JavaScript的执行包括两个阶段：一是载入文档内容，并执行<script>标签中的代码。二是异步事件驱动阶段。

JS是单线程的，单线程意味着任何时候JS引擎只会做一件事。单线程意味着JS执行的时候不能响应用户的输入，就是说JS不能执行得过长。如果你执行一个计算密集型的任务，它会给文档的加载带来很大的延迟。Web worker是一个用来执行密集型任务而不冻结用户界面的后台线程，web worker线程里面的代码不能访问DOM，只能和主线程和其他worker通过异步事件进行通信，并且web worker不能修改JS程序的基础单线程执行模型。

1. 浏览器创建Document对象，开始解析Web页面，解析HTML元素和他们的文本内容后添加Element对象和Text节点到文档中，此时document.readyState的值为loading
2. 当遇到没有async和defer的<script>元素时，把这些元素添加到文档中，然后开始执行行内或者外部脚本。同步执行。同步脚本可以看到他之前的文档内容和自身的<script>元素。
3. 遇到async的脚本，他们会异步下载，并尽快执行，但是解释器并不会停下来等他下载。
4. 文档解析完成，此时document.readyState变为’interactive’。
5. 开始执行defer脚本，异步async脚本当然也有可能会在这一阶段执行。
6. Defer脚本执行完成，Document对象触发DOMContentLoaded事件。标志着程序执行从同步执行阶段转换到了异步事件驱动阶段，但此时可能仍然有异步脚本在执行。
7. 文档完全解析完毕，但还要等待一些其他内容载入，如图片之类的。当所有的内容完成载入，async和defer脚本执行完毕，document.readyState变为complete，Window对象触发load事件。
8. 从此刻起，会完全进入异步事件驱动阶段。