

As funções próprias do construtor para inicializar objetos recém -criados.Fazendo isso está coberto no capítulo 9.

6.2.3 Protótipos

Antes de podermos cobrir a terceira técnica de criação de objetos, devemos fazer uma pausa por um momento para explicar protótipos.Quase todo objeto JavaScript tem um segundo objeto JavaScript associado a ele.Este segundo objeto é conhecido como protótipo, e o primeiro objeto herda as propriedades do protótipo.

Todos os objetos criados por literais de objeto têm o mesmo objeto de protótipo, e podemos nos referir a este protótipo objeto no código JavaScript como `Object.prototype`.Objetos criados usando a nova palavra -chave e um Invocação do construtor Use o valor da propriedade do protótipo de o construtor funciona como seu protótipo.Então o objeto criado por novo objeto `()` herda do `object.prototype`, assim como o Objeto criado por `{}` faz.Da mesma forma, o objeto criado por novo `Array ()` usa `Array.prototype` como seu protótipo e o objeto Criado por `new date ()` usa `date.prototype` como seu protótipo.

Isso pode ser confuso ao aprender o JavaScript pela primeira vez.Lembrar: Quase todos os objetos têm um protótipo, mas apenas um número relativamente pequeno de objetos têm uma propriedade de protótipo.São esses objetos com propriedades de protótipo que definem os protótipos para todos os outros objetos.

`Object.Prototype` é um dos objetos raros que não possuem protótipo: Não herda nenhuma propriedade.Outros protótipos objetos são normais objetos que têm um protótipo.A maioria dos construtores embutidos (e a maioria