construtores definidos pelo usuário) têm um protótipo que herda
Object.prototype.Por exemplo, Date.Prototype herda
propriedades do object.prototype, então um objeto de data criado por
new Date () herda as propriedades de ambos DATE.Prototype e
Object.prototype.Esta série vinculada de objetos de protótipo é
conhecido como uma cadeia de protótipo.
Uma explicação de como a herança da propriedade funciona é no §6.3.2.
O capítulo 9 explica a conexão entre protótipos e construtores
em mais detalhes: mostra como definir novas "classes" de objetos por
escrever uma função construtora e definir sua propriedade de protótipo para
o objeto de protótipo a ser usado pelas "instâncias" criadas com isso
construtor.E aprenderemos a consultar (e até mudar) o
protótipo de um objeto no §14.3.

6.2.4 Object.Create ()

Object.create () cria um novo objeto, usando seu primeiro argumento como
O protótipo desse objeto:

```
Seja o1 = object.create ({x: 1, y: 2});// O1 herda
propriedades x e y.
O1.x + o1.y // => 3
```

Você pode passar nulo para criar um novo objeto que não tenha um
protótipo, mas se você fizer isso, o objeto recém -criado não herdará
Qualquer coisa, nem mesmo métodos básicos como ToString () (o que significa
Também não funcionará com o operador +):

```
Seja o2 = object.create (nulo);// O2 herda não
adereços ou métodos.
```