

Lesson 02 — What the Internet Actually Is

DNS, IP, HTTPS, browsers, CDNs — what happens between pressing Enter and the page appearing.

IT Landscape

Module 1

The question

*You type **wikipedia.org** and hit Enter.
What happens?*

We'll answer it in 11 slides.

Step 1: Your computer needs an address

- `wikipedia.org` is a **name**.
- Computers talk in **numbers** — IP addresses like `208.80.154.224`.
- Translating names ↔ numbers is what **DNS** does.

DNS = the phonebook of the internet.

DNS in one diagram

Browser → **Recursive resolver** (e.g. 8.8.8.8, 1.1.1.1)

↓

Root → **.org authoritative** → **wikipedia.org authoritative**

↓

IP returned (208.80.154.224)

Each step is cached for seconds-to-days — usually you only hit the resolver.

Step 2: Open a connection

- Your computer opens a **TCP connection** to that IP.
- Port **443** — the standard port for HTTPS.

TCP = the dial tone of the internet. It's the reliable, in-order byte stream every web app rides on top of.

Step 3: Wrap it in TLS

- **HTTPS = HTTP + encryption.**
- The browser asks for a **certificate**.
- The certificate must be signed by a **Certificate Authority** the browser trusts (Let's Encrypt, DigiCert, ...).
- The two sides derive a shared key. Everything after is encrypted.

 **Padlock icon** = encryption + verified identity.

Step 4: Send the HTTP request

```
GET / HTTP/2  
Host: wikipedia.org  
Accept: text/html  
User-Agent: Mozilla/5.0 ...
```

This is — literally — what your browser sends down the encrypted pipe.

Step 5: The server responds

```
HTTP/2 200 OK
Content-Type: text/html
Content-Length: 84211

<html>...</html>
```

200 OK = "here you go". Other common ones: **301** redirect, **404** not found, **500** server error.

CDNs: the same page, served from near you

Wikipedia's origin lives in a couple of data centers. You don't.

- A **CDN** (Content Delivery Network) keeps copies at **edge nodes** all over the world.
 - You talk to the closest copy — maybe 50 ms away, instead of 200 ms.
 - Brands: **Cloudflare**, **Akamai**, **AWS CloudFront**, **Fastly**.
-

The browser renders

1. Parses the HTML.
2. Fetches CSS, JavaScript, images, fonts — **each one is another HTTP request**.
3. Runs the JavaScript.
4. Paints pixels.

One "page load" is often **hundreds of requests** in the background.

Why this matters when something breaks

- **DNS issue** → this one site unreachable, other sites still work.
- **Certificate expired** → padlock warning, scary red page.
- **Slow page** → could be DNS, your network, the CDN, the origin server, or the page's own JavaScript. Any of them.

Knowing the layers turns "the internet is broken" into a question you can actually narrow down.

What this course won't teach you

Honest list of things we're *not* covering:

- **BGP** — how ISPs route between each other.
- **IPv6** details and dual-stack realities.
- The **TLS handshake math**.
- **TCP congestion control**.

They exist, they matter, they're someone's full-time job. See the *Reading on* list in the theory page.

Recap

- **Name** → **number** = DNS.
- **Encrypted call** = HTTPS (TCP + TLS + HTTP).
- **CDN** = a "near you" copy of the page.

Coming up: Lesson 03 — what "the cloud" actually sells.

