C# Enum

Enum in C# is also known as enumeration. It is used to store a set of named constants such as season, days, month, size etc. The enum constants are also known as enumerators. Enum in C# can be declared within or outside class and structs.

Enum constants has default values which starts from 0 and incremented to one by one. But we can change the default value.

Points to remember

- enum has fixed set of constants
- enum improves type safety
- enum can be traversed

C# Enum Example

Let's see a simple example of C# enum.

```
1. using System;
2. public class EnumExample
3. {
4.     public enum Season { WINTER, SPRING, SUMMER, FALL }
5.
6.     public static void Main()
7.     {
8.         int x = (int)Season.WINTER;
9.         int y = (int)Season.SUMMER;
10.        Console.WriteLine("WINTER = {0}", x);
11.        Console.WriteLine("SUMMER = {0}", y);
12.    }
13. }
```

Output:

```
WINTER = 0
SUMMER = 2
```

C# enum example changing start index

```
1. using System;
2. public class EnumExample
3. {
4.     public enum Season { WINTER=10, SPRING, SUMMER, FALL }
5.
6.     public static void Main()
7.     {
8.         int x = (int)Season.WINTER;
```

```
        int y = (int)Season.SUMMER;
        Console.WriteLine("WINTER = {0}", x);
        Console.WriteLine("SUMMER = {0}", y);
    }
}
```

Output:

```
WINTER = 10
SUMMER = 12
```

C# enum example for Days

```
using System;
public class EnumExample
{
    public enum Days { Sun, Mon, Tue, Wed, Thu, Fri, Sat };

    public static void Main()
    {
        int x = (int)Days.Sun;
        int y = (int)Days.Mon;
        int z = (int)Days.Sat;
        Console.WriteLine("Sun = {0}", x);
        Console.WriteLine("Mon = {0}", y);
        Console.WriteLine("Sat = {0}", z);
    }
}
```

Output:

```
Sun = 0
Mon = 1
Sat = 6
```

C# enum example: traversing all values using getNames()

```
using System;
public class EnumExample
{
    public enum Days { Sun, Mon, Tue, Wed, Thu, Fri, Sat };

    public static void Main()
    {
        foreach (string s in Enum.GetNames(typeof(Days)))
        {
            Console.WriteLine(s);
        }
```

```
12.   }
13. }
```

Output:

```
Sun
Mon
Tue
Wed
Thu
Fri
Sat
```

C# enum example: traversing all values using getValues()

```
1.  using System;
2.  public class EnumExample
3.  {
4.     public enum Days { Sun, Mon, Tue, Wed, Thu, Fri, Sat };
5.
6.     public static void Main()
7.     {
8.        foreach (Days d in Enum.GetValues(typeof(Days)))
9.        {
10.          Console.WriteLine(d);
11.       }
12.    }
13. }
```

Output:

```
Sun
Mon
Tue
Wed
Thu
Fri
Sat
```