.NET/C# Syntax.

Modern .NET/C# Course

.NET Academy

Content

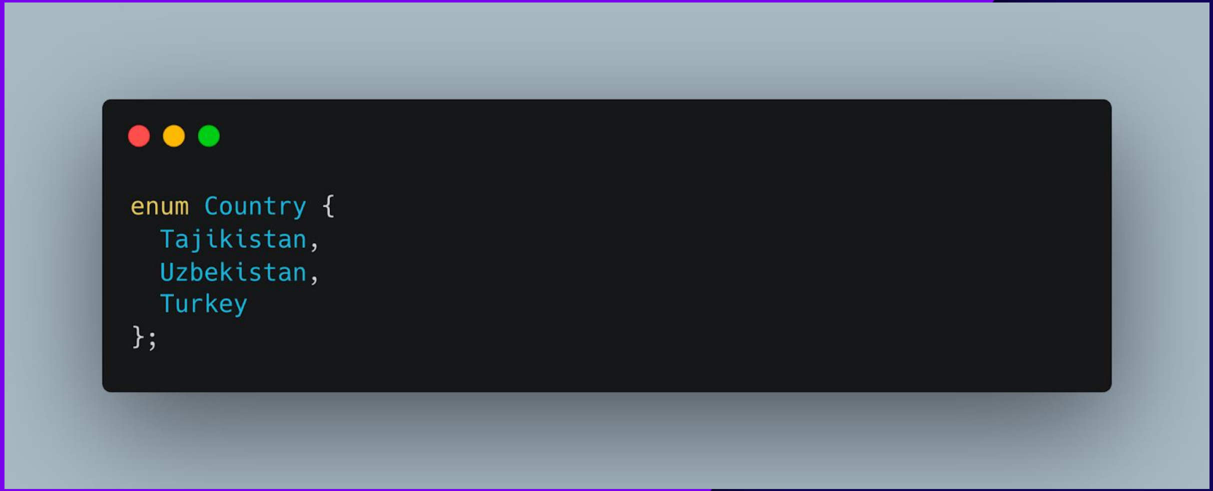
- Enumerations and Constraints.
- Operators.

Enumerations and Constraints.

In the C# language, enum (also called enumeration) is a user-defined value type used to represent a list of named integer constants. It is created using the enum keyword inside a class, structure, or namespace. It improves a program’s readability, maintainability and reduces complexity.

Enumerations and Constraints.

The basic syntax of declaring an enum is:



```
enum Country {
  Tajikistan,
  Uzbekistan,
  Turkey
};
```

Enumerations and Constraints.

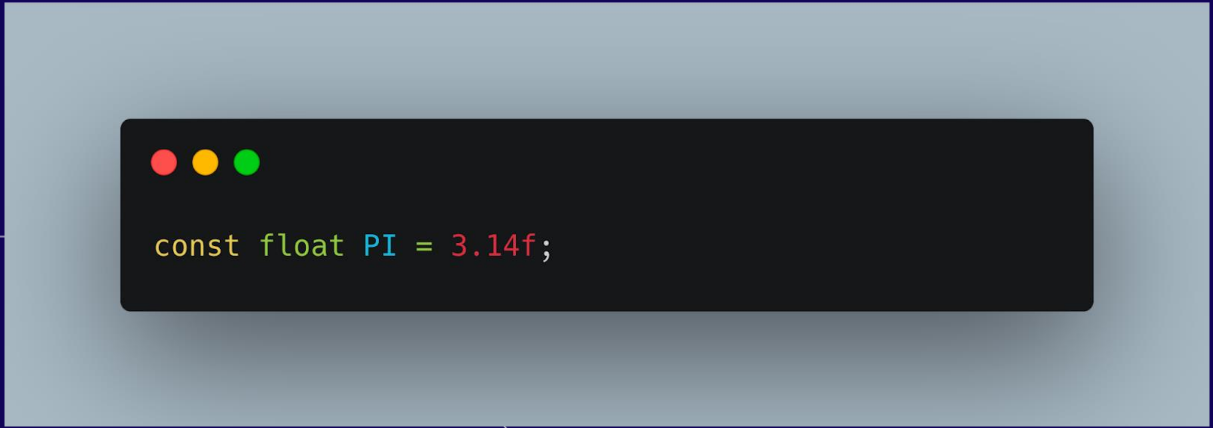
If we do not assign values to the enum items, the items are assigned an integer value by the compiler by default starting from 0. The first item will be assigned 0 and increment by one each time we add an item.

The Constants variables are the immutable values that are known at the time of program compilation and do not change their values for the lifetime of the program.

.NET Academy

Enumerations and Constraints.

In C#, if we declare a variable by using the const keyword, then it is a constant variable and the value of the constant variable can’t be modified once after its declaration. So, it is mandatory to initialize the constant variable at the time of its declaration only. Suppose, you want to declare a constant PI in your program, then you can declare the constant variable as follows:



```
const float PI = 3.14f;
```

Operators.

Operators are symbols that are used to perform operations on operands. Operands may be variables and/or constants.
For example, in 2+3, + is an operator that is used to carry out addition operation, while 2 and 3 are operands.

Operators.

Operators are used to manipulate variables and values in a program. C# supports a number of operators that are classified based on the type of operations they perform.

- Basic assignment operator (=) is used to assign values to variables.

- Arithmetic operators are used to perform arithmetic operations such as addition (+), subtraction (-), multiplication (*), division (/), modulo (%).

- Relational operators are used to check the relationship between two operands. If the relationship is true the result will be true, otherwise it will result in false.

- Logical operators are used to perform logical operation such as and, or. Logical operators operates on boolean expressions (true and false) and returns boolean values. Logical operators are used in decision making and loops.

- Unlike other operators, the unary operators operates on a single operand (++, —, !, +, -).

.NET Academy

Questions for the study of the topic

1. What is an enumeration (enum) in C# and how does it improve a program's readability and maintainability?
2. When values are not explicitly assigned to enum items, how are they assigned by default, and what is the starting value?
3. What is the role of operators in C# and how do they manipulate variables and values?
4. Name some of the arithmetic operators in C# and describe the operations they perform.
5. What are logical operators, and in what context are they typically used in C#?

Homework

1. What is the basic syntax for declaring an enum in C#? Can you provide an example of declaring an enum?
2. What is the purpose of constants in C#? How is a constant variable declared, and what are its characteristics?
3. Can you give examples of basic assignment operators and explain their purpose in C#?
4. Explain the use of relational operators in C# and how they determine the relationship between operands.
5. How do unary operators differ from other operators, and what are some examples of unary operators in C#?

Thank you for your attention.

.NET Academy