The order of processing scripts and style sheets

1. **Scripts is executed synchronously**

   The model of the web is synchronous. We expect scripts to be parsed and executed immediately when the parser reaches a <script> tag. The parsing of the document halts until the script has been executed. If the script is external then the resource must first be fetched from the network––this is also done synchronously, and parsing halts until the resource is fetched. This was the model for many years and is also specified in HTML4 and 5 specifications.

2. **Speculative(推测的) parsing**

   Both WebKit and Firefox do this optimization. While executing scripts, another thread parses the rest of the document and finds out what other resources need to be loaded from the network and loads them. In this way, resources can be loaded on parallel connections and overall speed is improved.

   — Note: the speculative parser only parses references to external resources like external scripts, style sheets and images: it doesn't modify the DOM tree––that is left to the main parser.

3. **Block scripts when parsing Style sheets**

   Conceptually it seems that since style sheets don't change the DOM tree, there is no reason to wait for them and stop the document parsing. There is an issue, though, of scripts asking for style information during the document parsing stage. If the style is not loaded and parsed yet, the script will get wrong answers and apparently this caused lots of problems.

   — Firefox blocks all scripts when there is a style sheet that is still being loaded and parsed.

   — WebKit blocks scripts only when they try to access certain style properties that may be affected by unloaded style sheets.

4.Render tree construction