

constante codificada.

```
Seja max = maxWidth || Preferências.MaxWidth || 500;
```

O problema com esse uso idiomático é que zero, a corda vazia e false são todos valores falsamente que podem ser perfeitamente válidos em alguns circunstâncias. Neste exemplo de código, se maxwidth for zero, esse valor será ignorado. Mas se mudarmos o || operador para ??, acabamos com uma expressão em que zero é um valor válido:

```
// Se a maxwidth for definida, use isso. Caso contrário, procure um valor em
```

```
// O objeto de preferências. Se isso não estiver definido, use um constante codificada.
```

```
Deixe Max = MaxWidth ?? Preferências.MaxWidth ?? 500;
```

Aqui estão mais exemplos mostrando como ?? funciona quando o primeiro operando é falsamente. Se esse operando é falsamente, mas definido, então ?? retorna. É somente quando o primeiro operando é "nulo" (ou seja, nulo ou indefinido) que este operador avalia e retorna o segundo operando:

```
Let Options = {Timeout: 0, Title: "", Verbose: false, n:  
nulo};
```

```
options.timeout ?? 1000 // => 0: conforme definido no objeto
```

```
options.title ?? "Untitled" // => "": conforme definido no  
objeto
```

```
options.verbose ?? verdadeiro // => false: conforme definido no  
objeto
```

```
options.quiet ?? false // => false: a propriedade não é  
definido
```

```
options.n ?? 10 // => 10: A propriedade é nula
```

Observe que o tempo limite, título e expressões detalhadas aqui teria valores diferentes se usássemos || em vez de ??.