

não em palavras -chave do idioma).O Unicode escapa para o personagem "é".

Por exemplo, é `\u00e9`;Aqui estão três maneiras diferentes de escrever um

Nome da variável que inclui este personagem:

Deixe Café = 1; // Defina uma variável usando um caractere unicode

`caf \u00e9 // => 1`;Acesse a variável usando uma fuga

Sequência

`caf \u {e9} // => 1`;Outra forma da mesma fuga

Sequência

As primeiras versões do JavaScript suportaram apenas a fuga de quatro dígitos

Sequência.A versão com aparelho encaracolado foi introduzido em ES6 para

apoiar melhor os pontos de codepates unicode que requerem mais de 16 bits, como

Como emoji:

`console.log ("\u {1f600}");` // imprime um rosto sorridente emoji

O Unicode Escapes também pode aparecer nos comentários, mas desde os comentários

são ignorados, eles são simplesmente tratados como caracteres ASCII nesse contexto

e não interpretado como unicode.

2.5.2 Normalização unicode

Se você usa caracteres não-ASCII em seus programas JavaScript, você

deve estar ciente de que o Unicode permite mais de uma maneira de codificar o

mesmo personagem.A string "é", por exemplo, pode ser codificada como o

caractere unicode único `\u00e9` ou como um ASCII regular `?e?` seguido

pelo sotaque agudo combinando Mark `\U0301`.Essas duas codificações

normalmente parece exatamente o mesmo quando exibido por um editor de texto, mas

Eles têm codificações binárias diferentes, o que significa que são consideradas

Diferente por JavaScript, que pode levar a programas muito confusos: