

▼ C# Fundamentals For

For:

for (referência de C#)

13/06/2018 • 2 minutos para o fim da leitura •   

A instrução `for` executa uma instrução ou um bloco de instruções enquanto uma expressão booliana especificada é avaliada como `true`.


Em qualquer ponto dentro do bloco de instrução `for`, você pode sair do loop usando a instrução `break` ou seguir para a próxima iteração no loop usando a instrução `continue`. Você também pode sair de um `for` loop pelas instruções `goto`, `Return` ou `throw`.

For:

Estrutura da instrução `for`

A instrução `for` define as seções de *inicializador*, *condição* e *iterador*:

C#


 Copiar

```
for (initializer; condition; iterator)
    body
```

Todas as três seções são opcionais. O corpo do loop é uma instrução ou um bloco de instruções.

A exemplo a seguir mostra a instrução `for` com todas as seções definidas:

C#

 Copiar

 Executar

```
for (int i = 0; i < 5; i++)
{
    Console.WriteLine(i);
}
```

For:


A seção *inicializador*

As instruções na seção de *inicializador* são executadas apenas uma vez, antes de entrar no loop. A seção *inicializador* é uma das seguintes:

- A declaração e a inicialização de uma variável de loop local, que não pode ser acessada de fora do loop.
- Zero ou mais expressões de instrução da lista a seguir, separadas por vírgulas:
 - instrução de [atribuição](#)
 - invocação de um método
 - prefixo ou sufixo da expressão [incrementar](#), como `++i` ou `i++`
 - prefixo ou sufixo da expressão [decrementar](#), como `--i` ou `i--`
 - criação de um objeto usando o operador [new](#)
 - expressão [await](#)

A seção *inicializador* no exemplo acima declara e inicializa a variável de loop local `i`:

C#

 Copiar


```
int i = 0
```


For:

A seção *condição*

A seção *condição*, se presente, deverá ser uma expressão booliana. Essa expressão é avaliada antes de cada iteração do loop. Se a seção *condição* não estiver presente ou a expressão booliana for avaliada como `true`, a próxima iteração do loop será executada; caso contrário, o loop será finalizado.

A seção *condição* no exemplo acima determina se o loop será encerrado com base no valor da variável de loop local:

C#	 Copiar
<code>i < 5</code>	


For:

A seção *iterador*

A seção *iterador* define o que acontece após cada iteração do corpo do loop. A seção *iterador* contém zero ou mais das expressões de instrução a seguir, separadas por vírgulas:

- instrução de [atribuição](#)
- invocação de um método
- prefixo ou sufixo da expressão [incrementar](#), como `++i` ou `i++`
- prefixo ou sufixo da expressão [decrementar](#), como `--i` ou `i--`
- criação de um objeto usando o operador [new](#)
- expressão [await](#)

A seção *iterador* no exemplo acima incrementa a variável de loop local:

C#	 Copiar
<code>i++</code>	

For:

Exemplos

O exemplo a seguir ilustra vários usos menos comuns das seções de instrução `for`: atribuir um valor a uma variável de loop externa na seção *inicializador*, invocar um método inicializa nas seções de *inicializador* e de *iterador* e alterar os valores de duas variáveis na seção de *iterador*. Selecione **Executar** para executar o código de exemplo. Depois disso, você pode modificar o código e executá-lo novamente.

C#

Copiar

Executar

```
int i;  
int j = 10;  
for (i = 0, Console.WriteLine($"Start: i={i}, j={j}"); i < j; i++, j--, Console.WriteLine($"Step: i={i}, j={j}"))  
{  
    // Body of the loop.  
}
```

O exemplo a seguir define o loop `for` infinito:

C#

Copiar

```
for ( ; ; )  
{  
    // Body of the loop.  
}
```

For:

- **Links de documentação oficial:**
- **Documentação:**
- **<https://docs.microsoft.com/pt-br/dotnet/csharp/language-reference/keywords/for>**



Dúvidas?