# Rendering a web page – step by step – Friendly Bit

Have you ever thought about what happens when you surf the web? It's not as simple as it seems:

1. You **type an URL** into address bar in your preferred browser.
2. The browser **parses the URL** to find the protocol, host, port, and path.
3. It **forms a HTTP request** (that was most likely the protocol)
4. To reach the host, it first needs to **translate** the human readable host **into an IP number**, and it does this by doing a DNS lookup on the host
5. Then a **socket needs to be opened** from the user's computer to that IP number, on the port specified (most often port 80)
6. When a connection is open, the **HTTP request is sent** to the host
7. The host **forwards the request** to the server software (most often Apache) configured to listen on the specified port
8. The **server inspects the request** (most often only the path), and **launches the server plugin needed** to handle the request (corresponding to the server language you use, PHP, Java, .NET, Python?)
9. The plugin gets access to the full request, and starts to prepare a HTTP response.
10. To construct the response a **database** is (most likely) **accessed**. A database search is made, based on parameters in the path (or data) of the request
11. Data from the database, together with other information the plugin decides to add, is **combined into a long string** of text (probably HTML).
12. The plugin **combines** that data with some meta data (in the form of HTTP headers), and **sends the HTTP response** back to the browser.
13. The browser receives the response, and **parses the HTML** (which with 95% probability is broken) in the response
14. A **DOM tree is built** out of the broken HTML
15. **New requests are made** to the server for each new resource that is found in the HTML source (typically images, style sheets, and JavaScript files). Go back to step 3 and repeat for each resource.
16. **Stylesheets are parsed**, and the rendering information in each gets attached to the matching node in the DOM tree
17. **Javascript is parsed and executed**, and DOM nodes are moved and style information is updated accordingly
18. The browser **renders the page** on the screen according to the DOM tree and the style information for each node
19. **You see** the page on the screen
20. You get annoyed the whole process was too slow.

I, too, get annoyed when the above steps take longer than one tenth of a second. But now at least I have some documentation to look at, while waiting the remaining fractions of a second before the page renders.

Spoiled we are, all of us.