

Em outras palavras, a expressão inicializa é avaliada uma vez, antes do Loop começa. Para ser útil, essa expressão deve ter efeitos colaterais (geralmente uma tarefa). JavaScript também permite a inicialização para ser uma Declaração de declaração variável para que você possa declarar e inicializar um contador de loop ao mesmo tempo. A expressão de teste é avaliada antes cada iteração e controla se o corpo do loop é executado. Se O teste avalia a um valor verdadeiro, a afirmação que é o corpo do O loop é executado. Finalmente, a expressão de incremento é avaliada. De novo, Isso deve ser uma expressão com efeitos colaterais para ser útil. Geralmente, é uma expressão de atribuição ou usa o ++ ou - operadores.

Podemos imprimir os números de 0 a 9 com um loop como o seguindo. Contrastá -lo com o equivalente enquanto o loop mostrado na Seção anterior:

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

Loops podem se tornar muito mais complexos do que este exemplo simples, de claro, e às vezes várias variáveis ??mudam com cada iteração de o loop. Esta situação é o único lugar que o operador de vírgula está comumente usado em JavaScript; Ele fornece uma maneira de combinar múltiplos inicialização e incremento expressões em uma única expressão

Adequado para uso em um loop:

```
deixe eu, j, soma = 0;  
para (i = 0, j = 10; i < 10; i++, j--) {  
  soma += i * j;  
}
```