

La instrucción `for` ejecuta una instrucción o un bloque de instrucciones mientras una expresión booleana especificada se evalúa como `true`.

En cualquier punto del bloque de instrucciones `for`, se puede salir del bucle mediante la instrucción `break`, o bien se puede ir a la siguiente iteración del bucle mediante la instrucción `continue`. También se puede salir de un bucle `for` mediante las instrucciones `goto`, `return` o `throw`.

Estructura de la instrucción `for`

La instrucción `for` define las secciones *inicializador*, *condición* e *iterador*:

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

Las tres secciones son opcionales. El cuerpo del bucle es una instrucción o un bloque de instrucciones.

En el siguiente ejemplo se muestra la instrucción `for` con todas las secciones definidas:

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

La sección *inicializador*

Las instrucciones de la sección *inicializador* se ejecutan solo una vez, antes de entrar en el bucle. La sección *inicializador* es cualquiera de las siguientes:

- La declaración e inicialización de una variable de bucle local, a la que no se puede tener acceso desde fuera del bucle.
- Ninguna, una o varias expresiones de instrucción de la siguiente lista, separadas por comas:
 - instrucción `assignment`
 - invocación de un método
 - expresión de incremento de prefijo o sufijo, como `++i` o `i++`
 - expresión de decremento de prefijo o sufijo, como `--i` o `i--`
 - creación de un objeto mediante la palabra clave `new`
 - expresión `await`

La sección *inicializador* del ejemplo anterior declara e inicializa la variable de bucle local `i`:

```
int i = 0
```

La sección *condición*

La sección *condición*, si está presente, debe ser una expresión booleana. Dicha expresión se evalúa antes de cada iteración del bucle. Si la sección *condición* no está presente o la expresión booleana se evalúa como `true`, se ejecutará la siguiente iteración del bucle; en caso contrario, se sale del bucle.

La sección *condición* del ejemplo anterior determina si el bucle finaliza en función del valor de la variable de bucle local:

```
i < 5
```

La sección *iterador*

La sección *iterador* define lo que sucede después de cada iteración del cuerpo del bucle. La sección *iterador* contiene ninguna o más de las siguientes expresiones de instrucción, separadas por comas:

- instrucción assignment
- invocación de un método
- expresión de incremento de prefijo o sufijo, como `++i` o `i++`
- expresión de decremento de prefijo o sufijo, como `--i` o `i--`
- creación de un objeto mediante la palabra clave `new`
- expresión `await`

La sección *iterador* del ejemplo anterior incrementa la variable de bucle local:

```
i++
```

Ejemplos

En el ejemplo siguiente se muestran varios usos menos comunes de las secciones de la instrucción `for`: asignar un valor a una variable de bucle externa en la sección *inicializador*, invocar un método en las secciones *inicializador* e *iterador*, y cambiar los valores de dos variables en la sección *iterador*.

```
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.
}
```

En el ejemplo siguiente se define el bucle for infinito:

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