

طبق [این معرفی](#)، جنریک ها باعث می شوند که نوع داده ای (data type) المان های برنامه در زمان استفاده از آن ها در برنامه مشخص شوند. به عبارت دیگر، جنریک به ما اجازه می دهد کلاس ها یا متدهایی بنویسیم که می توانند با هر نوع داده ای کار کنند.

نکاتی از جنریک ها:

برای به حداکثر رسانی استفاده مجدد از کد، type safety و کارایی است. بیشترین استفاده مشترک از جنریک ها جهت ساختن کالکشن کلاس ها (collection classes) است. تا حد ممکن از جنریک کالکشن کلاس ها (generic collection classes) جدید فضای نام System.Collections.Generic بجای کلاس هایی مانند ArrayList در فضای نام System.Collections استفاده شود. شما می توانید اینترفیس جنریک، کلاس جنریک، متد جنریک و عامل جنریک سفارشی خودتان تهیه کنید. جنریک کلاس ها، ممکن است در دسترسی به متدهایی با نوع داده ای خاص محدود شود. بوسیله reflection، می توانید اطلاعاتی که در یک جنریک در زمان اجرا (run-time) قرار دارد بدست آورید.

انواع جنریک ها:

کلاس های جنریک

اینترفیس های جنریک

متدهای جنریک

عامل های جنریک

در قسمت اول به معرفی کلاس جنریک می پردازیم.

کلاس های جنریک [کلاس جنریک](#) یعنی کلاسی که می تواند با چندین نوع داده کار کند برای آشنایی با این نوع کلاس به کد زیر دقت کنید:

```
using System;
using System.Collections.Generic;

namespace GenericApplication
{
    public class MyGenericArray<T>
    {
        // تعریف یک آرایه از نوع جنریک
        private T[] array;

        public MyGenericArray(int size)
        {
            array = new T[size + 1];
        }

        // بدست آوردن یک آیتم جنریک از آرایه جنریک
        public T getItem(int index)
        {
            return array[index];
        }

        // افزودن یک آیتم جنریک به آرایه جنریک
        public void setItem(int index, T value)
        {
            array[index] = value;
        }
    }
}
```

در کد بالا کلاسی تعریف شده است که می تواند بر روی آرایه هایی از نوع داده ای مختلف عملیات درج و حذف را انجام دهد. برای تعریف کلاس جنریک کافی است عبارت <T> بعد از نام کلاس خود اضافه کنید، سپس همانند سایر کلاس ها از این نوع داده ای در

کلاس استفاده کنید. در مثال بالا یک آرایه از نوع T تعریف شده است که این نوع، در زمان استفاده مشخص خواهد شد. (یعنی در زمان استفاده از کلاس مشخص خواهد شد که چه نوع آرایه ای ایجاد می‌شود)

در کد زیر نحوه استفاده از کلاس جنریک نشان داده شده است، همانطور که مشاهده می‌کنید نوع کلاس int و char در نظر گرفته شده است (نوع کلاس، زمان استفاده از کلاس مشخص می‌شود) و سپس آرایه‌هایی از نوع int و char ایجاد شده است و 5 آیتم از نوع int و char به آرایه‌های هم نوع افزوده شده است.

```
class Tester
{
    static void Main(string[] args)
    {
        // تعریف یک آرایه از نوع عدد صحیح
        MyGenericArray<int> intArray = new MyGenericArray<int>(5);

        // افزودن اعداد صحیح به آرایه ای از نوع عدد صحیح
        for (int c = 0; c < 5; c++)
        {
            intArray.setItem(c, c*5);
        }

        // بدست آوردن آیتم‌های آرایه ای از نوع عدد صحیح
        for (int c = 0; c < 5; c++)
        {
            Console.Write(intArray.getItem(c) + " ");
        }
        Console.WriteLine();

        // تعریف یک آرایه از نوع کاراکتر
        MyGenericArray<char> charArray = new MyGenericArray<char>(5);

        // افزودن کاراکترها به آرایه ای از نوع کاراکتر
        for (int c = 0; c < 5; c++)
        {
            charArray.setItem(c, (char)(c+97));
        }

        // بدست آوردن آیتم‌های آرایه ای از نوع کاراکتر
        for (int c = 0; c < 5; c++)
        {
            Console.Write(charArray.getItem(c) + " ");
        }
        Console.WriteLine();
        Console.ReadKey();
    }
}
```

زمانی که کد بالا اجرا می‌شود خروجی زیر بدست می‌آید:

```
0 5 10 15 20
a b c d e
```