

de volta ao topo do loop e avaliando a expressão novamente.Outro maneira de dizer isso é que o intérprete executa a declaração repetidamente enquanto a expressão é verdadeira.Observe que você pode criar um loop infinito com a sintaxe enquanto (true).

Geralmente, você não deseja que o JavaScript execute exatamente o mesmo operação repetidamente.Em quase todos os loops, um ou mais

As variáveis ??mudam com cada iteração do loop.Desde as variáveis

Mudança, as ações executadas executando a declaração podem diferir cada tempo através do loop.Além disso, se a variável em mudança ou

Variáveis ??estão envolvidas na expressão, o valor da expressão pode

Seja diferente a cada vez através do loop.Issso é importante;de outra forma,

Uma expressão que começa a verdade nunca mudaria, e o loop

nunca terminaria!Aqui está um exemplo de um loop de tempo que imprime o números de 0 a 9:

```
deixe count = 0;
while (contagem <10) {
  console.log (contagem);
  contagem ++;
}
```

Como você pode ver, a contagem de variáveis ??começa em 0 e é incrementada

Cada vez que o corpo do loop corre.Uma vez que o loop execute 10

vezes, a expressão se torna falsa (ou seja, a contagem de variáveis ??é não mais de 10), o fim da declaração termina e o intérprete

pode passar para a próxima declaração no programa.Muitos loops têm um

contador de contra -variável.Os nomes da variável i, j e k são

comumente usado como contadores de loop, embora você deva usar mais

Nomes descritivos se tornar seu código mais fácil de entender.