Resumen

En este módulo, se ha revisado la estructura básica de un programa de Rust. La función `main()` es el punto de entrada para todos los programas de Rust. La macro `println!` se puede usar para mostrar valores de variable y el progreso del programa. Las variables se definen con la palabra clave `let`. Sus valores se pueden declarar como mutables (modificables) con la palabra clave mut (inmutables por defecto).

Se han explorado los conceptos básicos del lenguaje, incluidos muchos tipos de datos principales y compuestos. Se ha analizado cómo trabajar con números enteros y de punto flotante, caracteres y cadenas de texto, y valores booleanos true/false. El lenguaje interpreta estrictamente los tipos de dato. Un programa solo se compilara y se ejecutara correctamente cuando los tipos de dato se definan y usen de la manera adecuada.

En el ejercicio, ha escrito una función para crear un automóvil mediante los datos almacenados en `struct` y `enum`.

Ha buscado instancias de la macro `todo!` en el programa de ejemplo y ha completado el código.

En el siguiente módulo de esta ruta de aprendizaje, descubrirá más tipos de datos de Rust y cómo usar expresiones condicionales if/else en un programa.

Más información

Visite los vínculos siguientes para obtener más información sobre algunos de los puntos analizados en este módulo:

- Introducción a Rust
- Más información sobre el tipo de estructura de programación clásica de C
- Revisión de los tipos de datos algebraicos

Documentación de referencia de Rust

- El lenguaje de programación Rust
- Revisión de las palabras clave de Rust

Rust: tipos de datos

- Trabajo con matrices
- Trabajo con valores booleanos
- Trabajo con caracteres
- Trabajo con números decimales y tipos de número de punto flotante
- Trabajo con enumeraciones
- Trabajo con tipos enteros
- Operaciones con cadenas
- Trabajo con estructuras
- Trabajo con tuplas

Rust: conceptos

- Comprensión de las variables, la mutabilidad y la propiedad reemplazada
- Más información sobre las funciones
- Muestra de la salida con la macro println!
- Indicación del código sin terminar con la macro tod