ausência de valor e geralmente pode ser usada de forma intercambiável.A igualdade
operador == considera -os iguais.(Use a igualdade estrita
operador === para distingui -los.) Ambos são valores falsamente: eles se comportam
Como false quando um valor booleano é necessário.Nem nulo nem
indefinido têm propriedades ou métodos.De fato, usando.ou [] para
Acesse uma propriedade ou método desses valores causa um TypeError.
Considero indefinido para representar um nível de sistema, inesperado, ou
ausência de valor e nulo para representar um nível de programa,
Ausência normal ou esperada de valor.Evito usar nulo e
indefinido quando posso, mas se eu precisar atribuir um desses valores a
uma variável ou propriedade ou passar ou retornar um desses valores para ou de um
função, eu geralmente uso nulo.Alguns programadores se esforçam para evitar nulos
inteiramente e use indefinido em seu lugar onde puder.

3.6 Símbolos

Os símbolos foram introduzidos no ES6 para servir como nomes de propriedades que não são de corda.
Para entender os símbolos, você precisa saber que o JavaScript's
Tipo de objeto fundamental é uma coleção não ordenada de propriedades,
onde cada propriedade tem um nome e um valor.Os nomes de propriedades são
tipicamente (e até ES6, era exclusivamente) strings.Mas em ES6 e
Mais tarde, os símbolos também podem servir a esse propósito:

```
Seja strname = "Nome da string";// uma string para usar como um
Nome da propriedade
deixe symname = símbolo ("propname");// um símbolo para usar como um
Nome da propriedade
typeof strname // => "string": strname é
uma corda
typeof symname // => "símbolo": symname é
```