JavaScript：

1. 封装：

私有数据属性必须在类的声明中声明，而且其名称需以 # 开头。

通过在属性前添加#将属性设为私有属性，然后在类内部使用，从外部无法访问该属性。

设置私有方法与其相同。

2. JSON：

JSON 是一种纯数据格式，它只包含属性，没有方法。

JSON 要求在字符串和属性名称周围使用双引号。单引号无效。

甚至一个错位的逗号或分号就可以导致 JSON 文件出错。

JSON 可以将任何标准合法的 JSON 数据格式化保存，不只是数组和对象。

对象和文本之间的转换：

parse(): 以文本字符串形式接受 JSON 对象作为参数，并返回相应的对象。

stringify(): 接收一个对象作为参数，返回一个对应的 JSON 字符串。

3. JavaScript 异步：

Promise:

fetch()返回的 promise 对象添加 then（）处理程序，之后调用 response.json（）方法，将新的 then（）处理程序传递到 response.json（）返回到 Promise 中。

在 promise 中，该对象提供了一个 catch（）方法来支持错误处理，当异步操作失败时，将结果传递给 catch（）的处理函数将被调用。

Promise 的几种状态：

待定（pending）：初始状态，既没有被兑现，也没有被拒绝。这是调用 fetch() 返回 Promise 时的状态，此时请求还在进行中。

已兑现（fulfilled）：意味着操作成功完成。当 Promise 完成时，它的 then() 处理函数被调用。

已拒绝（rejected）：意味着操作失败。当一个 Promise 失败时，它的 catch() 处理函数被调用。

Async 和 await：

Async：在函数开头添加 async，使之成为异步函数。

Await：在异步函数中，你可以在调用一个返回 Promise 的函数之前使用 await 关键字。这使得代码在该点上等待，直到 Promise 被完成，这时 Promise 的响应被当作返回值，或者被拒绝的响应被作为错误抛出。

使用 setTimeout（）来实现 alarm（）函数。