

para uma alternativa.

Antes do ES6, esse idioma é frequentemente usado em funções para fornecer padrão

Valores para parâmetros:

```
// copie as propriedades de O para P e retorne P
```

```
cópia da função (o, p) {
```

```
P = P || {}; // Se nenhum objeto foi aprovado para P, use um recém  
objeto criado.
```

```
// O corpo da função vai aqui
```

```
}
```

No ES6 e mais tarde, no entanto, esse truque não é mais necessário porque o

O valor do parâmetro padrão pode ser simplesmente escrito na função

Definição em si: função cópia (o, p = {}) {...}.

4.10.3 Lógico não (!)

O !O operador é um operador unário; É colocado antes de um único operando.

Seu objetivo é inverter o valor booleano de seu operando. Por exemplo, se

X é verdade, !X avalia como falso. Se x é falsamente, então !X é verdadeiro.

Ao contrário do && e || operadores, o ! operador converte seu operando para
um valor booleano (usando as regras descritas no capítulo 3) antes

invertendo o valor convertido. Isso significa isso! sempre retorna verdadeiro

ou falso e que você pode converter qualquer valor x em seu equivalente

Valor booleano aplicando este operador duas vezes: !!x (consulte §3.9.2).

Como um operador unário, ! tem alta precedência e se liga firmemente. Se você

deseja inverter o valor de uma expressão como p && q, você precisa usar

Parênteses: !(P && q). Vale a pena notar duas leis de booleanos