

## HTTP 통신이란?

① HTTP란 HyperText Transfer Protocol의 약자로 HTML 파일을 전송하는 프로토콜이라는 의미를 가진다. 웹브라우저에서 통신이 일어나며, 초기에는 HTML 파일을 전송하려는 목적으로 만들어졌으나 현재는 JSON, Image 파일 등 또한 전송한다.

### HTTP 통신의 통신 방식

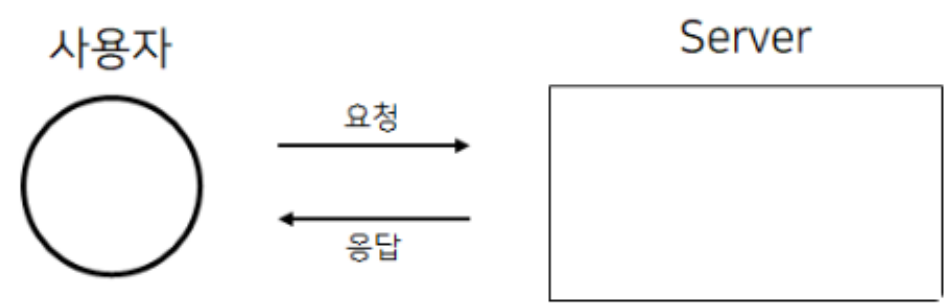


그림1. HTTP 통신

HTTP 통신은 클라이언트에서 서버로 요청을 보내고 서버가 응답하는 방식으로 통신이 이루어진다. 응답에는 클라이언트의 요청에 따른 결과를 반환한다. 이로 인해 다음과 같은 특성을 가진다.

★ 클라이언트의 요청이 있을 때 서버가 응답하는 방식. 단방향 통신.

서버의 응답에는 응답 코드가 같이 전송되며, 사용자는 응답 코드와 메시지 응답으로부터 오는 메시지 바디를 통해 요청 값을 전달 받는다.

초기에는 서버는 응답한 후 클라이언트(사용자)의 Connection을 곧바로 끊어버렸으나, 최근에는 성능상의 이유(Connection을 맺고 끊는 비용이 비싸다)로 Keep Alive 옵션을 통해 일정 기간 동안 클라이언트와 Connection을 유지하는 방식으로 통신이 가능해졌다.

### 정리

- HTTP 통신은 클라이언트의 요청이 있을 때 서버가 응답하는 단방향 통신이다.
- JSON, Image, HTML 파일 등 다양한 파일을 전송 받을 수 있다.
- 응답을 받은 후 Connection이 끊어지는 것이 기본 동작이지만, 성능 상으로 필요하다면 Keep Alive 옵션을 주어 일정 시간 동안 Connection을 유지하는 것이 가능하다.

# 소켓 통신이란?

소켓이란 <sup>있음</sup> 두 프로그램이 서로 데이터를 주고 받을 수 양쪽(두 프로그램 모두)에 생성되는 통신 단자이다.



그림2. 소켓 통신

소켓 통신이란 서버와 클라이언트 양방향 연결이 이루어지는 통신으로, 클라이언트도 서버로 요청을 보낼 수 있고 서버도 클라이언트로 요청을 보낼 수 있는 통신으로 다음과 같은 특성을 가진다.

★ 클라이언트와 서버 양쪽에서 서로에게 데이터 전달을 하는 방식의 양방향 통신.

보통 스트리밍이나 실시간 채팅 등 실시간으로 데이터를 주고 받아야 하는 경우(Connection을 자주 맺고 끊는 HTTP 통신보다)소켓 통신이 적합하다. 소켓 통신은 계속해서 Connection을 들고 있기 때문에 HTTP 통신에 비해 많은 리소스가 소모된다.

## 정리

- 자주 데이터를 주고 받는 환경이 아닌 경우 HTTP 통신을 통해 받는 것이 유리하다.
- 자주 데이터를 주고 받아야 하는 환경에서는 소켓 통신이 유리하다.
- ★ HTTP 통신은 사용자가 서버에 요청을 보내는 단방향 통신인 반면, 소켓 통신은 양방향 통신이다.