

유노코딩과 함께하는 자바스크립트 기초 강의

비동기 데이터 통신





HTTP

HTTP(HyperText Transfer Protocol)는 웹 브라우저와 웹 서버가 데이터를 주고받을 수 있게 디자인된 통신 규약이다.

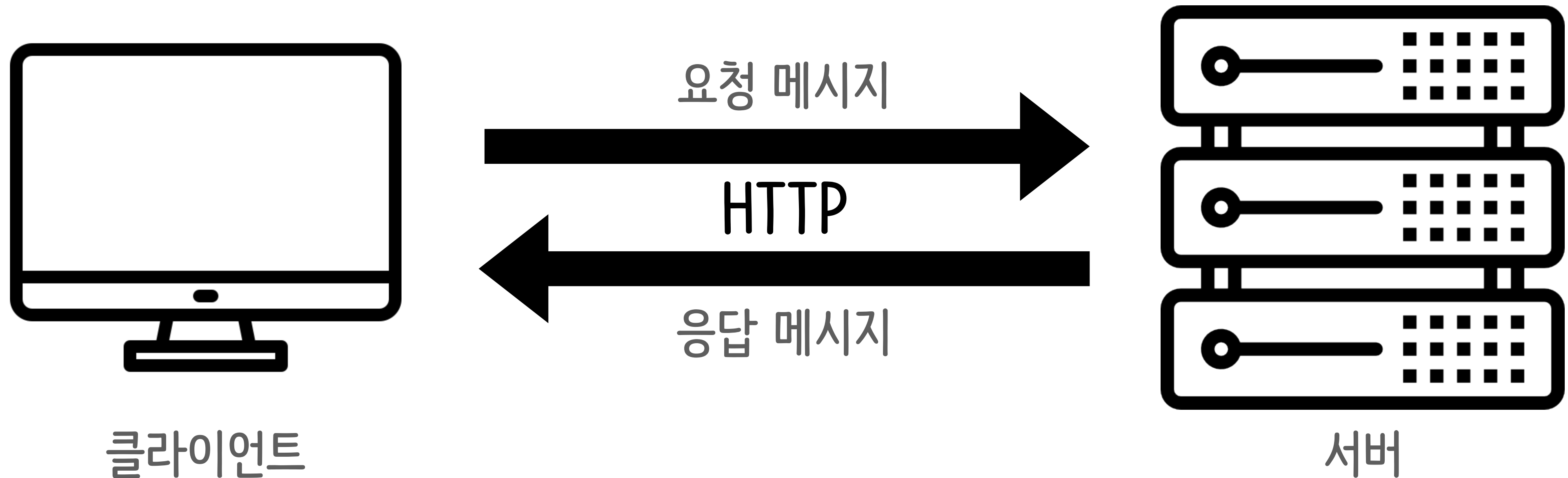
HTTP는 웹에서 이루어지는 모든 데이터 교환의 기초이며, 이 규약을 암호화하여 보안을 확보한 규약은 HTTPS라 한다.

=> 데이터를 주고 받는다? HTML, 이미지, 영상, 음성 파일 등을 주고 받는다!



HTTP 통신

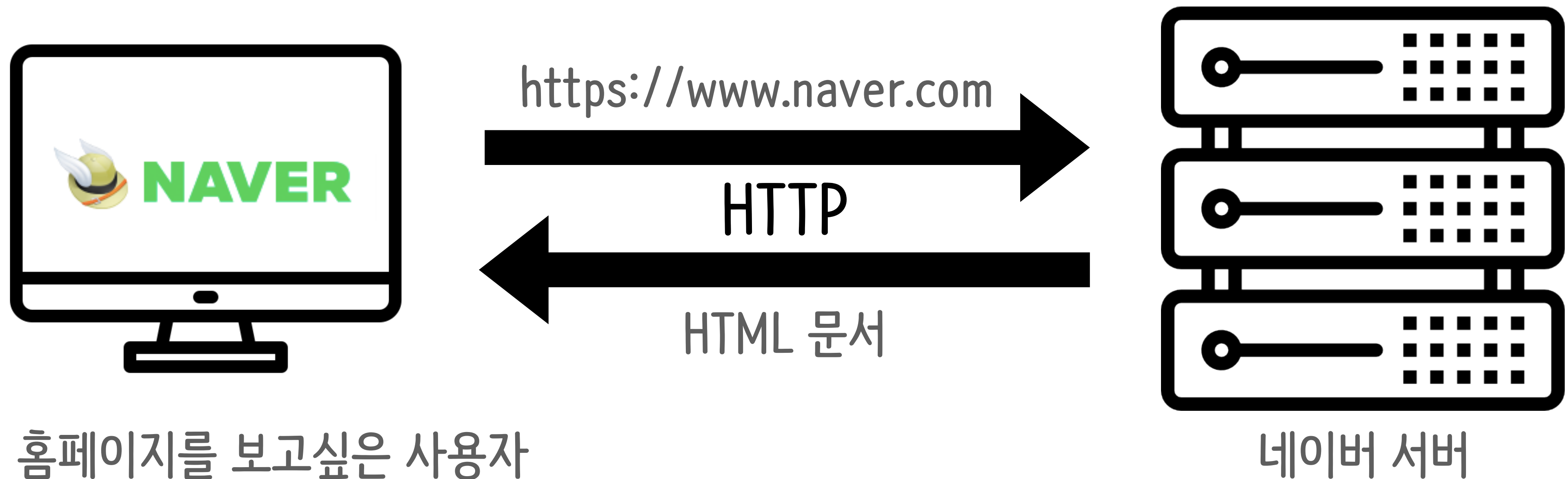
HTTP에서는 클라이언트(사용자)가 서버(호스트)에 요청 메시지를 보내고 이에 대해 서버가 응답 메시지를 반환한다. 서버는 응답 메시지를 반환한 후에 초기 상태로 돌아간다. 이러한 특징으로 인해 HTTP를 stateless 규약이라고도 한다.





생활 속 HTTP 통신

HTTP에서는 클라이언트(사용자)가 서버(호스트)에 요청 메시지를 보내고 이에 대해 서버가 응답 메시지를 반환한다. 서버는 응답 메시지를 반환한 후에 초기 상태로 돌아간다. 이러한 특징으로 인해 HTTP를 stateless 규약이라고도 한다.





HTTP 요청과 응답 - 요청

HTTP 요청 메시지는 요청 행, 요청 헤더, 메시지 본문으로 구성된다.

- 요청 행: 요청 메소드, URL, HTTP 버전 정보 등을 포함한다.
- 요청 헤더: 메시지 정보, 메시지에 저장한 데이터 정보 등을 포함한다.
- 메시지 본문: 서버 측으로 보내는 데이터를 포함한다.

=> 요청 메소드란? GET, POST 등 데이터 송수신 방식을 지정한 것



HTTP 요청과 응답 - 응답

HTTP 응답 메시지는 응답 행, 응답 헤더, 메시지 본문으로 구성된다.

- 응답 행: 상태 코드, 보충 메시지, HTTP 버전 정보 등을 포함한다.
- 응답 헤더: 메시지 정보, 메시지에 저장한 데이터 정보 등을 포함한다.
- 메시지 본문: 클라이언트 측으로 보내는 데이터를 포함한다.

=> 상태 코드란? 요청이 성공적으로 완료되었는지를 보여주는 숫자 코드



주요 상태 코드

상태 코드는 요청이 성공적으로 완료되었는지를 보여주는 숫자 코드이며, 각 숫자마다 의미가 정해져 있다.

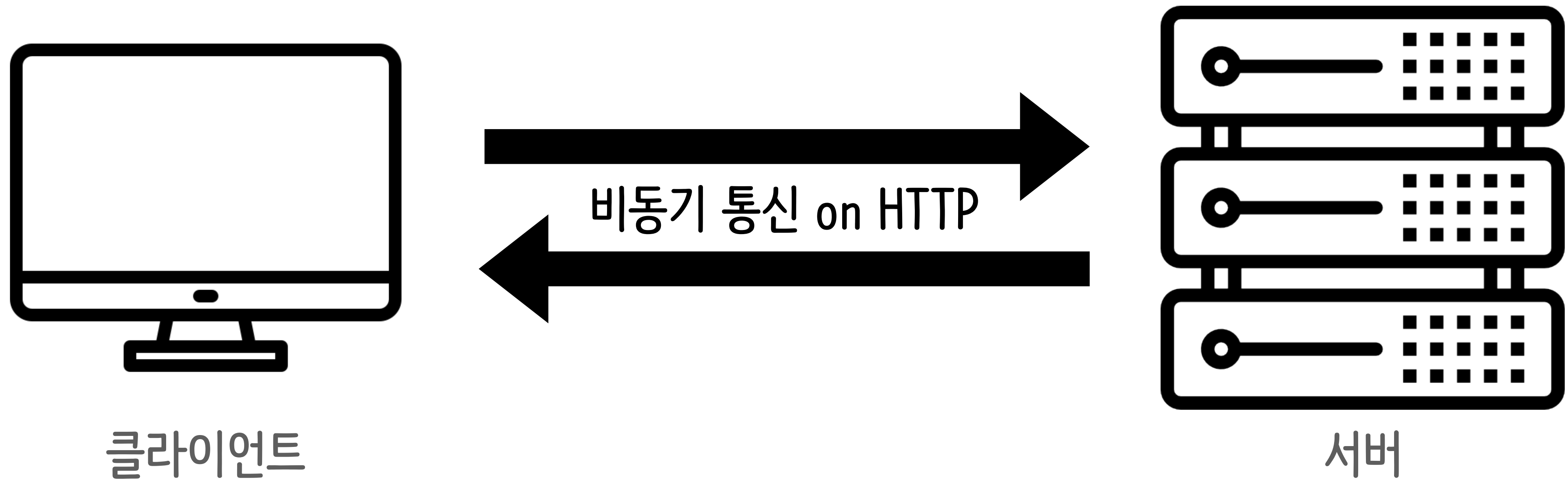
상태 코드	상태 이름	내용
200	OK	요청 성공
401	unauthorized	인증 실패
403	Forbidden	액세스 허용되지 않음
404	Not Found	요청한 리소스를 찾지 못함
500	Internal Server Error	서버 내부 오류 발생
503	Server Unanailable	서비스를 일시적 사용 불가



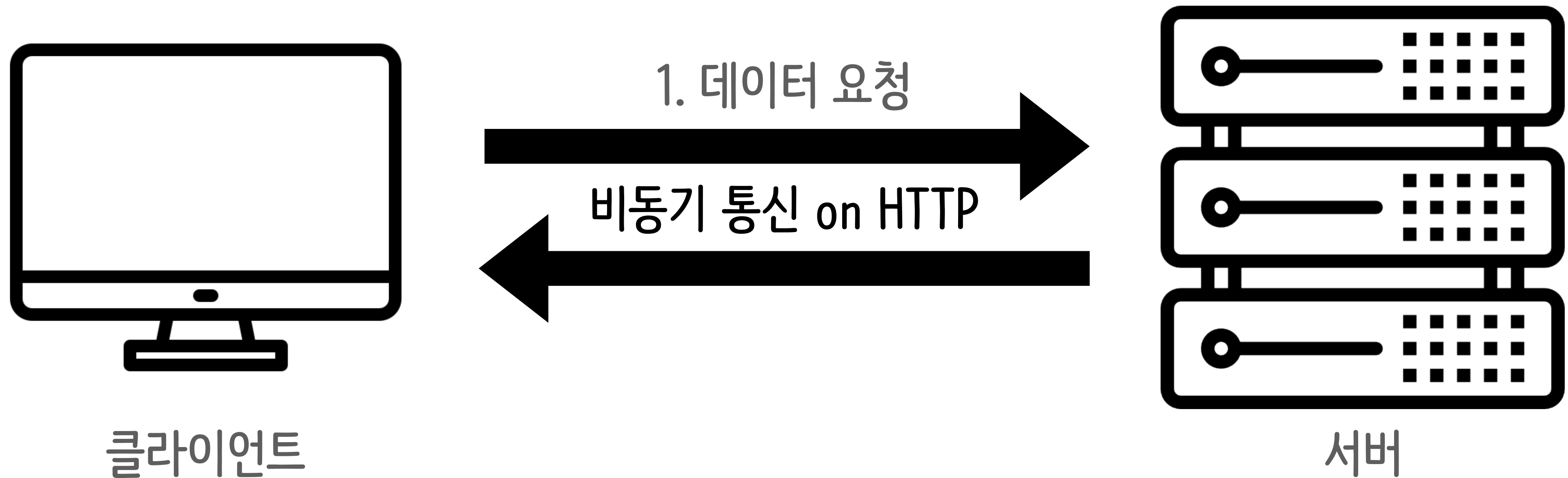
XMLHttpRequest

XMLHttpRequest는 서버와 상호작용하기 위해 사용하는 자바스크립트 빌트인 생성자이다. 이를 이용하면 서버(URL)로부터 데이터를 받아와 전체 페이지의 새로고침 없이도 페이지의 일부만 업데이트하는 비동기 방식의 데이터 통신을 수행할 수 있다.

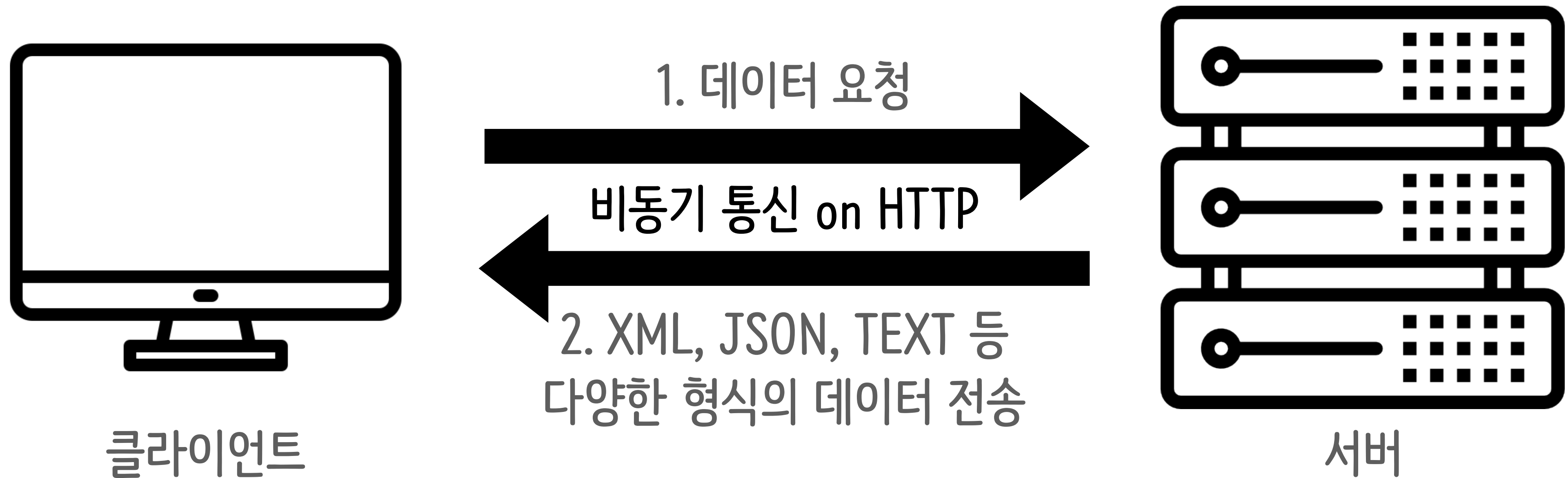
이러한 프로그래밍 기법을 Ajax라 한다! Ajax(Asynchronous JavaScript And XML)는 웹 서버와 비동기로 통신하고 DOM을 이용하여 웹 페이지를 동적으로 갱신하는 프로그래밍 기법을 말한다.



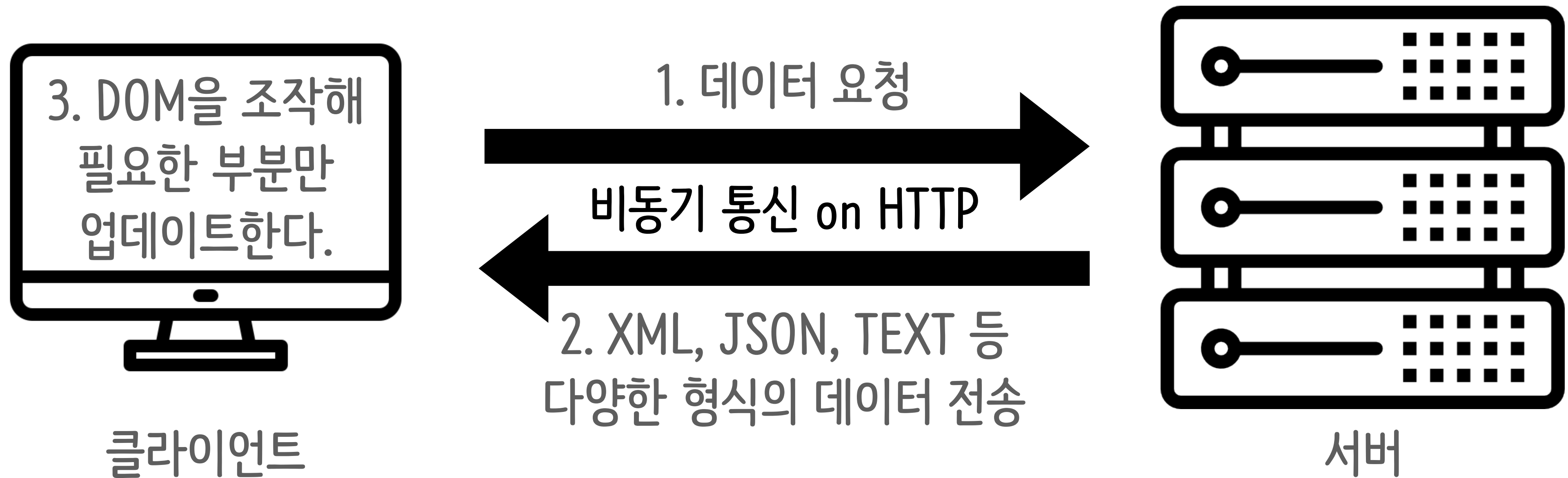
Ajax 는 비동기 프로그래밍 방식이다



Ajax 는 비동기 프로그래밍 방식이다



Ajax 는 비동기 프로그래밍 방식이다



Ajax 는 비동기 프로그래밍 방식이다



XMLHttpRequest 사용 흐름

Ajax 기법에 기초한 XMLHttpRequest 비동기 통신은 다음 세 단계를 거친다.

- XMLHttpRequest 객체를 생성한다!
- 서버와 통신할 때 필요한 정보 및 처리 방법 등을 기입한다!
- 요청을 전송해 통신을 시작한다!

=> 객체 생성은 new 연산, 아시죠?



이벤트 핸들링

비동기 통신을 할 때는 서버와의 통신 상태를 감시할 수 있다. 그리고 이때 발생하는 이벤트에 대한 이벤트 핸들러를 등록할 수 있다. 아래는 XMLHttpRequest 객체의 주요 이벤트 몇 가지를 정리한 표이다.

이벤트명	발생하는 시기
abort	요청이 취소되었을 때
error	요청이 실패했을 때
load	요청을 성공하여 응답을 가져올 수 있을 때
progress	데이터를 주고받는 중
timeout	요청 시간을 초과했을 때



요청 초기화 및 전송

서버에 요청을 보낼 때는 XMLHttpRequest 객체의 open 메소드로 요청을 초기화하고, 이후 send 메소드로 보내주는 순서로 작업을 수행한다.

```
// open 메소드 사용시 기본 형태
const request = new XMLHttpRequest()
request.open("HTTP메소드", "서버URL")

// 초기화는 했고, 이제 요청 보내자!
request.send()
```



수고하셨습니다 :))