# HTML learning roadmap

Perfect 🚀 Let's move to the **next stage after TCP connection**.

Now that the **browser ↔ server** communication channel is established, the **actual request** begins.
This step is called **HTTP Request / Response Cycle**.

---

## 📒 HTTP Request and Response

### 🔹 What is HTTP?

- **HTTP (HyperText Transfer Protocol)** = the application layer protocol used by browsers and servers to exchange data.
- Works **on top of TCP** (already established via 3-way handshake).
- It is **stateless** → every request is independent (though cookies/sessions help track state).

---

### 🔹 HTTP Request (Client → Server)

When you type `http://example.com/` and hit Enter, the browser sends a request like:

```
pgsql

GET / HTTP/1.1
Host: example.com
User-Agent: Chrome/119.0
Accept: text/html,application/xhtml+xml
Accept-Language: en-US,en;q=0.9
Connection: keep-alive
```

**Breakdown of parts:**

1. **Request Line** → `GET / HTTP/1.1`
   - `GET` = method (others: POST, PUT, DELETE, etc.).
   - `/` = resource path.
   - `HTTP/1.1` = protocol + version.
2. **Headers** (metadata about the request)

- `Host:` → the domain requested.
- `User-Agent:` → browser + OS details.
- `Accept:` → formats browser accepts (HTML, JSON, images…).
- `Connection: keep-alive` → reuse same TCP connection for multiple requests.

3. **Body** (optional)
   - Present in methods like `POST`, `PUT` (e.g., form data, JSON payload).

---

### 🔹 HTTP Response (Server → Client)

Server replies with something like:

```
php-template

HTTP/1.1 200 OK
Date: Sat, 22 Aug 2025 09:45:00 GMT
Server: Apache/2.4.41
Content-Type: text/html; charset=UTF-8
Content-Length: 1256
Connection: keep-alive

<html>
  <head><title>Example</title></head>
  <body><h1>Hello, World!</h1></body>
</html>
```

**Breakdown of parts:**

1. **Status Line** → `HTTP/1.1 200 OK`
   - `200` = status code (✅ success).
   - Other codes:
     - `404 Not Found`
     - `500 Internal Server Error`
     - `301 Moved Permanently`
2. **Headers** (metadata about response)
   - `Content-Type: text/html` → tells browser type of data.
   - `Content-Length` → number of bytes in response body.
   - `Set-Cookie` → used for login sessions, personalization.
3. **Body** → actual content (HTML, JSON, image, etc.).

### 🔹 Important HTTP Concepts

- **Statelessness:**
  Each request is independent; server does not remember past requests.
  → Cookies, sessions, and tokens are used to maintain state.
- **Persistent Connections (keep-alive):**
  Instead of opening a new TCP connection for every resource, one connection is reused.
  → Faster page loads, especially for multiple assets (CSS, JS, images).
- **HTTP Versions:**
  - **HTTP/1.1** → default, allows persistent connections.
  - **HTTP/2** → multiplexing (multiple requests in same connection).
  - **HTTP/3** → built on QUIC (uses UDP for faster, secure transfer).

---

📌 At this stage:

- Browser has sent an HTTP request.
- Server replied with an HTTP response.
- Response body contains the **HTML document**.

---

👉 Next step will be: **Browser Rendering Pipeline** (how HTML → CSS → JS → final webpage is displayed).

Do you want me to **wrap up HTTP request/response into notes** like I did for DNS & TCP before we move to rendering, or directly continue to rendering?