Agora, suponha que você atribua à propriedade X do objeto o.Se o já
possui uma propriedade própria (não-herdada) chamada X, depois a tarefa
simplesmente altera o valor desta propriedade existente.Caso contrário, o
A tarefa cria uma nova propriedade chamada x no objeto o.Se o
Anteriormente herdou a propriedade X, essa propriedade herdada é agora
escondido pela propriedade própria recém -criada com o mesmo nome.
A atribuição de propriedade examina a cadeia de protótipo apenas para determinar
se a tarefa é permitida.Se o herdar uma propriedade somente leitura
Nomeado X, por exemplo, a atribuição não é permitida.(Detalhes
sobre quando uma propriedade pode ser definida está em §6.3.3.) Se a tarefa for
permitido, no entanto, sempre cria ou define uma propriedade no original
Objeto e nunca modifica objetos na cadeia de protótipos.O fato disso
A herança ocorre ao consultar propriedades, mas não quando as definir
é uma característica fundamental do JavaScript, porque nos permite seletivamente
substituir propriedades herdadas:

```
Seja unitcircle = {r: 1};// um objeto para herdar
de
Seja c = object.create (unitcircle);// c herda a propriedade
r
c.x = 1;c.y = 1;// c define dois
propriedades próprias
c.r = 2;// c substitui seu
propriedade herdada
unitcircle.r // => 1: o protótipo é
não afetado
```

Há uma exceção à regra de que uma tarefa de propriedade também
falha ou cria ou define uma propriedade no objeto original.Se o herdar o
Propriedade X, e essa propriedade é uma propriedade acessadora com um setter
Método (ver §6.10.6), então esse método do setter é chamado em vez de