파일을 로드하는 img 태그, 자바스크립트를 로드하는 script 태그 등은 개별적으로 전송되고 개별적으로 응답

이와 달리, HTTP 2.0은 다중 요청/응답 가능

- 추가적인 캐싱지원, entity tag(Etag): 두 리소스가 동일한지, cache-control: 캐싱 정책 정의
- if-unmodified-since, if-match, if-none-match 추가
- 모든 TCP 커넥션은 지속 커넥션

<https://stackoverflow.com/questions/246859/http-1-0-vs-1-1>

### 5. HTTP 2.0 (2015년)

- HTTP 1.1 성능 개선
- 구글의 SPDY 프로토콜 기반

## 웹 애플리케이션 종류

인터넷과 상호작용 할 수 있는 웹 애플리케이션은 많음.

### 1. 프록시

- 보안, 성능 최적화를 위해 사용 (바이러스 검사, 성인 콘텐츠 차단, 캐시 프록시)
- 클라이언트와 서버 사이에 위치하여, 모든 HTTP 요청을 받아 서버에 전달. 사용자 대신 서버에 접근

### 2. 캐시

- 웹 캐시, 웹 프록시는 자주 찾는 문서의 사본을 복사해 두는 HTTP 프록시 서버

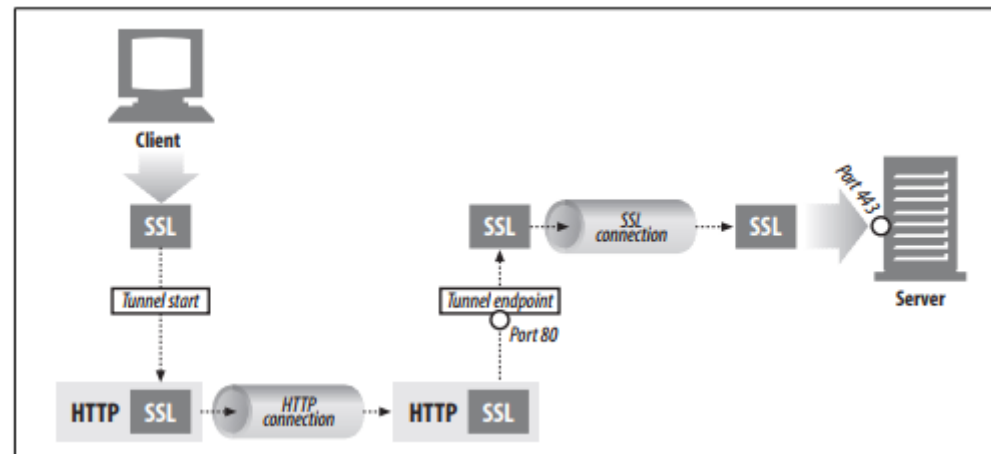
### 3. 게이트웨이

- 다른 서버들의 중개자로 동작
- HTTP 프로토콜 트래픽을 다른 프로토콜 트래픽으로 변환

- 진짜 서버인 것처럼 요청을 다루므로, 클라이언트는 게이트웨이와 통신하고 있음을 알아채지 못함

#### 4. 터널

- HTTP 프로토콜을 지원하지 않는 애플리케이션에 HTTP 애플리케이션을 사용해 접근하는 방법
- 터널을 사용하면 HTTP 커넥션을 통해서, HTTP가 아닌 트래픽 전송 가능



#### 5. 사용자 에이전트

- 사용자를 위해 HTTP 요청을 만들어주는 프로그램
- 대표적인 예로 브라우저 존재

### URI, URL, URN

- URL과 URN은 URI의 부분 집합
  - URL은 리소스의 구체적인 위치
  - URN은 위치에 영향받지 않는 유일무이한 이름
- URL은 HTTP 프로토콜 뿐만 아니라, 다른 프로토콜도 가능
  - mailto:president@whitehouse.gov
  - ftp://ftp.lots-o-books.com/pub/complete-price-list.xls
  - rtsp://www.joes-hardware-com:554/interview/cto\_video
- URL을 통해서 리소스에 쉽게 접근 가능
  - FTP 서버에 접근하려고 FTP 클라이언트 필요 없음
  - 메일 전송하려고 메일 프로그램 필요 없음

### URL 문법

- URL 문법은 스킴에 따라서 달라짐. 큰 차이는 없음
- 일반적으로 9개 부분으로 나뉨
  - <스킴>://<사용자 이름>:<비밀번호>@<호스트>:<포트>/<경로>;<파라미터>?<질의>#<프래그먼트>

#### 1. 호스트와 포트

- 호스트는 리소스를 가지고 있는 호스트 장비
- 포트는 서버가 열어놓은 네트워크 포트

#### 2. 사용자 이름과 비밀번호

- FTP 서버에서 요구

### 3. 경로

- 리소스의 위치
- 각 경로조각은 자체만의 파라미터 가짐

### 4. 파라미터

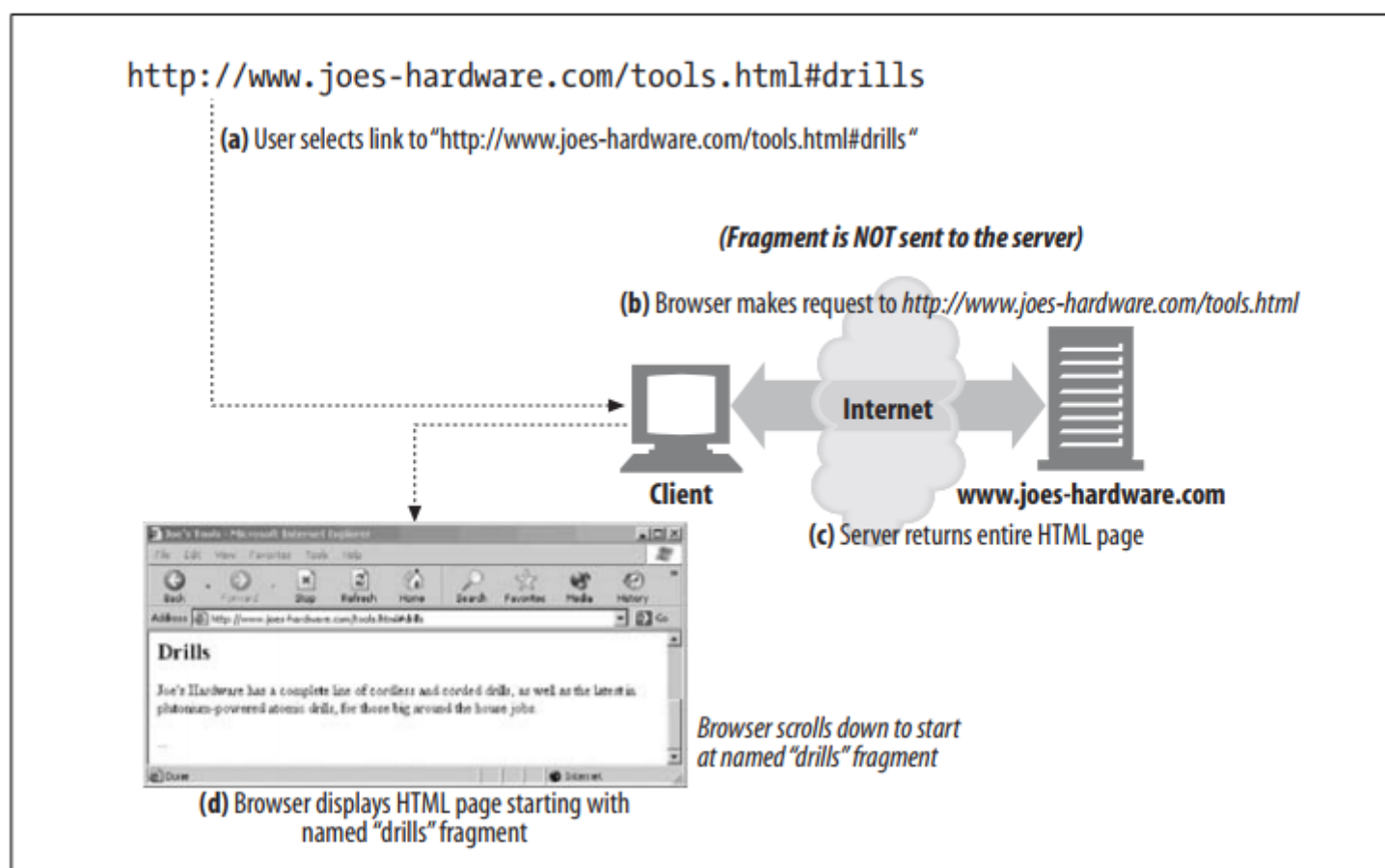
- 이름=값 쌍의 리스트

### 5. 질의

- 이름=값 쌍의 리스트
- &로 구분

### 6. 프래그먼트

- 리소스의 특정 부분 가리킴(텍스트 문서에서 특정 절)
- 서버는 객체 일부가 아닌 전체만 다룸. 클라이언트는 서버에 프래그먼트를 전달하지 않음.



<https://velog.io/@wka01/프론트엔드-면접-문제-은행-HTML-질문-답변#자바스크립트-동작-원리이벤트-루프>

## 상대 URL

- 상대 URL은 `./hammers.html`
- 기저 URL은 `http://www.joes-hardware.com/tools.html`
- 절대 URL은 `http://www.joes-hardware.com/hammers.html`
- 위치를 변경하더라도 새로운 기저 URL에 의해 해석

## URL 확장

- 브라우저는 URL 자동 확장

### 1. 호스트명 확장

- yahoo를 입력하면 `www.yahoo.com`으로 확장

## 2. 히스토리 확장

- 사용자가 방문했던 URL 저장
- http://www.joes- 입력하면 http://www.joes-hardware.com 으로 확장

## URL에서 안전하지 않은 문자

- URL은 어떤 인터넷 프로토콜에서든 안전하게(정보가 유실될 위험 없이) 전송 & 가독성 되어야(있어야) 함.  
SMTP는 특정 문자를 제거할 수도 있는 방식 사용. 그러므로 URL은 상대적으로 작고, 일반적으로 안전한 문자만(ASCII 문자 집합 일부) 허용.  
이외의 문자를 포함하는 경우 이스케이프(안전한 문자로 바꿈, 인코딩) 처리
- 인코딩은 % 뒤에 16진수 두 자리. 공백의 경우, %20