

HTTP(**HTTP**, **Hypertext Transfer Protocol**)는 클라이언트와 서버 간에 데이터를 주고받기 위한 프로토콜입니다. 웹에서 가장 일반적으로 사용되는 프로토콜 중 하나이며, **REST API**와 함께 많이 사용됩니다.

HTTP는 클라이언트가 서버에 요청(**Request**)을 보내고, 서버는 클라이언트에게 응답(**Response**)을 반환하는 방식으로 동작합니다. 이때, 클라이언트와 서버 간의 통신은 요청과 응답 메시지로 이루어집니다.

HTTP 요청은 다음과 같은 구성 요소로 이루어집니다:

1. **HTTP 메서드**: 클라이언트가 서버에게 수행하고자 하는 동작을 나타냅니다. 가장 흔히 사용되는 메서드는 **GET**, **POST**, **PUT**, **DELETE** 등이 있습니다. **GET**은 리소스의 조회, **POST**는 리소스의 생성, **PUT**은 리소스의 갱신, **DELETE**는 리소스의 삭제를 의미합니다.
2. **URI(Uniform Resource Identifier)**: 서버의 자원을 식별하는 고유한 경로를 나타냅니다. 예를 들어, **http://example.com/users**는 서버의 **users** 자원을 가리킵니다.
3. **HTTP 헤더(Header)**: 요청이나 응답에 대한 부가적인 정보를 담고 있습니다. 예를 들어, 요청의 인증 정보, 캐시 설정, 데이터 타입 등을 헤더에 포함할 수 있습니다.
4. **HTTP 본문(Body)**: 요청이나 응답에 대한 추가적인 데이터를 담고 있습니다. 주로 **POST**나 **PUT** 메서드에서 사용되며, **JSON**이나 **XML** 형식으로 데이터를 전송할 수 있습니다.

HTTP 응답은 다음과 같은 구성 요소로 이루어집니다:

1. **상태 코드(Status Code)**: 서버가 요청을 처리한 결과를 나타내는 세 자리 숫자로 이루어진 코드입니다. 예를 들어, **200**은 성공, **404**는 찾을 수 없음, **500**은 서버 오류를 의미합니다.
2. **HTTP 헤더**: 응답에 대한 부가적인 정보를 담고 있습니다. 예를 들어, 응답의 데이터 타입, 캐시 설정, 인증 정보 등을 헤더에 포함할 수 있습니다.
3. **HTTP 본문**: 서버가 클라이언트에게 반환하는 데이터를 담고 있습니다. 주로 **JSON**이나 **XML** 형식으로 데이터를 전송합니다.

브라우저에서 **URL**을 입력하고 요청하여 서버에서 응답하는 과정.

1. 사용자가 웹 브라우저의 주소창에 **URL**을 입력합니다. 예를 들어, "**http://example.com**"과 같이 입력합니다.

2. 브라우저는 입력된 **URL**을 해석하고, 요청을 보내기 위해 **HTTP** 요청 메시지를 생성합니다. 이때, **HTTP** 요청 메시지에는 다음과 같은 정보가 포함됩니다:

- **HTTP** 메서드: 일반적으로 **GET** 메서드가 사용되며, 서버로부터 리소스를 요청합니다.

- **URI**: 입력된 **URL**에 포함된 경로를 **URI**로 설정합니다.

- **HTTP** 헤더: 요청에 대한 부가적인 정보를 포함합니다. 예를 들어, 사용자 에이전트(**User Agent**) 정보, 쿠키, 인증 정보 등이 포함될 수 있습니다.

3. 브라우저는 생성된 **HTTP** 요청 메시지를 서버로 전송합니다. 이때, **HTTP** 요청은 **TCP/IP** 프로토콜을 사용하여 인터넷을 통해 서버로 전달됩니다.

4. 서버는 받은 요청을 처리하고, 해당 리소스에 대한 응답을 생성합니다. 이때, 응답에는 다음과 같은 정보가 포함됩니다:

- 상태 코드: 요청의 처리 결과를 나타내는 세 자리 숫자로 이루어진 코드입니다. 예를 들어, **200**은 성공, **404**는 찾을 수 없음, **500**은 서버 오류를 의미합니다.

- **HTTP** 헤더: 응답에 대한 부가적인 정보를 포함합니다. 예를 들어, 응답의 데이터 타입, 캐시 설정, 인증 정보 등이 포함될 수 있습니다.

- **HTTP** 본문: 서버가 클라이언트로 반환하는 데이터가 포함됩니다. 예를 들어, **HTML**, **JSON**, 이미지 등의 데이터가 포함될 수 있습니다.

5. 서버는 생성된 **HTTP** 응답 메시지를 **TCP/IP** 프로토콜을 사용하여 브라우저로 전송합니다.

6. 브라우저는 받은 **HTTP** 응답 메시지를 해석하고, 화면에 적절한 방식으로 결과를 표시합니다. 예를 들어, **HTML** 응답의 경우 브라우저는 **HTML**을 렌더링하여 웹 페이지를 표시합니다. 다른 형식의 응답에는 브라우저가 해당 형식을 해석하여 표시합니다.