O método HasOwnProperty () de um objeto testa se isso
O objeto possui uma propriedade própria com o nome fornecido.Retorna falsa para
Propriedades herdadas:
Seja o = {x: 1};
O.HasownProperty ("x") // => true: o tem um próprio
Propriedade x
O.HasownProperty ("y") // => false: o não tem um
Propriedade y
O.HasownProperty ("ToString") // => false: ToString é um
propriedade herdada
O PropertyIsenumerable () refina o
teste HASOWNPROPERTY ().Ele retorna verdadeiro apenas se o nomeado
A propriedade é uma propriedade própria e seu atributo enumerável é verdadeiro.
Certas propriedades embutidas não são enumeráveis.Propriedades criadas por
O código JavaScript normal é enumerável, a menos que você tenha usado um dos
Técnicas mostradas no §14.1 para torná-las que não são inebriantes.
Seja o = {x: 1};
O.Propertyisenumerable ("x") // => true: o tem um próprio
propriedade enumerável x
O.Propertyisenumerable ("ToString") // => false: não um próprio
propriedade
Object.prototype.propertyisenumerable ("tostring") // =>
Falso: não enumerável
Em vez de usar o operador em In, muitas vezes é suficiente simplesmente consultar
a propriedade e o uso! == para garantir que não seja indefinido:
Seja o = {x: 1};
o.x! == indefinido // => true: o tem uma propriedade x
o.y! == indefinido // => false: o não tem um
Propriedade y
O.ToString! == indefinido // => true: o herda uma toque