

```
const Café = 1; // Esta constante é nomeada "Caf \ u {e9}"
const Café = 2; // Esta constante é diferente: "Cafe \ u {301}"
Café // => 1: esta constante tem um valor
Café // => 2: Esta constante indistinguível tem um
valor diferente
```

O padrão Unicode define a codificação preferida para todos os personagens e especifica um procedimento de normalização para converter texto em um canônico forma adequada para comparações. JavaScript assume que o código -fonte está interpretando já foi normalizada e não faz nenhum normalização por conta própria. Se você planeja usar caracteres unicode em seu Programas JavaScript, você deve garantir que seu editor ou algum outro A ferramenta executa a normalização unicode do seu código -fonte para prevenir você de acabar com diferente, mas visualmente indistinguível identificadores.

2.6 Semicolons opcionais

Como muitas linguagens de programação, o JavaScript usa o semicolon (;) para separar declarações (consulte o capítulo 5) um do outro. Isso é importante para deixar claro o significado do seu código: sem um separador, o final de uma declaração pode parecer o começo de o próximo, ou vice -versa. Em JavaScript, você geralmente pode omitir o Semicolon entre duas declarações se essas declarações forem escritas em linhas separadas. (Você também pode omitir um ponto e vírgula no final de um programa ou se o próximo token no programa for uma cinta curta fechada:}). Muitos Programadores JavaScript (e o código neste livro) usam semicolons para marque explicitamente os fins das declarações, mesmo onde elas não estão obrigatório. Outro estilo é omitir semicolons sempre que possível, usando eles apenas nas poucas situações que os exigem. Qualquer estilo que você