How browser and browser engines work- short high-level representation.

A web browser commonly referred to as a browser is an application software for accessing the World Wide Web. When a user requests a web page, browser retrieves the necessary content from the web server and then displays the page on the user's devices. The whole process is packed with complex and interesting procedures. To mention some of them handling user inputs, sending a URL request, preparation for rendering process, committing navigation, and rendering requested page. When you start an application, a process is created. A process can be described as an application's executing program. A thread is the one that lives inside of process and executes any part of its process's program. Modern browser (multi-threaded process), alleviate the problems of old browser (single-threaded process). These solutions include an improved stability, high performance and enhanced security using sandboxing techniques.

A web browser engine is a core software component of every web browser. For instance, google chrome has V8 engine while safari uses WebKit engine. These engines, in simple expression, translates raw data that we human can't understand into understandable page. To do this, the engine will undergo sophisticated procedures. These includes parsing HTML documents, creating an abstract syntax tree (AST), interpretation of tree, profiling, compilation, and optimization for better performance. In fact, modern browser engines get much smarter than before and makes decision ahead of time to give the best experience. One of the areas people working to achieve such efficiency is optimization. JavaScript engine performs compiler optimization, garbage collection, hot code management, caching and other runtime aspect that keeps JavaScript program running efficiently. In addition, function inlining and use of hidden classes used to make the work more efficient.