Private Properties and Information Hiding with JavaScript

The true die-hard fans of object-oriented design will notice that when using the prototype method for adding properties and functions to a JavaScript object, the added properties and functions are public and accessible to all other objects. For functions this typically isn’t a problem, as most functions should be exposed to external clients anyway. But in the case of properties, the fans of object-oriented design will point out that public properties break the concept of information hiding. Object-oriented design dictates that an object’s properties should be private and therefore not accessible to external clients. External clients should be able to access an object’s private properties only through publicly available functions.

A little known fact about JavaScript is that it is possible to create private properties that are not accessible to external clients and instead are accessible only via the object’s methods. Douglas Crockford\(^3\) has demonstrated a method of creating private properties in JavaScript. It’s rather simple, as summarized here:

- Private properties are defined in the constructor function using the \texttt{var} keyword.
- Private properties can be publicly accessed only by \textit{privileged functions}. A privileged function is a function that has been defined in the constructor using the \texttt{this} keyword. Privileged functions are accessible to external clients but also have access to the object’s private properties.

Let’s consider the \texttt{Vehicle} class from the earlier example. Say you want to make the \texttt{wheelCount} and \texttt{curbWeightInPounds} properties private and accessible only via publicly available methods. The new \texttt{Vehicle} object now looks like Listing 5-4.

**Listing 5-4. The Rewritten Vehicle Object**

```javascript
function Vehicle() {
    var wheelCount = 4;
    var curbWeightInPounds = 4000;

    this.getWheelCount = function() {
        return wheelCount;
    }

    this.setWheelCount = function(count) {
        wheelCount = count;
    }

    this.getCurbWeightInPounds = function() {
        return curbWeightInPounds;
    }

    this.setCurbWeightInPounds = function(weight) {
        curbWeightInPounds = weight;
    }
}
```

\(^3\) http://www.crockford.com/