ignore qualquer coisa que se segue.Se o primeiro personagem não espacial não for parte de um literal numérico válido, eles retornam Nan:

```
parseint ("3 ratos cegos") // => 3
parsefloat ("3,14 metros") // => 3,14
parseint (" -12.34") // => -12
parseint ("0xff") // => 255
parseint ("0xff") // => 255
parseint (" -0xff") // => -255
parsefloat (". 1") // => 0,1
parseint ("0,1") // => 0
parseint (". 1") // => nan: os números inteiros não podem começar
com "."
parsefloat ("$ 72,47") // => nan: os números não podem começar
com "$"
```

parseint () aceita um segundo argumento opcional especificando o Radix (base) do número a ser analisado.Os valores legais estão entre 2 e 36. Por exemplo:

```
parseint ("11", 2) // => 3: (1*2 + 1)
parseint ("ff", 16) // => 255: (15*16 + 15)
parseint ("zz", 36) // => 1295: (35*36 + 35)
parseint ("077", 8) // => 63: (7*8 + 7)
parseint ("077", 10) // => 77: (7*10 + 7)
```

3.9.3 Objeto de conversões primitivas

As seções anteriores explicaram como você pode converter explicitamente valores de um tipo para outro tipo e explicaram JavaScript's conversões implícitas de valores de um tipo primitivo para outro Tipo primitivo.Esta seção abrange as regras complicadas que JavaScript usa para converter objetos em valores primitivos.É longo e obscuro, e se esta é a sua primeira leitura deste capítulo, você deve sentir