Given: 10 objects total, each **10 MB**. Server transmission rate **100 Mbps**. "Let **25 ms** denote the time required to send a TCP request from PC to server." Non-persistent HTTP (a new TCP connection per object).

Assumption (explicit): The problem's "25 ms to send a TCP request from PC to server" is a one-way latency. So the round-trip time (RTT) = 2 × 25 ms = **50 ms**.

Also using the common simplified model for non-persistent HTTP: for each object we need about **2 RTTs** (one RTT for TCP handshake and one RTT for request/first byte of response) + the transmission time to send the object. I will use that model.

Step 1 — compute RTT

one-way latency = 25 ms →
RTT = 2 × 25 ms = **50 ms**.

Step 2 — transmission time for one object

Object size = 10 MB.
Convert to megabits: 10 MB = 10 × 8 = **80 Mbits**.
Server bandwidth = 100 Mbps, so transmission time = size / bandwidth = 80 Mbits ÷ 100 Mbps = 0.8 s = **800 ms**.

(Checked digits: 80 ÷ 100 = 0.8 → 0.8 s → 800 ms.)

Step 3 — time to get one object using non-persistent HTTP

Using model: time per object ≈ 2 × RTT + transmission time
= 2 × 50 ms + 800 ms = 100 ms + 800 ms = **900 ms**.

Step 4 — total for 10 objects

10 objects × 900 ms per object = 9,000 ms = **9.0 seconds**.

Final answer

The web browser will take **9,000 ms (9.0 s)** to load the page under the assumptions above.

Case A — Persistent HTTP (no pipelining, requests sent one-by-one on same TCP connection)

Model: one TCP connection established once; TCP handshake costs 1 RTT. For the first object you need the handshake + request/response latency (so ≈ 2 × RTT) + transmission. Each *additional* object requires 1 RTT + transmission (because connection exists, but you wait for request/response).

1. First object: `2 × RTT + transmission = 2×50 ms + 800 ms = 100 ms + 800 ms = 900 ms`.
2. Each of the remaining 9 objects: `1 × RTT + transmission = 50 ms + 800 ms = 850 ms`.
3. Total = `900 ms + 9 × 850 ms = 900 + 7,650 = 8,550 ms = **8.55 s**`.

Result (persistent, non-pipelined): 8,550 ms (8.55 s).

Case B — Persistent HTTP with pipelining (server can start sending responses after one request round; requests are queued so fewer RTTs)

If you can pipeline the requests (send multiple HTTP requests without waiting for each response), a common textbook model is:

- 1 RTT for TCP handshake + 1 RTT to send requests and get first byte of response = `2 × RTT = 100 ms`, then server transmits all objects. Transmission of objects is still limited by bandwidth and is effectively serial total transmission = `10 × 800 ms = 8,000 ms`. So total ≈ `100 ms + 8,000 ms = 8,100 ms = **8.1 s**.

Result (persistent + pipelining): 8,100 ms (8.1 s).

Short summary

- Non-persistent (earlier): **9,000 ms (9.0 s)**.
- Persistent, non-pipelined: **8,550 ms (8.55 s)**.
- Persistent with pipelining: **8,100 ms (8.1 s)**.