Objects

* JavaScript is fundamentally about objects. Arrays are objects. Functions are objects. Objects are objects. So what are objects? Objects are collections of name-value pairs. The names are strings, and the values are strings, numbers, booleans, and objects (including arrays and functions). Objects are usually implemented as hashtables so values can be retrieved quickly.
* If a value is a function, we can consider it a method. When a method of an object is invoked, the this variable is set to the object. The method can then access the instance variables through the this variable.
* Objects can be produced by constructors, which are functions which initialize objects. Constructors provide the features that classes provide in other languages, including static variables and methods.

Public

In the constructor

* This technique is usually used to initialize public instance variables. The constructor's this variable is used to add members to the object.

function Container(param) {

this.member = param;

}

* So, if we construct a new object

var myContainer = new Container('abc');

* then myContainer.member contains 'abc'.

In the prototype

* This technique is usually used to add public methods. When a member is sought and it isn't found in the object itself, then it is taken from the object's constructor's prototype member. The prototype mechanism is used for inheritance. It also conserves memory. To add a method to all objects made by a constructor, add a function to the constructor's prototype:

Container.prototype.stamp = function (string) {

return this.member + string;

}

* So, we can invoke the method

myContainer.stamp('def')

which produces 'abcdef'.