**What is the process between you typing a URL into a browser and the site finishing loading?**

1. You enter the website address into the address bar of the browser.

The Domain Name System (DNS) is a database that maintains a list of website names and all the IP addresses it links to. DNS makes it easier for people to remember websites by their domain name instead of memorizing the IP address.

1. When the user enters the website address, the browser checks the cache for a DNS record of the corresponding IP address of the website address. The browser runs this check in four caches

* the browser cache – the browsers maintains a cache of DNS records for a fixed duration and is the first place to check
* the OS cache – if the browser doesn’t contain the record, it makes a system call to the underlying operating system
* the router cache – if it still fails to retrieve the record, it continues its search in your router
* the ISP cache – if all the above search fails, the search moves to your ISP

Maintaining this data in caches help regulate network traffic and improve data transfer times.

1. If the browser fails to find the IP address in the ISP cache too, the ISP cache then sends off a DNS query to search multiple DNS servers to find the IP address. These requests are sent in small data packets that travel through various networking equipment between the client and the server before it reaches the correct DNS server. If the packets get lost, the user is presented with a request failed error. If it does reach the correct DNS server, it grabs the IP address and returns it to the browser.
2. On receiving the IP address, the browser will use internet protocols like TCP, to transfer data packets between your computer and the server. The connection is established using a process called the TCP/IP three-way handshake.
3. Once the TCP connection is established, the browser will send a GET request asking for the web page. A POST request is used if you must enter credentials or submit a form.
4. Upon receipt of the request, the server then handles the request and passes the request to a proper request handler. This may be a program written in PHP, ASP.NET etc. The request handler assembles a response in a particular format.
5. The server then sends out a HTTP response that contains the webpage along with status code, compression type, privacy information etc. The status code is useful to look up if you encounter an error as it tells us the status of the response.
6. The browser now displays this HTTP content. It does this in phases, first it renders the bare bones HTML page, then it sends out a GET request for additional elements on the page. These files are cached in the browser so it doesn’t have to fetch it the next time you visit the site.
7. Once the html page is loaded, the user can interact with the site. If he/she needs to login, a POST request is made and the browser will send the form data to the server for processing.  
   The server will authenticate the user credentials and the send the proper HTML content back to the user.