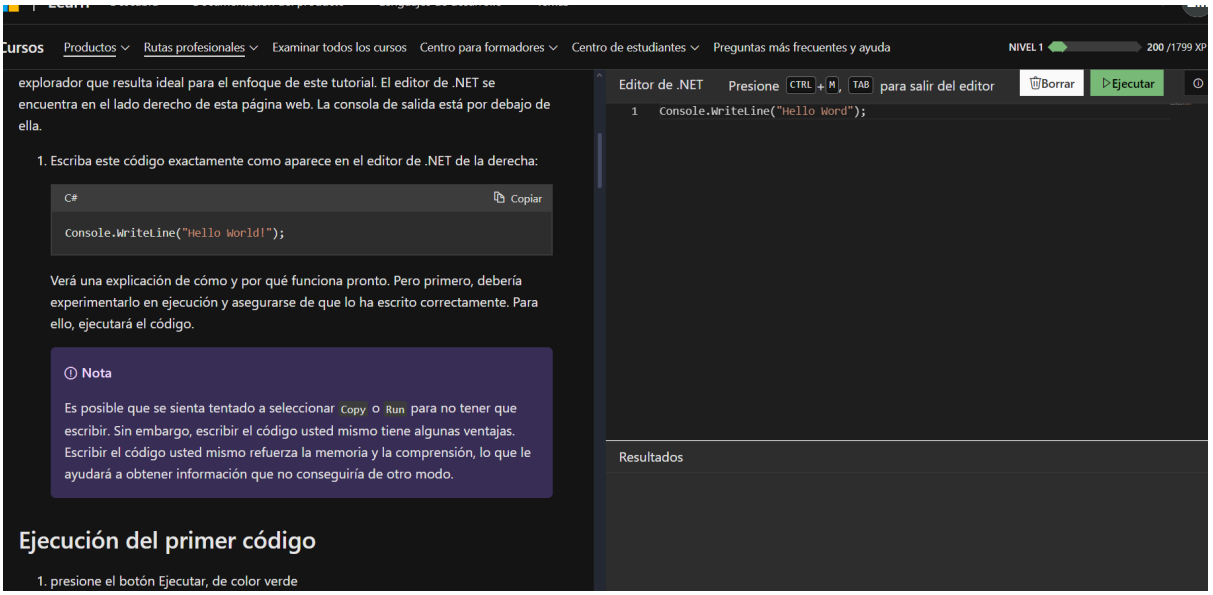
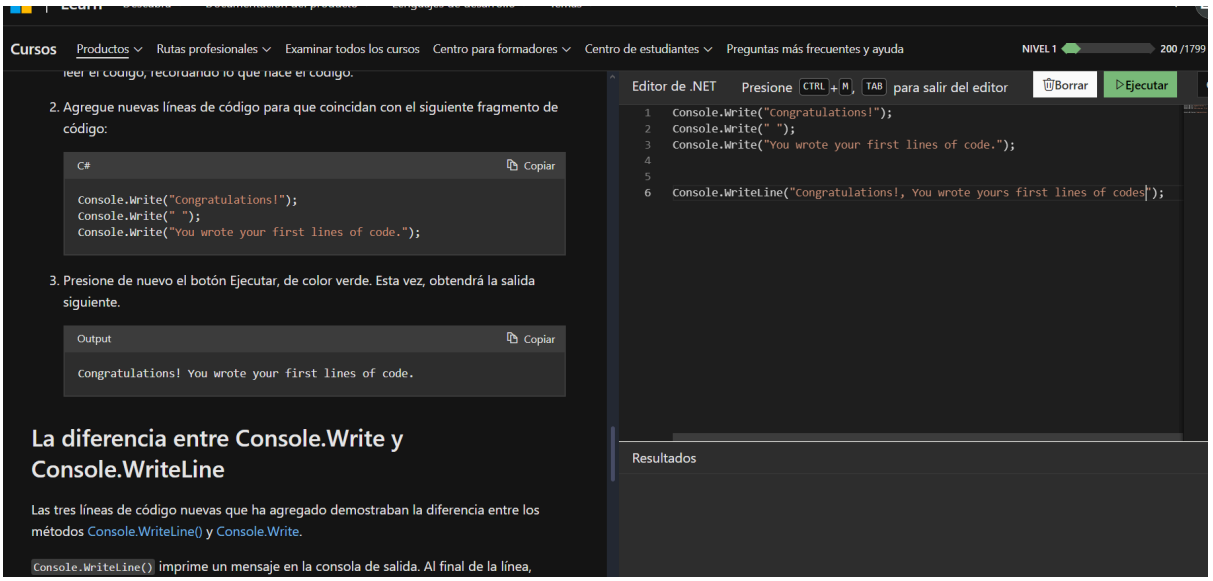
Universidad Rafael Landívar
Facultad de Ingeniería.
Licenciatura en ingeniería en informática y sistemas
Sección: 9
Catedrático: Ing. AGUILAR ROJAS LUIS ENRIQUE

Actividad 1-Semana 17
Pensamiento Computacional

Estudiante: Esteban Josué Maldonado Velasco 1053424

Guatemala, 10 de mayo de 2024


Cursos
Productos
Rutas profesionales
Examinar todos los cursos
Centro para formadores
Centro de estudiantes
Preguntas más frecuentes y ayuda
NIVEL 1
200 /1799 XP
explorador que resulta ideal para el enfoque de este tutorial. El editor de .NET se encuentra en el lado derecho de esta página web. La consola de salida está por debajo de ella.
1. Escriba este código exactamente como aparece en el editor de .NET de la derecha:
C#
Copiar
Console.WriteLine("Hello World!");
Verá una explicación de cómo y por qué funciona pronto. Pero primero, debería experimentarlo en ejecución y asegurarse de que lo ha escrito correctamente. Para ello, ejecutará el código.
Nota
Es posible que se sienta tentado a seleccionar Copy o Run para no tener que escribir. Sin embargo, escribir el código usted mismo tiene algunas ventajas. Escribir el código usted mismo refuerza la memoria y la comprensión, lo que le ayudará a obtener información que no conseguiría de otro modo.
Ejecución del primer código
1. presione el botón Ejecutar, de color verde
Editor de .NET
Presione CTRL + M, TAB para salir del editor
Borrar
Ejecutar
1 Console.WriteLine("Hello Word");
Resultados


Cursos
Productos
Rutas profesionales
Examinar todos los cursos
Centro para formadores
Centro de estudiantes
Preguntas más frecuentes y ayuda
NIVEL 1
200 /1799
leer el código, recordando lo que hace el código.
2. Agregue nuevas líneas de código para que coincidan con el siguiente fragmento de código:
C#
Copiar
Console.Write("Congratulations!");
Console.Write(" ");
Console.Write("You wrote your first lines of code.");
3. Presione de nuevo el botón Ejecutar, de color verde. Esta vez, obtendrá la salida siguiente.
Output
Copiar
Congratulations! You wrote your first lines of code.
La diferencia entre Console.Write y Console.WriteLine
Las tres líneas de código nuevas que ha agregado demostraban la diferencia entre los métodos Console.WriteLine() y Console.Write.
Console.WriteLine() imprime un mensaje en la consola de salida. Al final de la línea,
Editor de .NET
Presione CTRL + M, TAB para salir del editor
Borrar
Ejecutar
1 Console.Write("Congratulations!");
2 Console.Write(" ");
3 Console.Write("You wrote your first lines of code.");
4
5
6 Console.WriteLine("Congratulations!, You wrote yours first lines of codes");
Resultados

Assignments

- [x] Complete the Write Your First C# Code module on Microsoft Learn. Then, answer the question below.

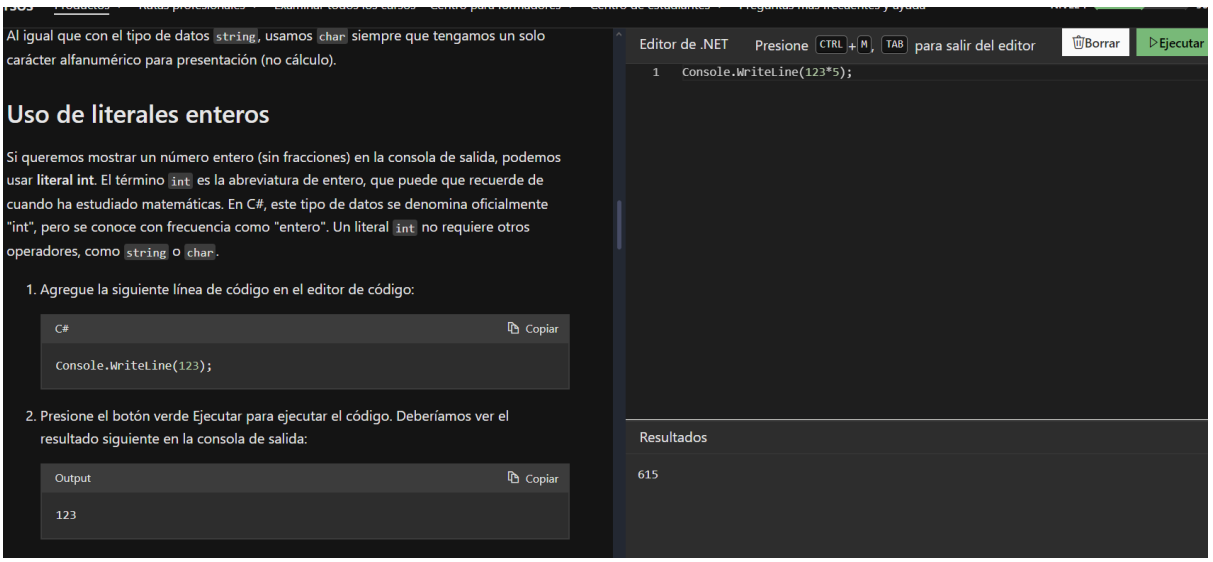
Question

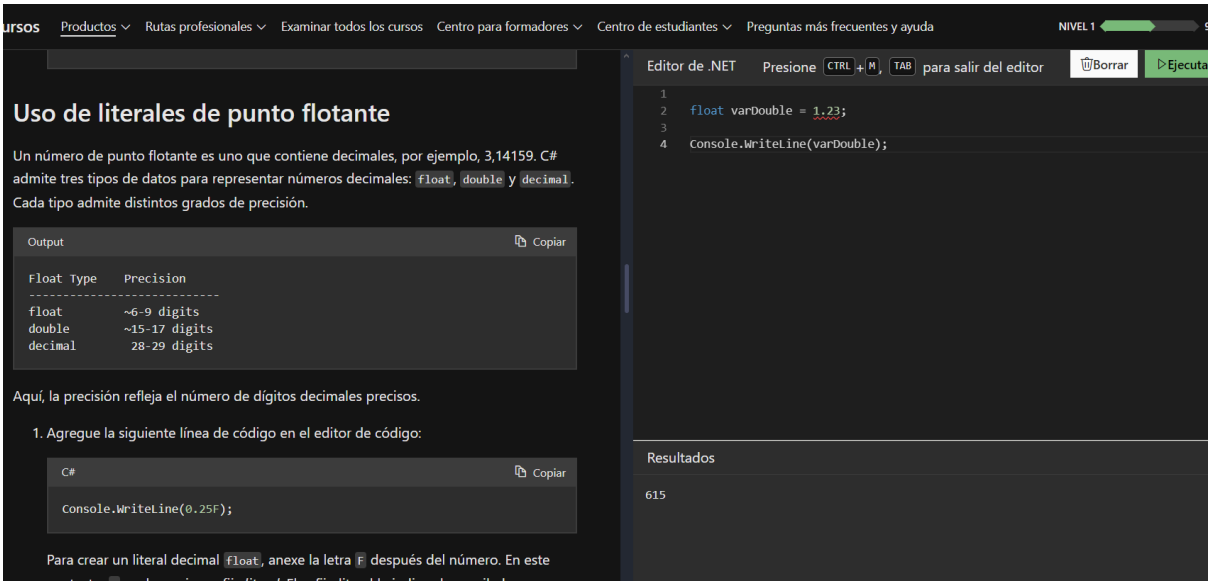
What is the difference between `Console.Write` and `Console.WriteLine`?

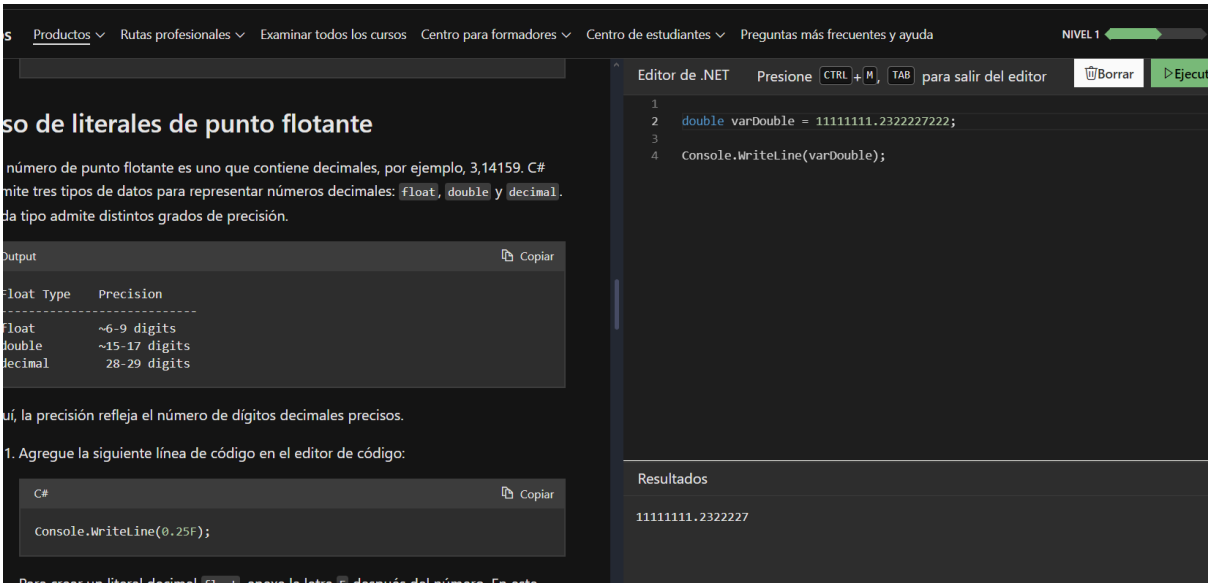
- ○ `Console.Write` prints the output on a new line.
- ○ `Console.WriteLine` prints the output on a new line.
- ◉ `Console.WriteLine` appends a new line after the output.

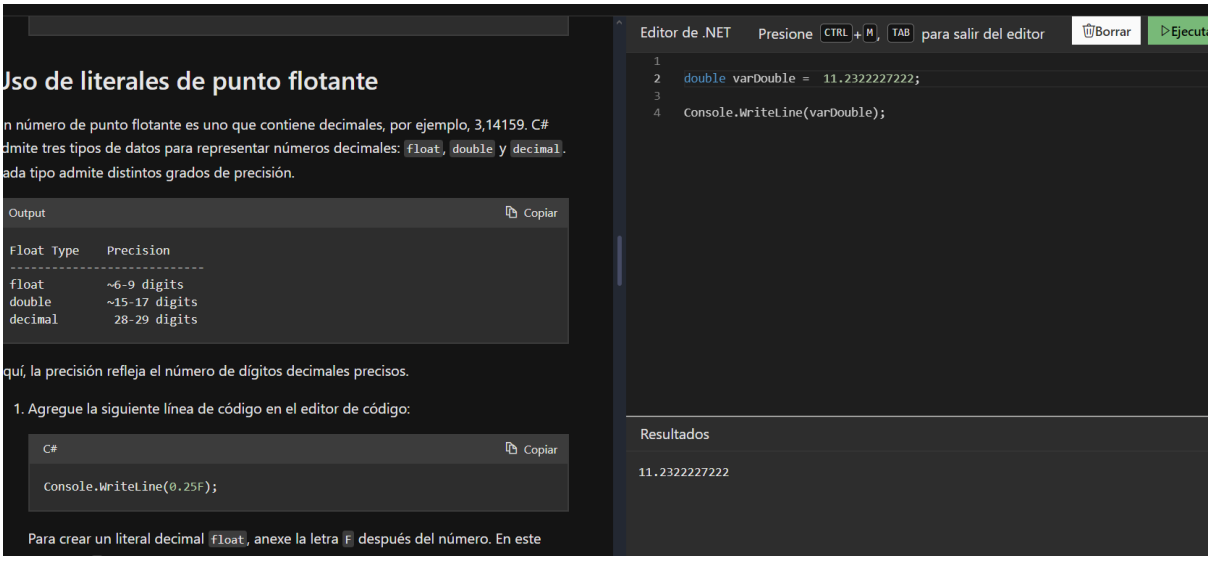
Check your answer

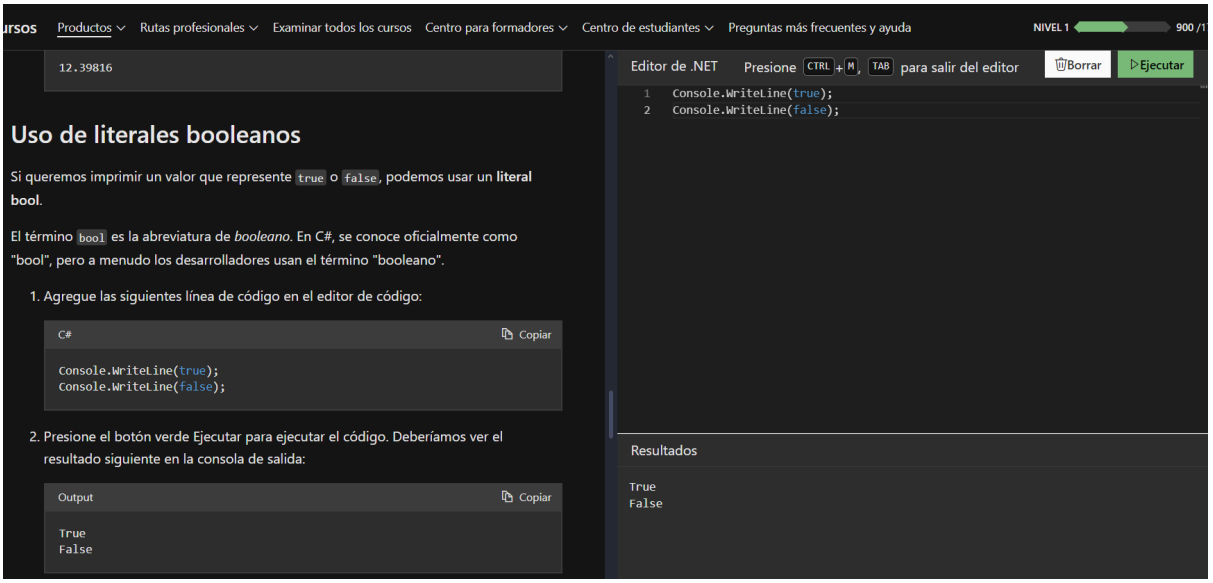
Ask for Help

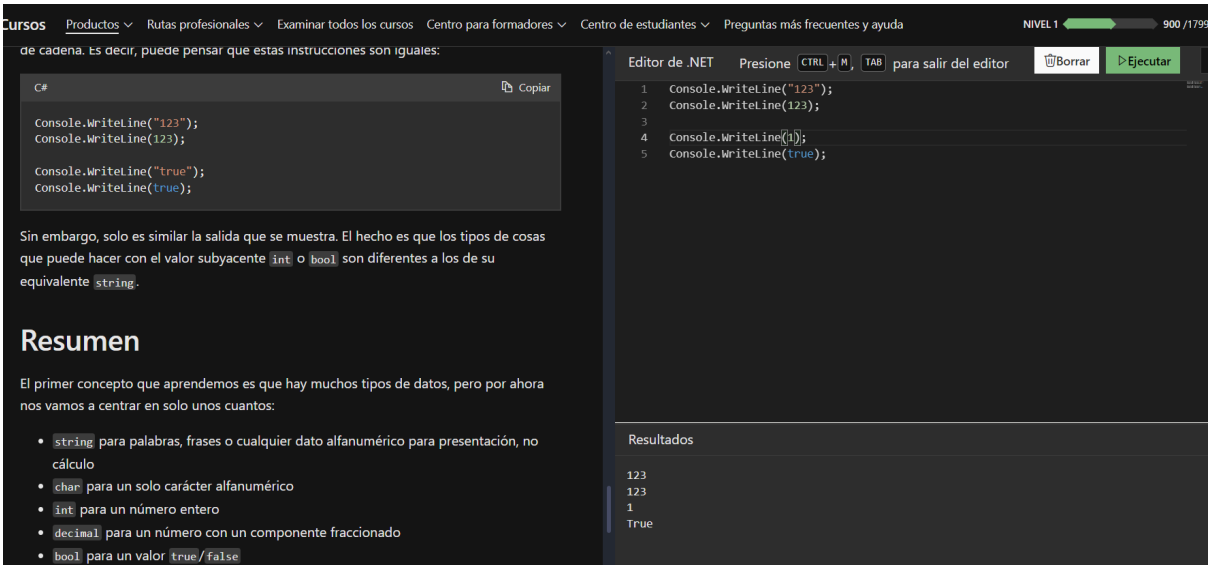


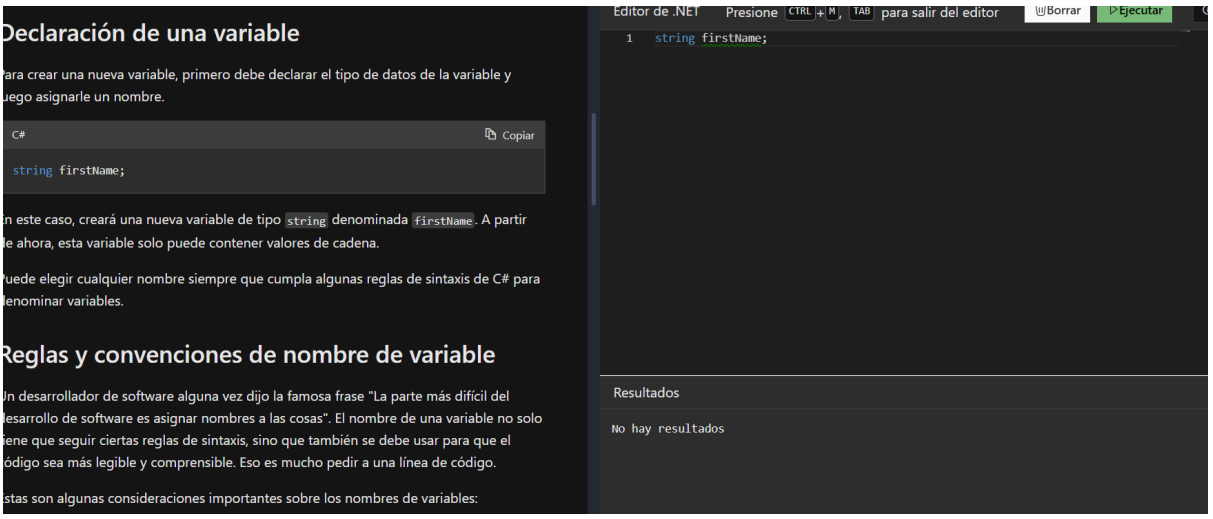

ursos Productos Rutas profesionales Examinar todos los cursos Centro para formadores Centro de estudiantes Preguntas más frecuentes y ayuda NIVEL 1
Uso de literales de punto flotante
Un número de punto flotante es uno que contiene decimales, por ejemplo, 3,14159. C# admite tres tipos de datos para representar números decimales: float, double y decimal. Cada tipo admite distintos grados de precisión.
Output Copiar
Float Type Precision
float ~6-9 digits
double ~15-17 digits
decimal 28-29 digits
Aquí, la precisión refleja el número de dígitos decimales precisos.
1. Agregue la siguiente línea de código en el editor de código:
C# Copiar
Console.WriteLine(0.25F);
Para crear un literal decimal float, anexe la letra F después del número. En este
Editor de .NET Presione CTRL + M, TAB para salir del editor Borrar Ejecuta
1 float varDouble = 1.23;
2
3 Console.WriteLine(varDouble);
Resultados
615


Productos Rutas profesionales Examinar todos los cursos Centro para formadores Centro de estudiantes Preguntas más frecuentes y ayuda NIVEL 1
so de literales de punto flotante
número de punto flotante es uno que contiene decimales, por ejemplo, 3,14159. C# mite tres tipos de datos para representar números decimales: float, double y decimal. da tipo admite distintos grados de precisión.
Output Copiar
loat Type Precision
loat ~6-9 digits
ouble ~15-17 digits
ecimal 28-29 digits
uí, la precisión refleja el número de dígitos decimales precisos.
1. Agregue la siguiente línea de código en el editor de código:
C# Copiar
Console.WriteLine(0.25F);
Editor de .NET Presione CTRL + M, TAB para salir del editor Borrar Ejecut
1
2 double varDouble = 11111111.2322227222;
3
4 Console.WriteLine(varDouble);
Resultados
11111111.2322227


Uso de literales de punto flotante
n número de punto flotante es uno que contiene decimales, por ejemplo, 3,14159. C# dmite tres tipos de datos para representar números decimales: float, double y decimal. ada tipo admite distintos grados de precisión.
Output Copiar
Float Type Precision
float ~6-9 digits
double ~15-17 digits
decimal 28-29 digits
quí, la precisión refleja el número de dígitos decimales precisos.
1. Agregue la siguiente línea de código en el editor de código:
C# Copiar
Console.WriteLine(0.25F);
Para crear un literal decimal float, anexe la letra F después del número. En este
Editor de .NET Presione CTRL + M, TAB para salir del editor Borrar Ejecut
1
2 double varDouble = 11.2322227222;
3
4 Console.WriteLine(varDouble);
Resultados
11.2322227222


ursos Productos Rutas profesionales Examinar todos los cursos Centro para formadores Centro de estudiantes Preguntas más frecuentes y ayuda NIVEL 1 900 /1
12.39816
Uso de literales booleanos
Si queremos imprimir un valor que represente true o false, podemos usar un literal bool.
El término bool es la abreviatura de booleano. En C#, se conoce oficialmente como "bool", pero a menudo los desarrolladores usan el término "booleano".
1. Agregue las siguientes línea de código en el editor de código:
C# Copiar
Console.WriteLine(true);
Console.WriteLine(false);
2. Presione el botón verde Ejecutar para ejecutar el código. Deberíamos ver el resultado siguiente en la consola de salida:
Output Copiar
True
False
Editor de .NET Presione CTRL + M, TAB para salir del editor Borrar Ejecutar
1 Console.WriteLine(true);
2 Console.WriteLine(false);
Resultados
True
False


Cursos Productos Rutas profesionales Examinar todos los cursos Centro para formadores Centro de estudiantes Preguntas más frecuentes y ayuda NIVEL 1 900 /1799
de cadena. Es decir, puede pensar que estas instrucciones son iguales:
C# Copiar
Console.WriteLine("123");
Console.WriteLine(123);
Console.WriteLine("true");
Console.WriteLine(true);
Sin embargo, solo es similar la salida que se muestra. El hecho es que los tipos de cosas que puede hacer con el valor subyacente int o bool son diferentes a los de su equivalente string.
Resumen
El primer concepto que aprendemos es que hay muchos tipos de datos, pero por ahora nos vamos a centrar en solo unos cuantos:
string para palabras, frases o cualquier dato alfanumérico para presentación, no cálculo
char para un solo carácter alfanumérico
int para un número entero
decimal para un número con un componente fraccionado
bool para un valor true/false
Editor de .NET Presione CTRL + M, TAB para salir del editor Borrar Ejecutar
1 Console.WriteLine("123");
2 Console.WriteLine(123);
3
4 Console.WriteLine(1);
5 Console.WriteLine(true);
Resultados
123
123
1
True


Declaración de una variable
Para crear una nueva variable, primero debe declarar el tipo de datos de la variable y luego asignarle un nombre.
C# Copiar
string firstName;
En este caso, creará una nueva variable de tipo string denominada firstName. A partir de ahora, esta variable solo puede contener valores de cadena.
Puede elegir cualquier nombre siempre que cumpla algunas reglas de sintaxis de C# para denominar variables.
Reglas y convenciones de nombre de variable
Un desarrollador de software alguna vez dijo la famosa frase "La parte más difícil del desarrollo de software es asignar nombres a las cosas". El nombre de una variable no solo tiene que seguir ciertas reglas de sintaxis, sino que también se debe usar para que el código sea más legible y comprensible. Eso es mucho pedir a una línea de código.
Estas son algunas consideraciones importantes sobre los nombres de variables:
Editor de .NET Presione CTRL + M, TAB para salir del editor Borrar Ejecutar
1 string firstName;
Resultados
No hay resultados

Estos son algunos ejemplos de declaraciones de variables que usan los tipos de datos que ha aprendido hasta ahora:

C# Copiar

```
char userOption;

int gameScore;

decimal particlesPerMillion;

bool processedCustomer;
```

Resumen

Esto es lo que ha aprendido hasta el momento sobre las variables:

- Las variables son valores temporales que se almacenan en la memoria del equipo.
- Para poder usar una variable, hay que declararla.
- Para ello, primero se selecciona un tipo de datos correspondiente al tipo de datos que se quiere almacenar y, luego, se asigna a la variable un nombre que siga las reglas.

Ahora que sabe cómo declarar una variable, vamos a aprender a establecer, recuperar e inicializar el valor de una variable.

Editor de .NET Presione CTRL + M, TAB para salir del editor Borrar Ejecutar

```
1  string firstName;
2
3  firstName = "Luis";
4
5  Console.WriteLine(firstName);
```

Resultados

```
Luis
```

Complete the Store and Retrieve Data Using Literal and Variable Values in C# module on Microsoft Learn. Then, answer the question below.

Question

Which of the following lines of code creates a variable correctly?

- ○ `int x = 12.3m;`
- ◉ `decimal x = 12.3m;`
- ○ `bool x = 'False';`

Check your answer

Ask for Help