

Toda expressão de invocação inclui um par de parênteses e um expressão antes dos parênteses abertos. Se essa expressão for uma propriedade de acesso, então a invocação é conhecida como um método de invocação. Nas invocações de método, o objeto ou a matriz que é o sujeito do acesso à propriedade se torna o valor dessa palavra-chave enquanto o corpo da função está sendo executado. Isso permite um objeto-paradigma de programação orientado no qual as funções (que chamamos "Métodos" quando usado dessa maneira) opera no objeto de que são papel. Veja o capítulo 9 para obter detalhes.

4.5.1 Invocação condicional

No ES2020, você também pode invocar uma função usando `?.()`. Em vez de `()`. Normalmente quando você invoca uma função, se a expressão à esquerda dos parênteses são nulos ou indefinidos ou qualquer outra não função, um `TypeError` é jogado. Com o novo `?.()` Sintaxe de invocação, se o expressão à esquerda do `?.` avalia para nulo ou indefinido, então toda a expressão de invocação avalia para indefinido e não uma exceção é lançada.

Objetos de matriz têm um método de classificação `()` que pode opcionalmente ser passado um argumento da função que define a ordem de classificação desejada para a matriz elementos. Antes do ES2020, se você quisesse escrever um método como `classin()` que leva um argumento de função opcional, você normalmente use uma instrução `IF` para verificar se o argumento da função foi definido antes de invocá-lo no corpo do `IF`:

Função quadrada (`x, log`) `{// O segundo argumento é um`