Why we use Doctype in html?

The HTML syntax of HTML5 requires a DOCTYPE to be specified to ensure that the browser renders the page in standards mode. The DOCTYPE has no other purpose and is therefore optional for XML. Documents with an XML media type are always handled in standards mode. [DOCTYPE]

The DOCTYPE declaration is <!DOCTYPE html> and is case-insensitive in the HTML syntax. DOCTYPEs from earlier versions of HTML were longer because the HTML language was SGML-based and therefore required a reference to a DTD. With HTML5 this is no longer the case and the DOCTYPE is only needed to enable standards mode for documents written using the HTML syntax. Browsers already do this for <!DOCTYPE html>.

2)how browser communicates with web server

The TCP/IP stack has four layers: Application, Transport, Internet, and Network. At each layer there are different protocols that are used to standardize the flow of information, and each one is a computer program (running on your computer) that's used to format the information into a packet as it's moving down the TCP/IP stack. A packet is a combination of the Application Layer data, the Transport Layer header (TCP or UDP), and the IP layer header (the Network Layer takes the packet and turns it into a frame).

Updated May 08, 2017

Web browsers like Internet Explorer, Firefox, Chrome and Safari rank among the most popular network applications in the world. They're used for basic information browsing but also for various other needs including online shopping and casual gaming.

Web servers are what supply the content for web browsers; what the browser requests, the server delivers through internet network connections.

**Client-Server Network Design and the Web**

Web browsers and web servers function together as a [client-server](https://www.lifewire.com/introduction-to-client-server-networks-817420) system.

In computer networking, client-server is a standard method for designing applications where data is kept in central locations (server computers) and efficiently shared with any number of other computers (the clients) on request. All web browsers function as clients that request information from websites (servers).

Numerous web browser clients can request data from the same website. Requests can happen at all different times or simultaneously. Client-server systems conceptually call for all requests to the same site to be handled by one server. In practice, however, because the volume of requests to web servers can sometimes grow very large, web servers are often built as a distributed pool of multiple server computers.

For very large websites popular in different countries around the world, this web server pool is geographically distributed to help improve the response time to browsers. If the server is closer to the requesting device, it would follow that the time it takes to deliver the content is faster than if the server were further away.

**Network Protocols for Web Browsers and Servers**

Web browsers and servers communicate via [TCP/IP](https://www.lifewire.com/transmission-control-protocol-and-internet-protocol-816255). Hypertext Transfer Protocol [(HTTP)](https://www.lifewire.com/hypertext-transfer-protocol-817944) is the standard application protocol on top of TCP/IP supporting web browser requests and server responses.

Web browsers also rely on [DNS](https://www.lifewire.com/definition-of-domain-name-system-816295) to work with [URLs](https://www.lifewire.com/definition-of-uniform-resource-locator-817778) like *http://www.lifewire.com*.

These protocol standards enable different brands of web browsers to communicate with different brands of web servers without requiring special logic for each combination.

Like most internet traffic, web browser and server connections normally run through a series of intermediate [network routers](https://www.lifewire.com/how-routers-work-816456).

A basic web browsing session works like this:

* The user specifies a URL in their browser (either from a bookmark or by typing it in)
* The browser initiates a TCP connection to the web server or server pool (using port 80 by default) via its [IP address](https://www.lifewire.com/what-is-an-ip-address-818393) as published in DNS. As part of this process, the browser also makes DNS lookup requests to convert the URL to an IP address
* After the server completes acknowledgment of its side of the TCP connection, the browser sends HTTP requests to the server to retrieve the content
* After the server replies with content for the page, the browser retrieves it from the HTTP packets and displays it accordingly. Content can include embedded URLs for advertising banners or other third-party content, that in turn triggers the browser to issue new TCP connection requests to those locations. The browser may also save temporary information about its connections to local files on the client computer called cookies
* Any errors encountered during the request for the content might be shown as [HTTP status lines](https://www.lifewire.com/http-status-lines-2623465)