

HTTP는 HTML 문서와 같은 리소스들을 가져올 수 있도록 해주는 프로토콜이다. 클라이언트와 서버가 문서를 주고 받는 클라이언트-서버 프로토콜이기도 하다. 여기서 클라이언트는 사용자이며, 사용자는 브라우저에게 클라이언트의 역할을 수행하도록 한다. 클라이언트에서 request를 HTTP를 통해 서버로 보내면, 서버에서 데이터를 처리하여 요청에 대한 응답 즉 response를 클라이언트로 다시 보낸다. 실제로는 클라이언트와 서버 간에 프록시와 같은 더 많은 컴퓨터들이 존재하여, 계층화된 설계에 따라 요청과 응답을 서로 주고받는다. 클라이언트 역할을 하는 브라우저는 서버에 HTML문서를 요청하고, 레이아웃 정보(CSS)와 이미지, 비디오와 같은 리소스들도 가져와 이를 혼합하여 웹페이지를 구성하고 사용자에게 보여준다. 그리고 HTTP가 이 과정에서 어떤 지시사항인지, 어떤 응답인지를 해석하는 역할을 한다. HTTP/1은 사람이 읽을 수 있게 고안되어 쉽게 접할 수 있으며, HTTP/2는 메시지를 이진 구조인 프레임 안으로 임베드 시켜 사람이 읽기 보다는 헤더 압축, 다중화 등의 최적화를 가능하게 한 버전이다. 이는 파이프라이닝에 비해 여러 요청을 처리하기 용이하면서도 구현하기 쉽다. HTTP/1은 TCP연결을 각 요청마다 열기 때문에 요청이 많으면 여러 TCP 연결이 있어야 해서 단일 TCP 연결을 공유하는 것에 비해 비효율적이다. 하지만 HTTP/2는 단일 TCP 연결에서 메시지를 다중 전송하는 multiplex에서 더 발전되어 있다. HTTP는 클라이언트와 서버의 합의를 통해 새로운 헤더를 만들 수 있어 확장성이 뛰어나다. HTTP는 상태를 저장하지 않는 프로토콜이기 때문에 쇼핑 바구니에 새 물건을 담아도 이전에 담았던 물건을 유지하는 것이 힘들다. 이를 데이터베이스에 항상 저장하고 불러오지 않고 쉽게 상태를 가지게 할 수 있는 것이 HTTP 쿠키이며, 사용자의 컴퓨터에 암호화된 컨텍스트를 저장하여 데이터베이스와의 지속적인 소통 없이도 상태를 가질 수 있게 해준다. HTTP가 동작하는 방식은 먼저 TCP 연결을 열어 메시지를 주고 받을 통로를 만든 뒤 HTTP 메시지를 서버에 전송하고, 서버에서 다시 전송된 응답을 읽어들이고 연결을 닫거나 TCP 연결을 재사용한다. HTTP 요청의 요소는 GET, POST와 같은 CRUD 메소드, 또는 OPTIONS, HEAD와 같은 추가적인 메소드, 리소스의 경로, HTTP 버전, 추가 정보 헤더, 그리고 body가 있고, 응답의 요소는 HTTP 버전, 상태 코드, 상태 메시지, 추가 정보 헤더, 그리고 body가 있다.